

www.asix.com.pl



Asix4Internet -

Wizualizacja i nadzór nad realizacją procesów przemysłowych z wykorzystaniem przeglądarki internetowej

*Dok. Nr PLP6071
Wersja: 23-09-2010*

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 © 2010, 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	Asix4Internet - wizualizacja i nadzór poprzez Internet	3
2	Wymagania modułów internetowych	5
2.1	Wymagania dotyczące systemu asix	5
2.2	Wymagania dotyczące systemu Windows	5
2.3	Konfiguracja systemu operacyjnego	5
2.4	Dostęp do bazy definicji zmiennych	5
2.5	Wymagania dotyczące projektowania aplikacji za pomocą As2HTML	5
3	Serwer Web Service	7
3.1	Instalacja	7
3.2	Plik konfiguracyjny Web.Config dla serwera Web Service	7
4	AsPortal - Portal Informacji Procesowych	9
4.1	Konfiguracja	10
4.1.1	Udostępnienie aplikacji na serwerze WWW	10
4.1.2	Plik konfiguracyjny Web.config	10
4.1.3	Konfigurator aplikacji AsPortal	12
4.1.4	Konfigurator wyglądu aplikacji AsPortal	13
4.1.5	Pliki definicji raportów	13
4.2	Opis funkcji dostępnych dla użytkownika	15
4.2.1	Dane bieżące	15
4.2.2	Dane archiwalne - tabela	16
4.2.3	Dane archiwalne - wykres	19
4.2.4	Alarmy	20
4.2.5	Baza definicji zmiennych	22
4.2.6	Okno wyboru zmiennych	22
4.2.7	Raporty	23
4.2.8	Projekty	25
4.2.9	Aplikacje	26
5	As2WWW - konwertowanie aplikacji systemu asix	27
5.1	As2WWW - konwertowanie aplikacji systemu asix	27
5.2	Przygotowanie katalogu projektu	28
5.3	Etapy konwersji	29
5.4	Struktura katalogów aplikacji internetowej	30
5.5	Udostępnienie aplikacji na serwerze WWW	30
5.6	Dodatkowe informacje dotyczące konwersji aplikacji systemu asix na aplikację internetową	31
6	As2HTML	33
6.1	As2HTML	33
6.2	Przygotowanie katalogu projektu	34
6.3	Przygotowanie projektu	35
6.4	Obiekt KOMUNIKATY	36
6.5	Obiekt LICZBA	37
6.6	Obiekt SŁUPEK	37
6.7	Obiekt PRZYCISK	39
6.8	Obiekt TEKST	41

6.9	Obiekt OBRAZKI	41
6.10	Obiekt TEKSTY	42
6.11	Obiekt ALARM	43
6.12	Obiekt ZEGAR	43
6.13	Obiekt WYKRES	44
6.13.1	Deklaracja wykresu	44
6.13.2	Projekt wykresu	45
6.13.3	Wykres statyczny generowany po stronie serwera	47
6.13.4	Wykres generowany po stronie klienta	48
6.13.5	Klasa ChartXMLData	50
6.13.6	Flagi ReadFlags	52
6.14	Modyfikowanie obiektów LICZBA i SŁUPEK	52
6.14.1	Stany zmiennej	52
6.14.2	Modyfikowanie wyglądu obiektów LICZBA i SŁUPEK	53
6.14.3	Modyfikowanie działania obiektu LICZBA	55
6.15	Wiązanie obiektu DHTML z wartością bieżącą zmiennej	56

1 Asix4Internet - wizualizacja i nadzór poprzez Internet

Pakiet **asix** zawiera narzędzia umożliwiające wizualizację i nadzór nad realizacją procesów przemysłowych z wykorzystaniem przeglądarki internetowej. Są to moduły związane z Internetem:

- **AsPortal** - Portal Informacji Procesowych,
- **As2HTML** - Biblioteka skryptów i stylów CSS umożliwiająca łatwe tworzenie aplikacji, w której wizualizacja wyświetlana jest w przeglądarce Internet Explorer 6.0 lub nowszej,
- **As2WWW** - Konwerter aplikacji systemu **asix** na aplikację internetową.

Do uruchomienia modułów i korzystania z nich lokalnie wymagane jest posiadanie klucza *HASP asix*. Do korzystania z modułów zdalnie wymagany jest klucz *HASP asix* rozszerzony o licencję klientów internetowych.

2 Wymagania modułów internetowych

2.1 Wymagania dotyczące systemu asix

Przy współpracy modułów internetowych z systemem **asix** pracującym na tym samym komputerze, obsługiwane są wszystkie rodzaje licencji systemu **asix**. Wyjątkiem jest dostęp do alarmów - wówczas wymagany jest **asix** w wersji serwer operatorski (symbol WAXS).

Przy współpracy z systemem **asix** pracującym na innym komputerze, obsługiwane są wszystkie wersje systemu **asix** (w tym również starsze wersje pracujące pod systemem operacyjnym DOS), a wymaganym rodzajem licencji systemu **asix** jest serwer operatorski (symbol WAXS).

2.2 Wymagania dotyczące systemu Windows

Do pracy modułów internetowych wymagany jest system operacyjny Windows XP Professional lub Windows 2003 lub nowszy.

2.3 Konfiguracja systemu operacyjnego

W systemie Windows XP konieczne jest wyłączenie funkcji *Proste udostępnianie plików*. W tym celu należy z menu *Start* wybrać polecenie *Mój komputer* – w ten sposób zostanie otworzone okno *'Mój komputer'*. W otwartym oknie należy z menu *Narzędzia* wybrać polecenie *Opcje folderów*, a następnie zakładkę *Widok*. Na liście *Ustawienia zaawansowane* należy odnaleźć opcję *Użyj prostego udostępniania plików* i opcję tę wyłączyć.

W systemie operacyjnym konieczne jest zainstalowanie serwera WWW (domyślnie podczas instalacji systemów nie jest on instalowany).

Kolejnym wymogiem jest instalacja pakietu .NET 2.0. Pakiet ten można pobrać z serwera www.microsoft.com lub z płyty instalacyjnej oprogramowania **asix**.

2.4 Dostęp do bazy definicji zmiennych

W systemach Windows należy nadać użytkownikowi *ASPNET* pełne prawa do katalogu, w którym znajduje się baza definicji zmiennych. W systemie Windows 2003 i nowszym pełne prawa do tego katalogu należy nadać użytkownikowi *NETWORK SERVICE*.

2.5 Wymagania dotyczące projektowania aplikacji za pomocą As2HTML

Narzędzia projektanta:

- Notatnik;
- Microsoft ASP.NET WebMatrix 0.6;
- FrontPage;

- Visual Studio 2003;
- dowolny inny edytor stron HTML;
- ponadto konieczna jest znajomość podstaw języka HTML.

3 Serwer Web Service

Serwer Web Service pakietu AsixConnect Server umożliwia dostęp do pełnej funkcjonalności aplikacji systemu **asix** za pośrednictwem protokołu XML Web Services. Z serwera Web Service jako źródła danych korzystają aplikacje stworzone przy użyciu modułów As2HTML i As2WWW.

3.1 Instalacja

Serwer WebService znajduje się w katalogu c:\AsixApp\WebService. Aby udostępnić go w sieci należy uruchomić program Internetowe usługi informacyjne. Program znajduje się w menu Start/Panel sterowania/Narzędzia administracyjne. W oknie programu należy podświetlić element Domyślna witryna sieci Web, z menu Akcja/Nowy wybrać polecenie Katalog wirtualny. Uruchamia się wizard, w którym jako Alias należy podać tekst WebService, a jako Katalog c:\AsixApp\WebService. Pozostałe opcje należy pozostawić bez zmian. Domyślnie przy próbach dostępu do serwera używana jest autentyfikacja Windows. Aby włączyć dostęp anonimowy należy podświetlić nowo utworzony katalog wirtualny, z menu *Akcja* wybrać polecenie *Właściwości*, wybrać zakładkę *Zabezpieczenia katalogów*, w polu *Dostęp anonimowy* nacisnąć przycisk *Edytuj* i włączyć opcję *Dostęp anonimowy*.

Od tego momentu serwer Web Service jest dostępny pod adresem:
http://nazwa_komputera/WebService/XConnectWebService.asmx.

Pod adresem:

http://nazwa_komputera/WebService/XConnectWebService.asmx?WSDL dostępny jest opis usług serwera w języku WSDL.

3.2 Plik konfiguracyjny Web.Config dla serwera Web Service

Serwer Web Service używa pliku konfiguracyjnego o nazwie Web.Config do przechowywania domyślnej nazwy kanału. Plik ten znajduje się w katalogu c:\AsixApp\WebService. Sposób definiowania kanałów opisany jest w dokumentacji pakietu AsixConnect.

Aby określić domyślną nazwę kanału należy w pliku *Web.Config*, w elemencie nadrzędnym *configuration* utworzyć element *appSettings*. Następnie w elemencie *appSettings* należy utworzyć jeden element *add* i zdefiniować w nim dwa atrybuty. Pierwszy atrybut należy nazwać *key* i nadać mu wartość „DefaultChannelName”. Drugi atrybut należy nazwać *value* i nadać mu jako wartość nazwę kanału. Nazwę kanału należy ująć w cudzysłowy.




PRZYKŁAD

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="DefaultChannelName" value="AsEmis" />
  </appSettings>
</configuration>
```

Przy definiowaniu kanału należy włączyć opcje *Zmienna kontrolna* i *Limity zmiennych*.

4 AsPortal - Portal Informacji Procesowych

Portal danych procesowych umożliwia przeglądanie wszystkich danych procesowych tj. bazy definicji zmiennych aplikacji, wartości bieżących i archiwalnych pomiarów, archiwum alarmów historycznych oraz listę alarmów aktywnych w postaci tabelarycznej oraz danych historycznych w postaci wykresów. Portal jest aplikacją uniwersalną i aby ją podłączyć do dowolnej aplikacji systemu **asix**, wystarczy skonfigurować ścieżkę do bazy definicji zmiennych.

 Proces konfigurowania aplikacji AsPortal realizowany jest w systemie **asix6** przez **Konfigurator aplikacji AsPortal**, wbudowany w moduł Architekt. Konfigurator automatycznie tworzy kanał AsixConnect, parametryzuje plik Web.Config i nadaje prawa dostępu do katalogu bazy definicji zmiennych. (Patrz: *Architekt – podręcznik użytkownika, rozdz. 3.19. Konfigurator aplikacji AsPortal*).



Portal informacji procesowych AsPortal umożliwia przeglądanie wszystkich rodzajów danych procesowych aplikacji systemu asix:

- Dane bieżące**
Okno danych bieżących służy do prezentacji atrybutów i wartości bieżących zmiennych systemu asix. Dane bieżące prezentowane są w postaci tabel. Kolumna zawierająca aktualne wartości zmiennych jest formatowana w zależności od jakości pomiaru.
- Dane archiwalne**
Okna danych archiwalnych pozwalają na prezentację historycznych wartości pomiarów w postaci tabel lub wykresu.
- Alarmy**
Aplikacja umożliwia prezentację bieżących alarmów występujących w systemie, oraz przeglądanie alarmów historycznych. Istnieje możliwość filtrowania wyświetlanych alarmów.
- Baza definicji zmiennych**
Baza definicji zmiennych umożliwia przeglądanie wszystkich zmiennych systemu asix.
- Raporty**
Portal pozwala na dostęp w sposób interaktywny do raportów procesowych w zakładce.
- Aplikacje**
Portal zawiera łącza do kolejnych pomocniczych aplikacji internetowych.

Rysunek: Strona główna programu AsPortal.

4.1 Konfiguracja

4.1.1 Udostępnienie aplikacji na serwerze WWW

Aby udostępnić aplikację należy wykonać następujące kroki:

- udostępnić katalog `c:\AsixApp\AsPortal` jako wirtualny katalog serwera WWW z domyślnymi opcjami (zaakceptować domyślną nazwę `AsPort`),
- jeżeli użytkownicy aplikacji nie mają skonfigurowanego prawa logowania do serwera Windows, to należy w zabezpieczeniach obu katalogów wirtualnych włączyć dostęp anonimowy.

Po wykonaniu powyższych kroków aplikacja jest dostępna lokalnie po wpisaniu w przeglądarce IE6 adresu:

`http://localhost/AsPortal`

Aplikacja jest dostępna w sieci lokalnej pod adresem

`http://<nazwa komputera w systemie Windows>/AsPortal`



Na przykład dla komputera o nazwie `AsixWeb` adresem aplikacji jest:

`http://AsixWeb/AsPortal`

4.1.2 Plik konfiguracyjny Web.config

AsPortal używa pliku konfiguracyjnego o nazwie `Web.Config` do:

1. przechowywania domyślnej nazwy kanału (jako zbioru opcji konfiguracyjnych serwerów pakietu AsixConnect, opcji połączeń z serwerem systemu **asix** i opcji bazy definicji zmiennych);
2. do definiowania listy wyświetlanych na stronach AsPortala atrybutów zmiennych;
3. do deklarowania połączeń z programami: `AsTrend`, `AsAlarm`, serwerem raportów MS SQL Reporting Services i maskami aplikacji systemu **asix** w wersji WWW

Plik `Web.Config` znajduje się w katalogu `c:\AsixApp\AsPortal`. Sposób definiowania kanałów opisany jest w dokumentacji pakietu AsixConnect.

ad.1

Domyślną nazwę kanału definiują atrybuty `key` i `value`. Atrybut `key` ma nadaną wartość „`DefaultChannelName`”. Drugiemu atrybutowi `value` należy nadać wartość będącą nazwą kanału (nazwę kanału należy ująć w cudzysłowy).



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="DefaultChannelName" value="AsEmis" />
  </appSettings>
```

Przy definiowaniu kanału należy włączyć opcje *Zmienna kontrolna* i *Limity zmiennych*.

ad.2

AsPortal umożliwia zdefiniowanie listy wyświetlanych atrybutów osobno dla danych bieżących, archiwalnych oraz bazy definicji zmiennych i okna wyboru zmiennych. W tym celu należy podać listy atrybutów w odpowiednich pozycjach pliku *Web.config*:

<i>CTableAttributesList</i>	– atrybuty w oknie danych bieżących,
<i>HTTableAttributesList</i>	– atrybuty w oknie danych archiwalnych prezentowanych w postaci tabeli,
<i>HTTableLegendList</i>	– atrybuty wyświetlane w legendzie na stronie wydruku tabeli danych archiwalnych,
<i>HTTableHeaderList</i>	– atrybuty wyświetlane w nagłówkach tabeli zawierającej wartości archiwalne,
<i>HTChartsAttributesList</i>	– atrybuty w oknie danych archiwalnych prezentowanych w postaci wykresu,
<i>HTChartsLegendList</i>	– atrybuty wyświetlane w legendzie na stronie wydruku wykresu danych archiwalnych,
<i>CurrentDataAttributesList</i>	– atrybuty wyświetlane w oknie wyboru zmiennych oraz w bazie zmiennych.

Istnieją dwa specjalne nazwy atrybutów. W przypadku danych bieżących jest nią *ValueCV*, która oznacza bieżącą wartość pomiarów. W oknach danych historycznych szczególną nazwą atrybutu jest *Aggregate* oznaczająca kolumnę agregatów.

W pakiecie **asix** istnieje możliwość skonfigurowania listy wyświetlanych atrybutów (wraz z liczbą wierszy na stronie) za pomocą **Edytora wyglądu aplikacji AsPortal**, wbudowanego w program Architekt. (Patrz: [Konfigurator wyglądu aplikacji AsPortal](#)).

**PRZYKŁAD**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key = "CurrentDataAttributesList" value = "Nazwa,Opis,Jednostka,NazwaAKPiA" />
    <add key = "CTableAttributesList" value = "Nazwa,NazwaAKPiA,Opis,ValueCV,Jednostka " />
    <add key = "HTTableAttributesList" value = "Nazwa, Aggregate, Opis"/>
    <add key = "HTChartsAttributesList" value = "Nazwa, Aggregate, ZakresPomiarowyLo " />
  </appSettings>
```

W pliku *Web.config* można również ustawić liczbę wierszy wyświetlanych z bazy zmiennych na jednej stronie. Określa to pozycja *VBPageSize*. Analogicznie pozycja *ReportsPagiSize* określa liczbę wierszy na stronie raportów. W obu przypadkach wpisanie wartości 0 spowoduje wyłączenie stronicowania.

**PRZYKŁAD**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key = "VBPageSize" value = "8" />
    <add key = "ReportsPagiSize" value = "0" />
  </appSettings>
```

ad.3

Deklaracja połączeń z programami: AsTrend, AsAlarm, serwerem raportów MS SQL Reporting Services i maskami aplikacji systemu **asix** w wersji WWW wymaga podania wartości następujących atrybutów:

key="AsixWwwUrl" value="/Fabryka_WWW"/	- adres aplikacji systemu asix w wersji WWW
key="AsTrendUrl" value="/AsTrend"/	- adres internetowej wersji programu AsTrend

key="AsAlarmUrl" value="/AsAlarmNet"/	- adres internetowej wersji programu AsAlarm
key="ReportServerUrl" value="ReportServer_SQLEXPRESS"/	- adres serwera raportów środowiska raportowego MS SQL Reporting Services

Zadeklarowania pustego tekstu dla atrybutu *value* oznacza blokadę dostępu do danego programu.

W przypadku zadeklarowania pustego tekstu dla atrybutu *value* dla wszystkich programów, zostanie zablokowane wyświetlanie strony 'Aplikacje'.

PRZYKŁAD

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="AsixWwwUrl" value="/Fabryka_WWW"/>
    <add key="AsTrendUrl" value="/AsTrend"/>
    <add key="AsAlarmUrl" value="/AsAlarmNet"/>
    <add key="ReportServerUrl" value="ReportServer_SQLEXPRESS"/>
  </appSettings>
```

4.1.3 Konfigurator aplikacji AsPortal

Wbudowany w program Architekt - konfigurator aplikacji AsPortal pozwala łatwo uruchomić portal informacji procesowych zaprojektowanej aplikacji.

Konfigurator aplikacji AsPortal uruchamiany jest komendą z menu głównego:

Aplikacja > Konfiguruj aplikacje internetowe > zakładka AsPortal

Obsługa Konfiguratora sprowadza się do użycia dwóch przycisków:

Konfiguruj na zakładce *Konfiguracja serwera* - tworzy kanał AsixConnect, parametryzuje plik Web.Config, nadaje prawa dostępu do katalogu bazy definicji zmiennych. Aby komenda zadziałała, należy w pierwszej kolejności zainstalować moduł Asix4Internet z instalatora pakietu **asix**.

Operacje *Konfiguruj* należy wykonać po każdej zmianie nazwy pliku lub katalogu bazy definicji zmiennych.

Uruchom AsPortal na zakładce *Klient* - pozwala uruchomić AsPortal z dostępem do danych procesowych pracującej aplikacji

Aplikacja jest dostępna lokalnie pod adresem:

`http://localhost/AsPortal`

Aplikacja jest dostępna w sieci lokalnej pod adresem:

http://<nazwa komputera w systemie Windows>/AsPortal

Dodatkowo, przycisk **Ustaw** na zakładce *Ustawienia dostępu* - pozwala skonfigurować dostęp do aplikacji AsPortal.

4.1.4 Konfigurator wyglądu aplikacji AsPortal

Zakres wyświetlanych informacji na temat zmiennych procesowych można określić wykorzystując **Edytor wyglądu aplikacji AsPortal** dostępny w programie Architekt, uruchamiany poleceniem:

menu *Aplikacje > Konfiguruj wygląd aplikacji AsPortal*

Edytor pozwala określić listę atrybutów wyświetlanych:

- w oknie podglądu bazy i wyboru zmiennych z podaniem liczby wierszy wyświetlanych na jednej stronie (zakładka **Baza definicji zmiennych**);
- w tabeli danych bieżących (zakładka **Dane bieżące**);
- w tabeli zmiennych, w nagłówkach tabeli wynikowej, w legendzie na stronie wydruku - dla danych prezentowanych w formie tabelarycznej (zakładka **Dane archiwalne - tabela**);
- w tabeli zmiennych, w legendzie na stronie wydruku - dla danych prezentowanych w postaci wykresów (zakładka **Dane archiwalne - wykres**).

Dodatkowo, zakładka **Aplikacje** umożliwia skonfigurowanie adresów aplikacji internetowych, które można uruchamiać z poziomu portalu AsPortal.

4.1.5 Pliki definicji raportów

Na stronie raportów znajduje się drzewo służące do grupowania raportów (dotyczy raportów tworzonych przy użyciu ASTEL/ASTER oraz raportów skryptowych). Elementy drzewa tworzone są na podstawie analizy nagłówek informacyjnych plików definicji raportów, utworzonych w danej aplikacji. Aby raporty były prawidłowo grupowane, należy dodać do plików definicji raportów następujący wiersz (nagłówek informacyjny):

```
znak_komentarza REPORT_INFO=GROUP:Wartosc1; PERIOD:Wartosc2
```

gdzie:

GROUP – nazwa grupy raportów (odpowiada węzłom drzewa);

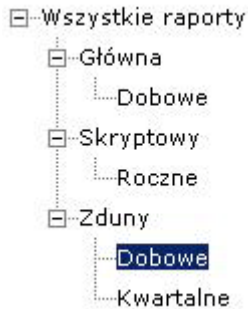
PERIOD – długość okresu czasu generowania raportu (odpowiada „liściom” drzewa). Dostępne wartości to:

- D - dzień
- W - tydzień
- M - miesiąc
- Q - kwartał
- Y - rok
- O – inny

**PRZYKŁAD**

Poniżej fragment kodu raportu z definicją nagłówka.

```
/ REPORT_INFO = GROUP:Zduny; PERIOD:D
```

**PRZYKŁAD**

Poniżej fragment kodu raportu z definicją nagłówka.

```

Edytor FABRYKA - raport dobowy
Plik Pomoc
/REPORT_INFO=GROUP:Główna ; PERIOD:D

{ FABRYKA - raport dobowy }

name "FABRYKA - raport dobowy"

begin 0 step 1 count 24

{ A000 "Temperatura spalin przed odemglaczem" }
{ A004 "Temperatura kwasu siarkowego" }
{ A008 "Temperatura wody ciepłej" }
{ A032 "Temperatura H2S przed piecem" }
{ A036 "Temperatura oparów przed piecem" }

symbol
{S0} "Raport dobowy":40-c,
{S1} "Pomiary temperatur":30-c,
{S2} 'max(A000,@,day<)' :6-1,
{S3} 'max(A004,@,day<)' :6-1,
{S4} 'max(A008,@,day<)' :6-1,
  
```

Rysunek: Fragment kodu raportu z definicją nagłówka.

4.2 Opis funkcji dostępnych dla użytkownika

4.2.1 Dane bieżące

Okno danych bieżących służy do prezentacji atrybutów i wartości bieżących zmiennych systemu **asix**. Dane bieżące prezentowane są w postaci tabeli. Kolumna zawierająca aktualne wartości zmiennych jest formatowana w zależności od jakości pomiaru.

Dla jakości dobrej:

- przy przekroczonym limicie ostrzegawczym wartość wyświetlana jest czerwoną czcionką,
- przy przekroczonym limicie alarmowym wartość wyświetlana jest czerwoną czcionką na żółtym tle.

Dla jakości złej:

- wartość wyświetlana jest białą czcionką na czerwonym tle,
- w przypadku wykrycia uszkodzenia czujnika w miejscu wartości wyświetlane jest „??”,
- w przypadku błędu komunikacji, wyświetlana jest ostatnia znana wartość ze znakiem „?”,
- w pozostałych przypadkach wyświetlany jest znak „?”.

Dostępne operacje, jakie można wykonać w tym oknie to:

- *Dodaj* – otwiera okno wyboru nowych zmiennych (szerzej opisane w punkcie [4.2.6 Okno wyboru zmiennych](#)),
- *Usuń wszystkie* – usuwa zmienne,
- *Odśwież* – pobiera aktualne wartości zmiennych i wyświetla je na ekranie,
- *Odświeżaj stale* - uruchamia mechanizm automatycznego odświeżania danych,
- *Drukuj* - drukowanie wartości zmiennych
- *Excel* - eksport danych do arkusza Excel

Wymień - przycisk umieszczony po prawej stronie rekordu zmiennej – otwiera okno wyboru zmiennych, nowo wybrana zmienna wstawiana jest w miejsce starej.

Usuń - przycisk umieszczony po prawej stronie rekordu zmiennej –usuwa rekord danej zmiennej z tabeli

The screenshot shows the AsPortal web application in Internet Explorer. The main content area is titled 'Dane bieżące' and contains a table with the following data:

Nazwa	Opis_zmiennej	Wartość		
A036	Temperatura oparów przed piecem	93	+<	>+
A056	Temperatura spalin w łączniku	366	+<	>+
A078	Temperatura wymurówki pieca	875	+<	>+
A086	Przepływ wody chłodzącej	153	+<	>+

Rysunek: AsPortal - widok danych bieżących.

4.2.2 Dane archiwalne - tabela

Okna danych archiwalnych pozwalają na prezentację historycznych wartości pomiarów w postaci tabeli lub wykresu.

Aby wyświetlić dane archiwalne w postaci tabeli, należy ustawić następujące pola określające okres czytania danych:

Data: Godzina: Długość: Interwał:

Przyciski w menu mają następujące funkcje:

- „<<” – przesunięcie czasu o długość okresu w tył,
- „>>” – przesunięcie czasu o długość okresu w przód
- „>|” – ustawienie koniec wykresu na chwile bieżącą.

Dostępne operacje, jakie można wykonać w oknie danych archiwalnych to:

- *Dodaj* – otwiera okno wyboru nowych zmiennych (szerzej opisane w punkcie [4.2.6 Okno wyboru zmiennych](#)),
- *Usuń wszystkie* – usuwa wszystkie zmienne,
- *Pokaż dane* - powoduje wyświetlenie danych w postaci tabeli,
- *Drukuj/ eksportuj* - powoduje wyświetlenie danych w postaci przeznaczonej do wydruku lub eksportu do wybranego formatu pliku (np. PDF, Csv)
- *Usuń* – usuwa zmienną,
- *Wymień* – otwiera okno wyboru zmiennych, nowo wybrana zmienna wstawiana jest w miejsce starej.

Dane archiwalne prezentowane w postaci tabeli

The screenshot shows the AsPortal web application interface. The main content area is titled "Dane archiwalne" (Historical Data). It features a toolbar with buttons for "Dodaj" (Add), "Usuń wszystkie" (Remove all), "Pokaż dane" (Show data), and "Drukuj / eksportuj" (Print / Export). Below the toolbar, there are input fields for "Data:" (2010-02-02), "Godzina:" (09:57), "Długość:" (15 minut), and "Interwał:" (1 minuta). The data is presented in two tables.

Idzwa	Agregat	Opis_zmiennej	Wymień	Usuń
A036	Start	Temperatura oparów przed piecem	Wymień	Usuń
A056	Start	Temperatura spalin w łączniku	Wymień	Usuń
A078	Start	Temperatura wymurówki pieca	Wymień	Usuń
A086	Start	Przepływ wody chłodzącej	Wymień	Usuń

Czas	A036 Start	A056 Start	A078 Start	A086 Start
2010-02-02 09:57:08	-33	320	745	199
2010-02-02 09:58:08	15	380	1045	139
2010-02-02 09:59:08	75	420	1045	79
2010-02-02 10:00:08	63	360	745	27
2010-02-02 10:01:08	3	300	445	87
2010-02-02 10:02:08	-21	240	605	147
2010-02-02 10:03:08	39	180	905	207
2010-02-02 10:04:08	99	200	1185	151

Rysunek: AsPortal - dane archiwalne w postaci tabeli.

Page 1 of 1

Fabryka

Dane archiwalne

Okres danych	
Początek okresu danych	2010-02-02 09:57:08
Długość okresu danych	00:15:00
Długość okresu agregacji	00:01:00

Nazwa	Agregat	Opis_zmiennej
A036	Start	Temperatura oparów przed piecem
A056	Start	Temperatura spalin w łączniku
A078	Start	Temperatura wymurówki pieca
A086	Start	Przepływ wody chłodzącej

Czas	A036	A056	A078	A086
2010-02-02 09:57:08	-33	320	745	199
2010-02-02 09:58:08	15	380	1045	139
2010-02-02 09:59:08	75	420	1045	79
2010-02-02 10:00:08	63	360	745	27
2010-02-02 10:01:08	3	300	445	87
2010-02-02 10:02:08	-21	240	605	147
2010-02-02 10:03:08	39	180	905	207
2010-02-02 10:04:08	99	200	1185	151
2010-02-02 10:05:08	39	260	885	91
2010-02-02 10:06:08	-21	320	585	31
2010-02-02 10:07:08	3	380	465	75
2010-02-02 10:08:08	63	420	765	135
2010-02-02 10:09:08	75	360	1065	195
2010-02-02 10:10:08	15	300	1025	163
2010-02-02 10:11:08	-33	240	725	103

Rysunek: AsPortal - widok danych archiwalnych w postaci tabelarycznej do wydruku.

4.2.3 Dane archiwalne - wykres

Aby wyświetlić dane archiwalne w postaci wykresu, należy ustawić następujące pola określające okres czytania danych:

Wykres:	Data:	Godzina:	Długość:	Interwał:
Liniowy	2010-02-02	09:57	0	15 minut
				1 minuta

Przyciski w menu mają następujące funkcje:

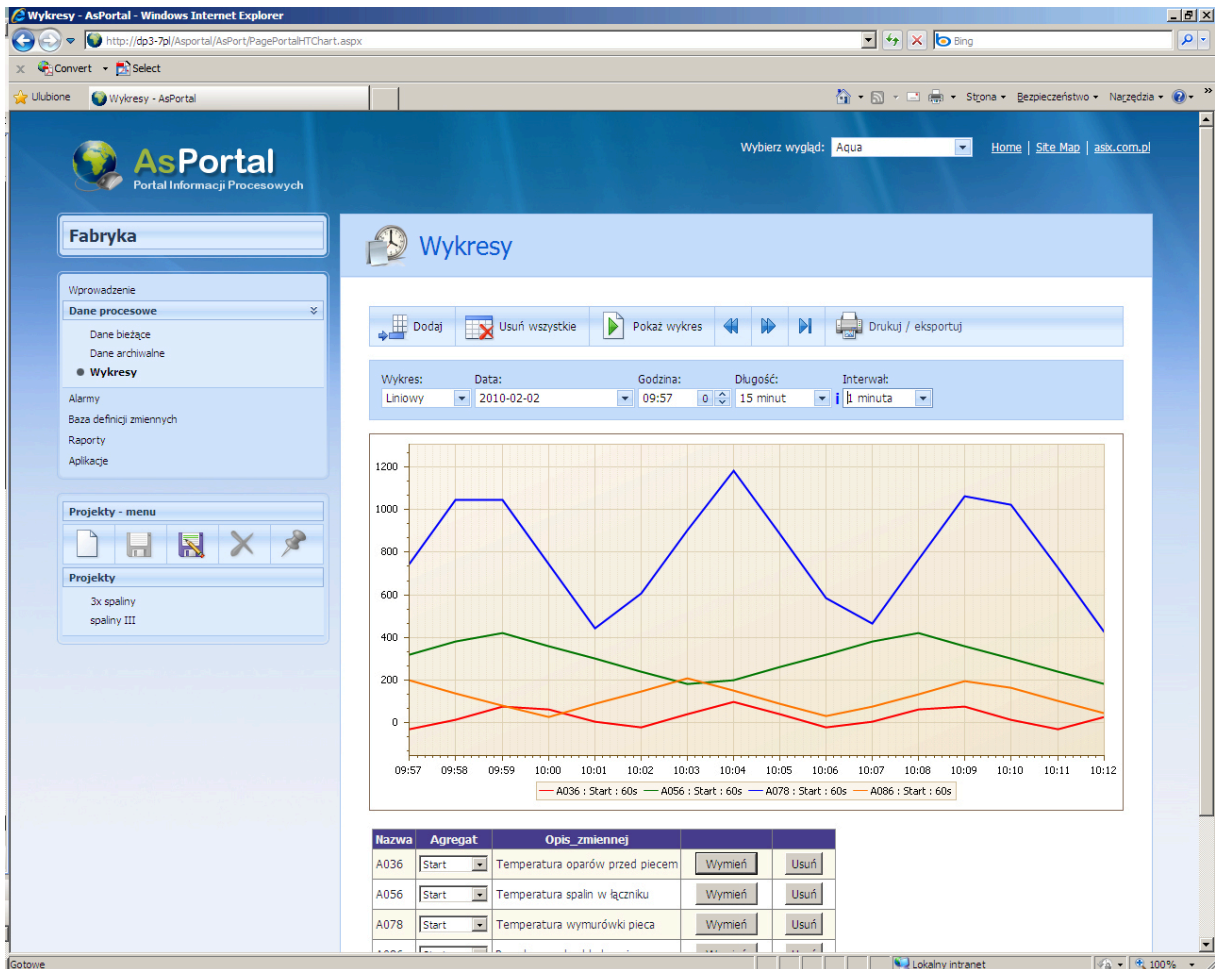
- „<<” – przesunięcie czasu o długość okresu w tył,
- „>>” – przesunięcie czasu o długość okresu w przód
- „>|” – ustawienie koniec wykresu na chwili bieżącej.

Dostępne operacje, jakie można wykonać w oknie danych archiwalnych to:

- *Dodaj* – otwiera okno wyboru nowych zmiennych (szerzej opisane w punkcie [4.2.6 Okno wyboru zmiennych](#)),
- *Usuń wszystkie* – usuwa wszystkie zmienne,
- *Pokaż wykres* – zastosowanie wprowadzonych zmian i wyświetlenie tabeli/wykresu,
- *Drukuj/ eksportuj* - powoduje wyświetlenie danych w postaci przeznaczonej do wydruku lub eksportu do wybranego formatu pliku (np. PDF, Csv)
- *Usuń* – usuwa zmienną,
- *Wymień* – otwiera okno wyboru zmiennych, nowo wybrana zmienna wstawiana jest w miejsce starej.

Dane archiwalne prezentowane w postaci wykresu

Ustawienie interwału agregowania na wartość *auto* powoduje automatyczne ustawienie jego wartości na 1/360 długości okresu wyświetlania. Jeżeli okres archiwizacji danej zmiennej jest większy niż wyliczona wartość interwału, to wartość interwału dla zmiennej jest zwiększana do okresu archiwizacji.



Rysunek: AsPortal - dane archiwalne w postaci wykresu.

4.2.4 Alarmy

Aplikacja umożliwia prezentację bieżących alarmów występujących w systemie, oraz przeglądanie alarmów historycznych. Tekst alarmu wyświetlany jest w kolorze zależnym od typu alarmu:

Systemowy
 Komunikat
 Ostrzeżenie
 Alarm
 Ważny alarm

Istnieje możliwość filtrowania wyświetlanych alarmów. Po naciśnięciu łącza *Pokaż filtr* rozwija się okno z dostępnymi opcjami filtrowania. Można filtrować alarmy ze względu na typ, status, grupę oraz zawarty tekst.

Naciśnięcie przycisku *Odśwież* powoduje odświeżenie danych.

W oknie alarmów historycznych należy ustawić horyzont wyświetlania danych (czas początkowy oraz długość okresu).

Poniższe rysunki prezentują okna alarmów – odpowiednio alarmów aktywnych z włączoną opcją filtrów oraz alarmów historycznych.

Alarmy aktywne - AsPortal - Windows Internet Explorer

http://dp3-7pl/Asportal/AsPort/PagePortalALActive.aspx

Wybierz wygląd: Aqua | Home | Site Map | asx.com.pl

Fabryka

Wprowadzenie
Dane procesowe
Alarmy
● **Alarmy aktywne**
Alarmy historyczne
Baza definicji zmiennych
Raporty
Aplikacje

Projekty - menu
Projekty

Alarmy aktywne

Odśwież Drukuj / eksportuj...

Filtr

Data ostatniego odczytu: 2010-02-02 10:43:40
Liczba aktywnych alarmów: 1
Filtrowanie włączone

ID	Tekst alarmu	Data
264	Zawór regulacyjny stężenia kwasu siarkowego dopływającego - remont	2010-02-02 10:31:13

Lokalny intranet

Rysunek: AsPortal - podgląd alarmów aktywnych.

Alarmy historyczne - AsPortal - Windows Internet Explorer

http://dp3-7pl/Asportal/AsPort/PagePortalALHistorical.aspx

Wybierz wygląd: Aqua | Home | Site Map | asx.com.pl

Fabryka

Wprowadzenie
Dane procesowe
Alarmy
Alarmy aktywne
● **Alarmy historyczne**
Baza definicji zmiennych
Raporty
Aplikacje

Projekty - menu
Projekty

Alarmy historyczne

Czytaj Drukuj / eksportuj...

Data: 2010-02-02 Godzina: 09:56 Długość: 15 minut

Filtr

Liczba alarmów: 4
Filtrowanie włączone

Id	Tekst alarmu	Data
304	Kłapa regulacyjna temperatury w aparacie kontaktowym (po III półce) - remont	P 2010-02-02 09:56:45
256	Kłapa regulacyjna temperatury gazów kwaśnych (spalin) - remont	K 2010-02-02 10:02:43
304	Kłapa regulacyjna temperatury w aparacie kontaktowym (po III półce) - remont	K 2010-02-02 10:05:09
320	Kłapa regulacyjna przepływu H2S do pieca - remont	K 2010-02-02 10:07:46

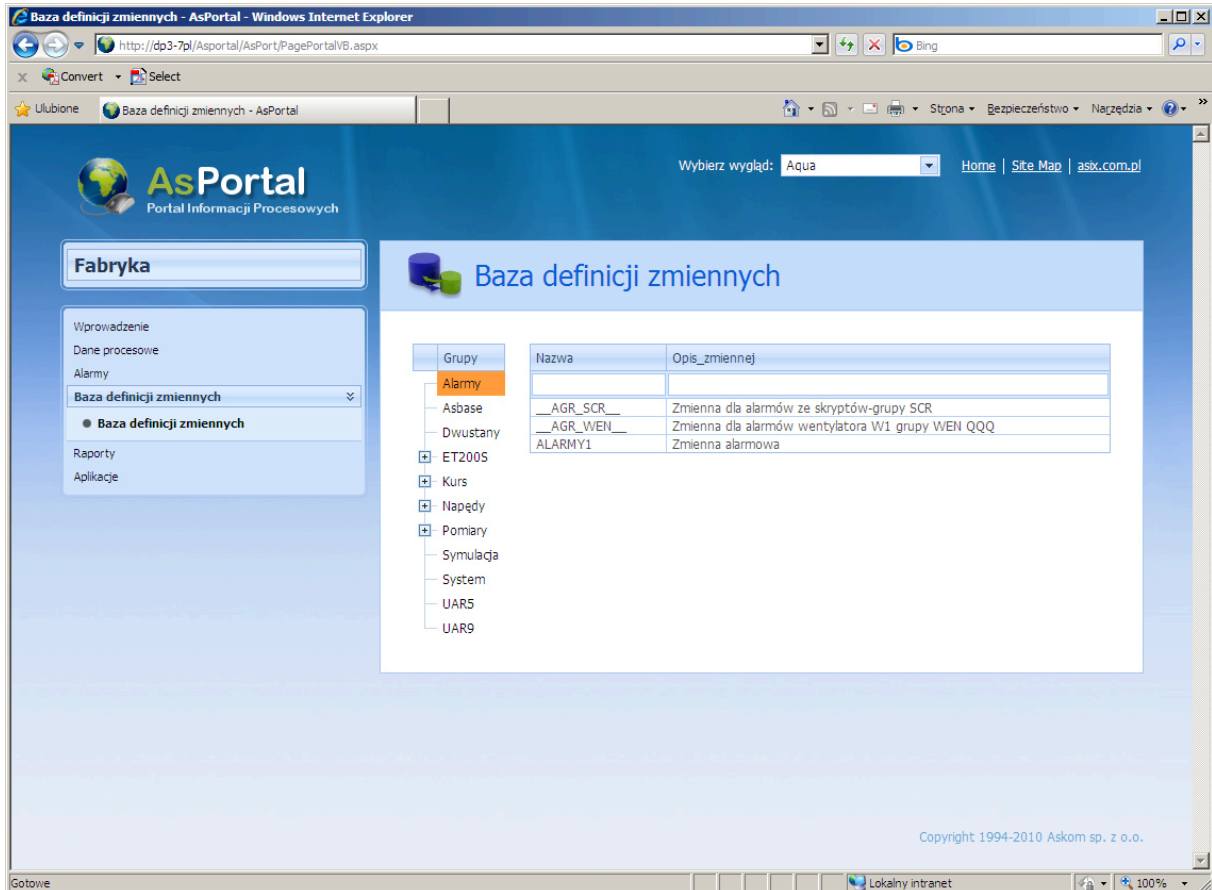
Gotowe

Lokalny intranet

Rysunek: AsPortal - podgląd alarmów historycznych.

4.2.5 Baza definicji zmiennych

Baza definicji zmiennych umożliwia przeglądanie wszystkich zmiennych systemu **asix**. Okno bazy przechowuje swoją ostatnią pozycję, dzięki czemu po ponownym otwarciu zostaje automatycznie załadowana ostatnio przeglądana strona.



Rysunek: AsPortal - widok bazy definicji zmiennych.

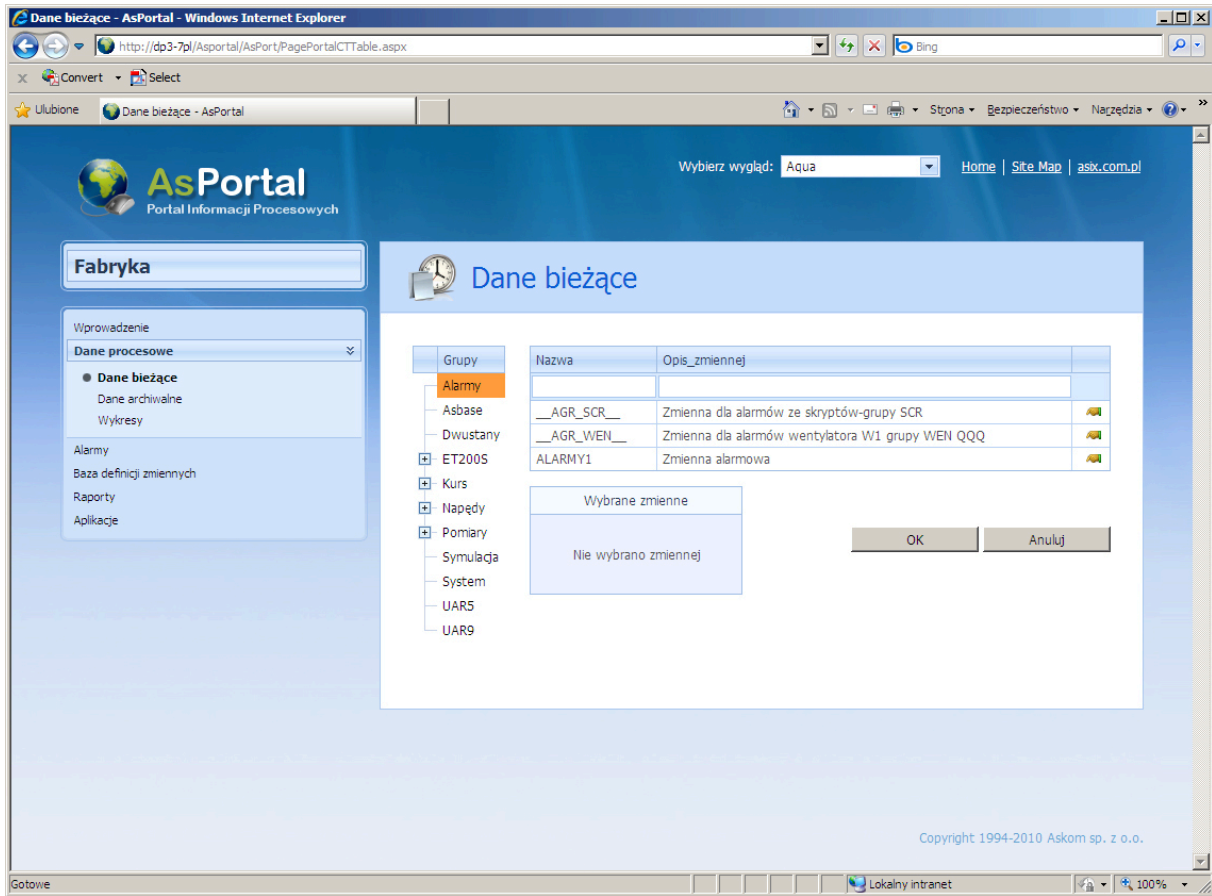
4.2.6 Okno wyboru zmiennych

W oknie wyboru zmiennych wszystkie dostępne zmienne podzielone są na grupy. Możliwe jest wybranie wielu zmiennych z różnych grup.

W dolnej części ekranu prezentowane są wybrane zmienne w postaci tabeli. Możliwe jest usunięcie dowolnej zmiennej z listy wybranych. Naciśnięcie przycisku **OK** powoduje przesłanie zmiennych do strony macierzystej.

W przypadku, gdy zmienna jest wymieniana, wybranie nowej zmiennej powoduje natychmiastowy powrót do strony macierzystej.

Okno wyboru zmiennych przechowuje swoją ostatnią pozycję osobno dla danych bieżących, archiwalnych oraz bazy zmiennych, co ułatwia dodawanie kolejnych zmiennych do projektów.



Rysunek: AsPortal - okno wyboru zmiennych.

4.2.7 Raporty

Portal pozwala na dostęp w sposób interaktywny do raportów wygenerowanych w aplikacji systemu **asix**. Po otwarciu strony widoczne jest drzewo grup raportów oraz tabela ze szczegółowymi informacjami.

The screenshot shows the AsPortal interface. On the left, a navigation menu includes 'Fabryka', 'Wprowadzenie', 'Dane procesowe', 'Alarmy', 'Baza definicji zmiennych', 'Raporty' (selected), and 'Aplikacje'. The main area is titled 'Raporty' and features a tree view with 'Bio' and 'Główna' groups. A date range filter is set to 'Od: 2007-01-02' and 'Do: 2010-02-02'. A table displays the following reports:

Data raportu	Nazwa raportu	Rozmiar pliku [B]	Data utworzenia pliku	Typ	Wybierz
2009-09-07	TWP Nawęglanie - raport dobowy (ERSA)	6463	2009-12-07 12:32:22	Dobowy	
2009-09-07	_Nawęglanie-biomasa - raport dobowy (URE)	4807	2009-12-07 12:33:00	Dobowy	
2009-09-01	TWP Nawęglanie - raport miesięczny (ERSA)	6626	2009-12-07 12:32:46	Miesięczny	
2009-09-01	_Nawęglanie-biomasa - raport miesięczny (URE)	7438	2009-12-07 12:33:04	Miesięczny	

Rysunek: AsPortal - zestawienie raportów.

Aby wyszukać żądane raporty, należy wybrać odpowiednią pozycję z drzewa grup raportów oraz ustawić następujące opcje filtrowania:

Przedział czasowy z którego pochodzą raporty

Zaznaczenie pola powoduje, że w przypadku gdy istnieje kilka raportów z danego okresu, wyświetlany będzie tylko najbardziej aktualny (ostatni) raport.

Dostępne operacje w oknie raportów:

- *Odśwież* – powoduje odświeżenie informacji o raportach z uwzględnieniem opcji filtrowania oraz wyświetlenie nowych raportów, jeżeli takie powstały od ostatniego odświeżania;
- *Drukuj* – po wybraniu raportu (wyświetleniu podglądu) możliwe jest otwarcie nowego okna z raportem gotowym do wydruku;
- *Wybierz* – wyświetlenie podglądu danego raportu.

The screenshot shows the AsPortal web interface in Internet Explorer. The main content area displays a report for the 'Bio' group, filtered by dates from 2007-01-02 to 2010-02-02. A table lists several reports, including 'TWP Nawęglanie - raport dobowy (ERSA)' and '_Nawęglanie-biomasa - raport dobowy (URE)'. Below this, a detailed report for '07.09.2009' is shown, containing a table with columns for 'Godz.', 'UDZIAŁ BIOM18', 'WAGA WĘGLA W7a', 'WAGA WĘGLA W7b', 'WAGA WĘGLA W15', 'WAGA WĘGLA W6d', 'WĘGIEL14 W7a,W7b,W15,W6d', 'WĘGIEL58 W7a,W7b,W15,W6d', 'WĘGIEL18 W7a,W7b,W15,W6d', 'WAGA BIOMASY W14', 'WAGA BIOMASY W6e, W14', and 'BIOMAS NA BL.'. The table contains 14 rows of data representing hourly intervals.

Rysunek: AsPortal - podgląd wybranego raportu.

4.2.8 Projekty

Aplikacja AsPortal umożliwia tworzenie projektów danych bieżących, archiwalnych oraz alarmów. W ramach projektu przechowywane są informacje o wybranych zmiennych (alarmach). Dla danych archiwalnych pamiętane są dodatkowo agregaty, interwał agregowania, przedział czasu oraz w przypadku wykresu jego typ.

Do obsługi projektów służy menu znajdujące się w lewej części ekranu:

The screenshot shows the AsPortal interface. On the left is a navigation menu for 'Fabryka' (Factory) with categories like 'Wprowadzenie', 'Dane procesowe', 'Alarmy', 'Baza definicji zmiennych', 'Raporty', and 'Aplikacje'. Below this is a 'Projekty - menu' section with icons for file operations and a list of projects including '3x spaliny' and 'spaliny III'. The main area is titled 'Dane bieżące' and features a toolbar with actions: 'Dodaj', 'Usuń wszystkie', 'Odśwież', 'Odświeżaj stale', 'Drukuj', and 'Excel'. Below the toolbar is a table of real-time data.

Nazwa	Opis_zmiennej	Wartość		
A036	Temperatura oparów przed piecem	57	+/-	✖
A056	Temperatura spalin w łączniku	282	+/-	✖
A078	Temperatura wymurówki pieca	555	+/-	✖
A086	Przepływ wody chłodzącej	49	+/-	✖

Rysunek: Widok menu projektów.

Polecenia menu mają następujące działanie:

- Nowy* – utworzenie nowego (pustego) projektu,
- Zapisz* – zapisanie zmian w projekcie,
- Zapisz jako* – zapisanie projektu pod wybraną nazwą,
- Usuń* – usunięcie projektu,
- Ustaw jako startowy* – ustawienie startowego projektu, który będzie ładowany automatycznie po otwarciu strony.

4.2.9 Aplikacje

Strona Aplikacje zawiera łącza do kolejnych wyspecjalizowanych aplikacji internetowych: AsTrend, AsAlarm, serwera raportów MS SQL Reporting Services i masek aplikacji systemu **asix** w wersji WWW.

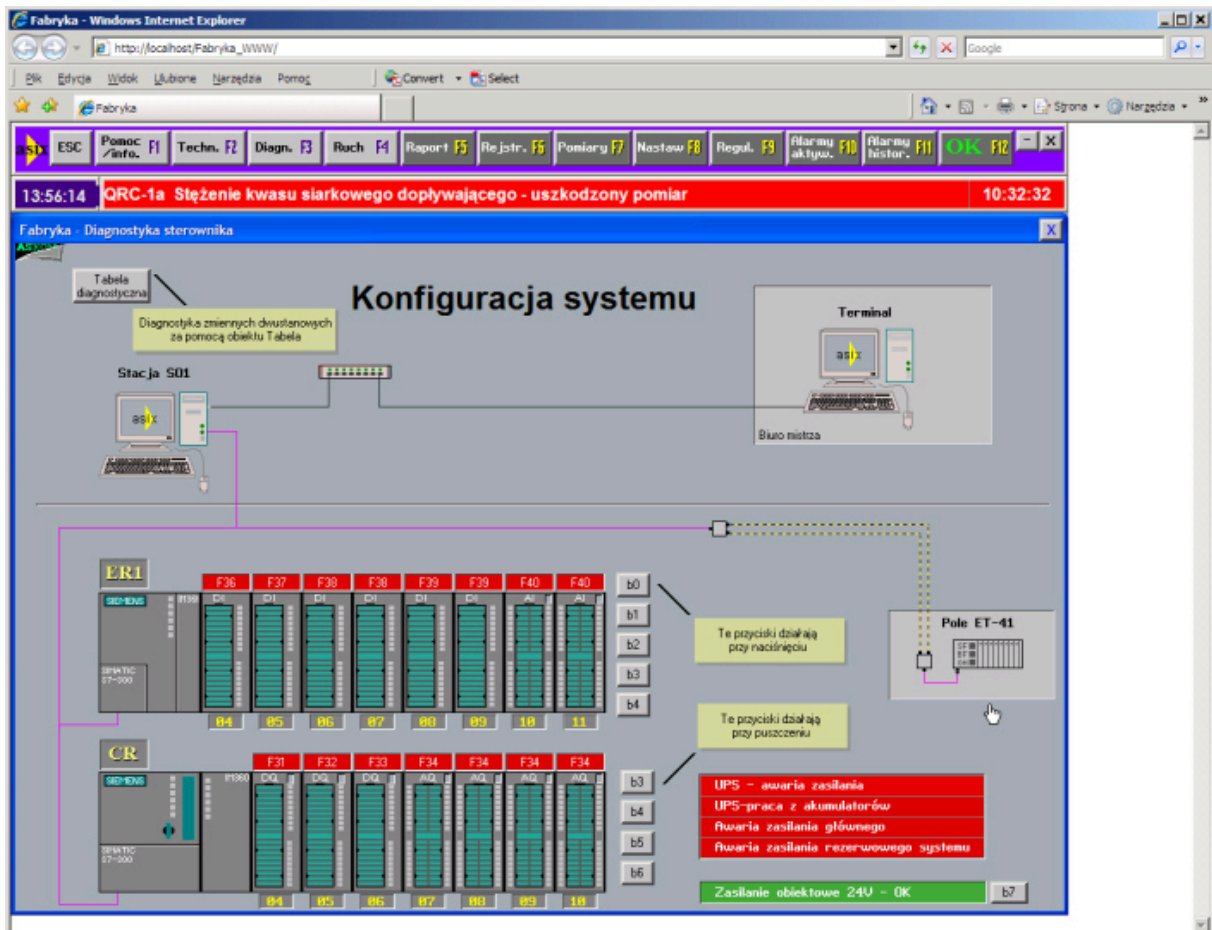
5 As2WWW - konwertowanie aplikacji systemu asix

5.1 As2WWW - konwertowanie aplikacji systemu asix

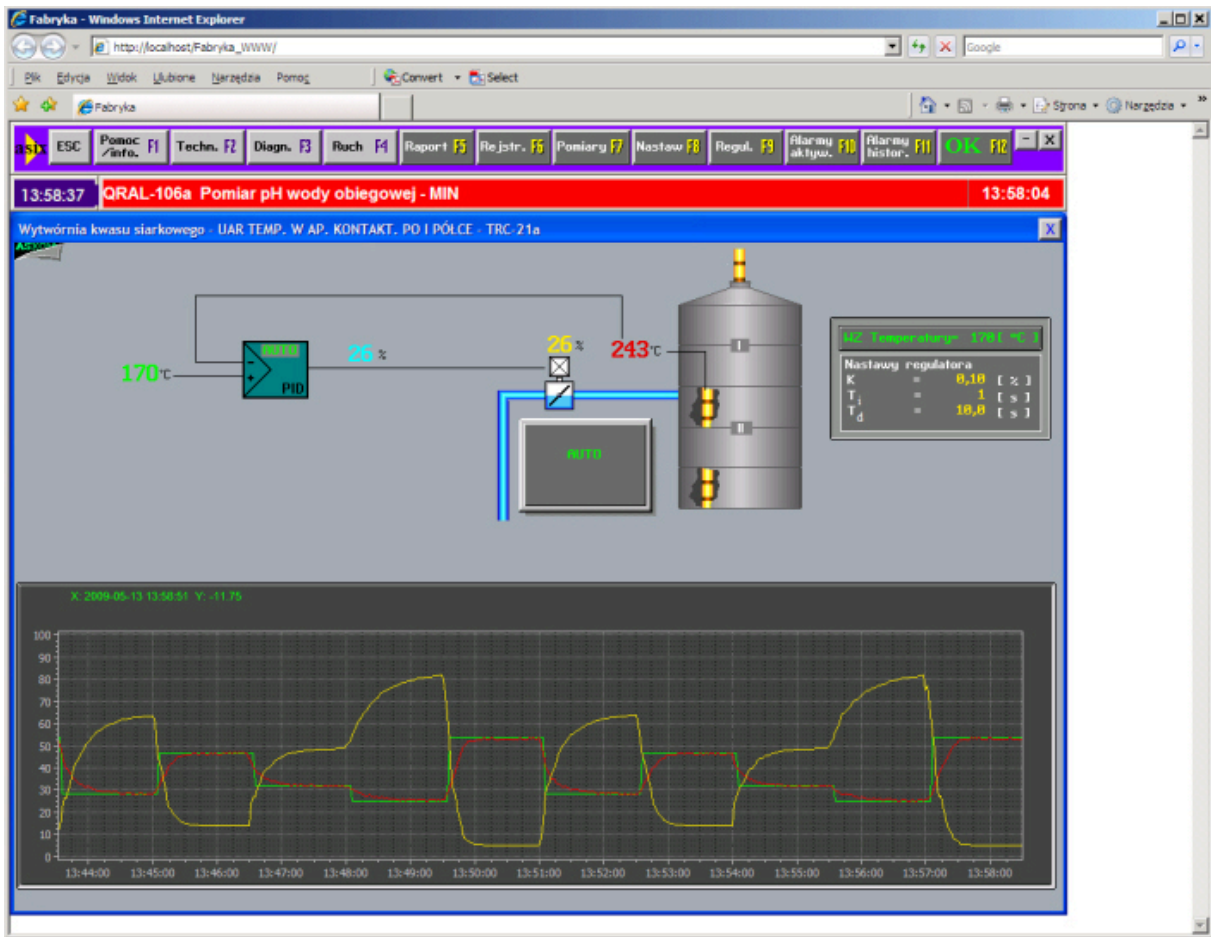
Pakiet Asix4Internet wyposażony jest w moduł As2WWW, który służy do konwersji aplikacji systemu **asix** do aplikacji internetowej. Otrzymana w wyniku konwersji aplikacja uruchamiana jest na serwerze internetowym i może być oglądana przez użytkownika w przeglądarce Internet Explorer 6/7.

i Wszystkie procesy związane z przekonwertowaniem aplikacji realizowane są przez **Generator aplikacji WWW**, wbudowany w program Architekt, który pozwala wygenerować gotową aplikację internetową jednym przyciśnięciem przycisku. (Patrz: *Architekt – podręcznik użytkownika*, rozdz. 3.20. *Generator aplikacji WWW*).

W przypadku konwertowania aplikacji systemu **asix** na aplikację internetową poprzez „ręczne” uruchamianie przez użytkownika programów realizujących kolejne etapy konwersji, należy postępować zgodnie z procedurą opisaną w kolejnych podrozdziałach (zawarty tam opis odnosi się do konwertowania aplikacji systemu **asix4**, działającej w oparciu o pliki .INI).



Rysunek: Widok maski (1) przekonwertowanej aplikacji asixa, oglądanej w IE 7.



Rysunek: Widok maski (2) przekonwertowanej aplikacji asixa, oglądanej w IE 7.

5.2 Przygotowanie katalogu projektu

W katalogu projektu należy założyć katalog bin i do katalogu skopiować plik C:\Program Files\Askom\Asix\XConnectNetCS.dll.

Aby udostępnić katalog w sieci, należy uruchomić program *Internetowe usługi informacyjne*. Program znajduje się w menu *Start/Panel sterowania/Narzędzia administracyjne*. W oknie programu należy podświetlić element *Domyślna witryna sieci Web*, z menu *Akcja/Nowy* wybrać polecenie *Katalog wirtualny*. Uruchamia się wizard, w którym jako *Alias* należy podać nazwę, pod jaką katalog ma być widoczny w Internecie, a jako *Katalog* pełną ścieżkę do katalogu projektu. Pozostałe opcje należy pozostawić bez zmian.

Domyślnie przy próbach dostępu do serwera używana jest autentyfikacja Windows. Aby włączyć dostęp anonimowy należy podświetlić nowo utworzony katalog wirtualny, z menu *Akcja* wybrać polecenie *Właściwości*, wybrać zakładkę *Zabezpieczenia katalogów*, w polu *Dostęp anonimowy* nacisnąć przycisk *Edytuj* i włączyć opcję *Dostęp anonimowy*.

5.3 Etapy konwersji

Konwersja aplikacji systemu **asix** do aplikacji internetowej składa się z kilku etapów. Etapy te to:

1. Konwersja plików masek MSK do plików w formacie XML,
2. Wygenerowania bitmap zawierających wszystkie obiekty statyczne masek (bitmapy używane będą jako tła stron internetowych)
3. Wyodrębnienie bitmap z pliku DAT aplikacji i zapisanie ich w formacie PNG,
4. Wygenerowanie strony startowej,
5. Konwersja plików masek z formacie XML do formatu serwera internetowego ASPX,
6. Konwersja plików definicji menu PUM do formatu serwera internetowego ASPX.

Każdy etap wykonywany jest przez osobny program. Parametry wywołań programów podane są poniżej. Każdy z programów obsługuje parametr /o za pomocą, którego określa się katalog wyjściowy programu.

Do konwersji plików masek MSK do plików w formacie XML służy program *msk2xml*. Parametry wywołania programu są następujące:

```
msk2xml <wzorzec_nazwy_plikow_masek> [/s] [/o<katalog_wynikowy>]
```

Jako <wzorzec_nazwy_plikow_masek> można podać pełną nazwę maski np. *MAPA_KW.MSK* lub wzorzec obejmujący wszystkie nazwy w katalogu np. **.MSK*.

Jeżeli użyty zostanie parametr /s to zostanie tylko wyświetlona statystyka obiektów użytych w maskach.

Do generowania bitmap zawierających obrazy obiektów statycznych masek służy program *msk2png*. Parametry wywołania programu są następujące:

```
msk2png <wzorzec_nazwy_plikow_masek> [<plik_ini>] [/d] [/i] [/o<katalog_wynikowy>]
```

Jako <wzorzec_nazwy_plikow_masek> można podać pełną nazwę maski np. *MAPA_KW.MSK* lub wzorzec obejmujący wszystkie nazwy w katalogu np. **.MSK*.

Jeżeli użyty zostanie parametr /d to dodane zostaną również obrazy obiektów dynamicznych.

Jeżeli użyty zostanie parametr /i to dodane zostaną również obrazy obiektów interaktywnych.

Do konwersji pliku bitmap aplikacji DAT do plików w formacie PNG służy program *DatToBmp*. Parametry wywołania programu są następujące:

```
DatToBmp <plik_dat> <nazwa_bitmap> [<plik_ini>] [/o<katalog_wynikowy>]
```

Jako parametr <plik_dat> należy podać nazwę pliku zawierającego wszystkie bitmapy aplikacji.

Jako parametr <nazwa_bitmap> należy podać nazwę bitmapy, która ma zostać wyodrębniona lub wzorzec tych nazw. Użycie nazwy * powoduje wyodrębnienie wszystkich bitmap.

Do wygenerowania strony startowej aplikacji służy program *ini2frameset*. Parametry wywołania programu są następujące:

```
ini2frameset <plik_ini> [/o<katalog_wynikowy>]
```

Jako parametr <plik_ini> należy podać nazwę pliku *ini* aplikacji. W wyniku działania programu powstaje plik *default.htm* która jest stroną startową aplikacji.

Do konwersji plików masek w formacie XML do stron ASPX służy program *xml2aspx*. Parametry wywołania programu są następujące:

```
xml2aspx <wzorzec_nazwy_plikow_xml> <plik_ini> [/o<katalog_wynikowy>]
```

Jako `<wzorzec_nazwy_plikow_xml>` zwykle podaje się `*.XML` co obejmuje wszystkie nazwy plików w katalogu.

Do konwersji plików definicji menu PUM do formatu serwera internetowego ASPX służy program `pum2aspx`. Parametry wywołania programu są następujące:

```
pum2aspx <wzorzec nazwy plikow pum> [/o<katalog_wynikowy]
```

Jako parametr `<wzorzec_nazwy_plikow_pum>` zwykle podaje się `*.PUM`, co obejmuje wszystkie nazwy plików w katalogu.

5.4 Struktura katalogów aplikacji internetowej

Pierwszym krokiem konwersji jest utworzenie katalogu, w którym ma być umieszczona aplikacja internetowa. Może to być katalog na tym samym poziomie w systemie plików, co katalog aplikacji systemu **asix** a jako jego nazwę można użyć nazwę katalogu aplikacji systemu **asix** z dodanym przyrostkiem `_www`. Następnie należy w katalogu aplikacji internetowej utworzyć podkatalogi `bin`, `img`, `tee` i `xml`.

Do katalogu `bin` należy skopiować plik `C:\Program Files\Askom\Asix\XConnectNetCS.dll`. W katalogu `img` będą umieszczane pliki bitmap, w katalogu `tee` szablony wykresów a w katalogu `pośredni` format maski, czyli format `xml`.

5.5 Udostępnienie aplikacji na serwerze WWW

Aby udostępnić aplikację należy wykonać następujące kroki:

- Udostępnić jako wirtualny katalog serwera WWW katalog `c:\AsixApp\WebService` z domyślnymi opcjami (zaakceptować domyślną nazwę `WebService`),
- Udostępnić katalog, w którym znajduje się aplikacja internetowa jako wirtualny katalog serwera WWW (na przykład katalog `C:\AsixApp\Fabryka_WWW` udostępnić jako `Fabryka_WWW`),
- Jeżeli użytkownicy aplikacji nie mają skonfigurowanego prawa logowania do serwera

Windows, to należy w zabezpieczeniach obu katalogów wirtualnych włączyć dostęp anonimowy.

Po wykonaniu powyższych kroków aplikacja jest dostępna lokalnie po wpisaniu w przeglądarce IE6 adresu:
`http://localhost/<nazwa katalogu wirtualnego aplikacji>`

Dla przykładowej aplikacji adres ten brzmiałby `http://localhost/Fabryka_WWW`.

Aplikacja jest dostępna w sieci lokalnej pod adresem

`http://<nazwa komputera w systemie Windows>/<nazwa katalogu wirtualnego aplikacji>`

Na przykład dla komputera o nazwie `AsixWeb` adresem aplikacji jest:

`http://AsixWeb/Fabryka_WWW`

5.6 Dodatkowe informacje dotyczące konwersji aplikacji systemu asix na aplikację internetową

Wytyczne dotyczące konwertowania aplikacji systemu **asix** do aplikacji internetowej.

Ze względu na techniczne i logiczne problemy mogące się pojawić w wyniku konwersji aplikacji systemu **asix** do aplikacji WWW zaleca się:

1. Unikać stosowania obiektów SELEKTOR – *obiekt pełni funkcje sterującą i nie podlega konwersji*
2. Obiekt LINIE zastąpić obiektem RUROCIĄG – *obiekt LINIE nie podlega konwersji*
3. Obiekty ELIPSY i PROSTOKĄTY należy zastąpić obiektem typu RUROCIĄG – *obiekty EIPSY i PROSTOKĄTY nie podlegają konwersji*
4. Obiekt statyczny nie może znajdować się na obiekcie dynamicznym – *podczas konwersji obiekty dynamiczne umieszczane są na obiektach statycznych.*
5. Przed generowaniem aplikacji Web należy zapewnić, aby obiekty WYKRES nie były zgrupowane – *grupowanie może zablokować konwersję obiektów WYKRES.*
6. Podczas generowania aplikacji internetowej na maszynie wirtualnej należy pozostawić maszynę w aktywnym oknie, nie wolno jej minimalizować ani przysłać innymi programami – *tworzenie masek statycznych wykorzystuje mechanizm Print Screen, który w takiej sytuacji zapisuje niepoprawny wygląd ekranów, masek.*
7. W aplikacji **asix** nie należy stosować plików graficznych o nazwach zawierających sekwencje „+.” (czyli plików *.*.*) – znaki „+.” interpretowane są przez IIS 7.0 jako sekwencja podwójnego anulowania generując błąd (*Błąd HTTP 404.11 — Not Found*).
8. Dla systemów operacyjnych Windows Vista, Windows 7, Windows 2008 oraz nowszych w celu poprawnego wygenerowania aplikacji WWW należy uruchomić Architekta jako administrator – *uprawnienia administratora są niezbędne w celu prawidłowego wykonania etapu konwersji opisanego jako: „Utwórz witrynę WWW. Tworzy stronę startową, tworzy katalog wirtualny aplikacji WWW.”*

Szczegółowe informacje dotyczące poszczególnych akcji i obiektów:

AKCJE

Akcje nieobsługiwane bez ostrzeżenia:

- SKRYPT
- STERUJ_BITY
- NIC

Akcje nieobsługiwane z ostrzeżeniem:

- UKRYJ_WSZYSTKO
- OPIS_ZMIENNEJ
- URUCHOM
- WYŚLIJ_STEROWANIA
- TABELA
- RAPORTUJ
- ASTREND

OBIEKTY

1. Zasady ogólne:

- brak obsługi cieniowania tekstu
- brak obsługi migotania tekstu i obrazków
- brak obsługi sterowania

2. Obiekty nieobsługiwane:

- PRZELICZNIK
- WYŁĄCZNIK
- PRZEŁĄCZNIK

3. Obiekty nieobsługiwane w przypadku gdy są dynamiczne:

- ELIPSY
- LINIE, LINIE PP
- PROSTOKĄTY

4. Ograniczenie w obsłudze pozostałych obiektów:

SŁUPEK

- obiekt musi być parametryzowany z bazy zmiennych
- brak obsługi konturu

PRZYCISK

- brak obsługi skrótów klawiszowych
- brak obsługi kolorów krawędzi oświetlonej i zacienionej dla pozycji wciśnięty i normalny
- brak obsługi parametru zaokrąglenie
- brak obsługi parametru otoczka
- brak obsługi parametru grubość ramki

WYKRES

- kierunki osi (tylko EN)
- opis wykresu (tylko Data-Czas-Wartość)
- brak obsługi krok osi X, krok osi Y, liczby podprzedziałów osi
- brak obsługi kursorów
- rodzaj wykresu (tylko liniowy)
- parametry linii (tylko kolor i tylko ciągła o grubości 1x1)
- brak obsługi znaczników punktów
- parametryzacja dynamiczna
- brak obsługi formatów legendy
- typ krzywej (tylko bieżący)
- wstępne przeliczanie (tylko 'brak' i 100%)

DATA+CZAS

- czas może być wyświetlany tylko w dwóch opcjach – z sekundami lub bez
- data może być wyświetlana tylko w dwóch opcjach DD.MM.RR lub DD.Miesiac.RRRR

Druga forma jest używana tylko jeśli jest zaznaczona opcja miesiąc słownie i nie jest zaznaczona opcja skrócony rok

- parametr duże litery nie jest obsługiwany
- parametr skrócony miesiąc nie jest obsługiwany
- parametr nieznaczące zera nie jest obsługiwany
- parametr łącznik daty nie jest obsługiwany

LICZBA

- obiekt musi być parametryzowany z bazy zmiennych
- brak obsługi wyrównania w pionie jest tylko wyśrodkowanie

TEKST

- brak obsługi napisów pionowych
- brak obsługi odstępów między wierszami

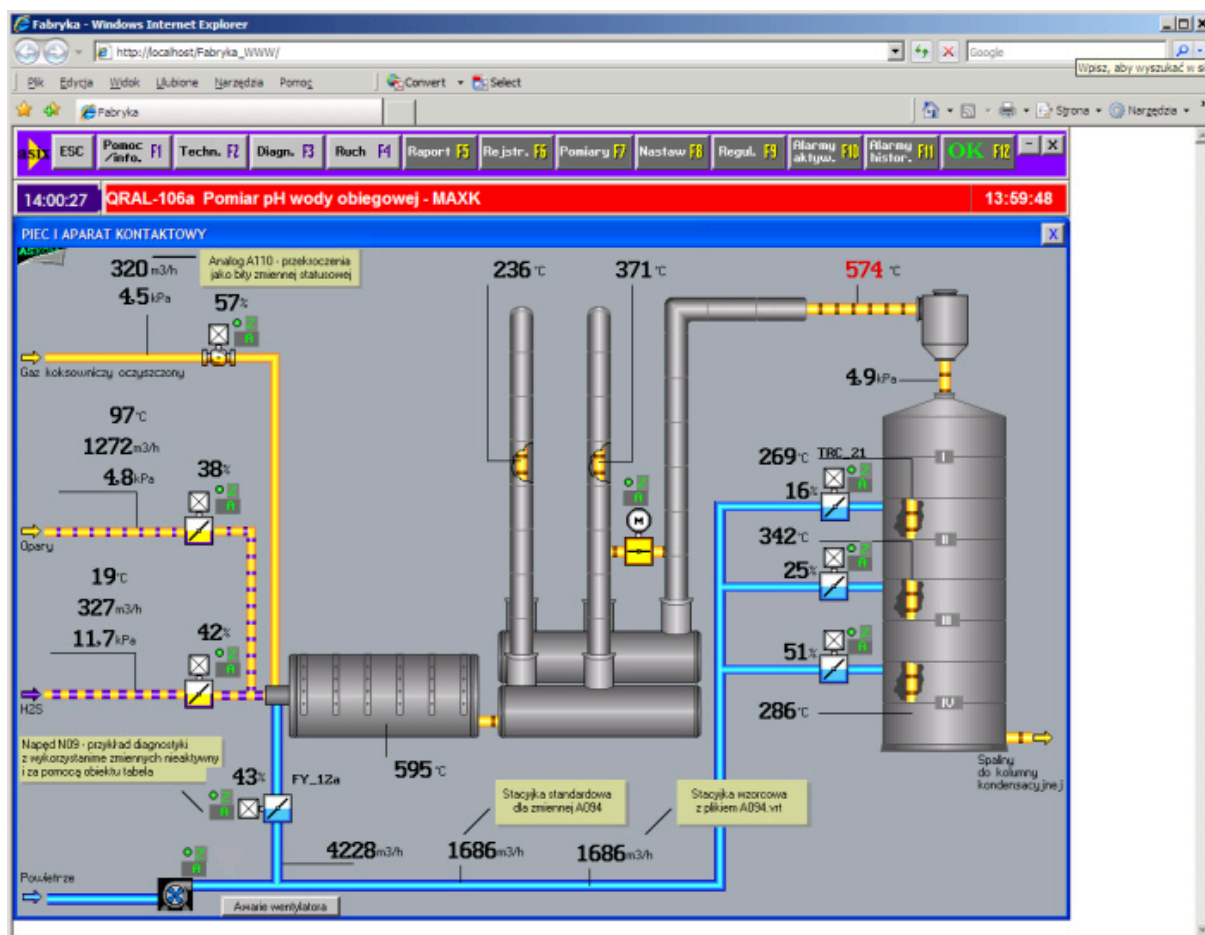
TEKSTY

- brak obsługi napisów pionowych
- brak obsługi odstępów między wierszami

6 As2HTML

6.1 As2HTML

Pakiet As2HTML umożliwia tworzenie dynamicznych stron HTML zawierających dane procesowe z aplikacji systemu **asix**.



Rysunek: Widok dynamicznej strony WWW z danymi procesowymi z aplikacji systemu **asix**.

Pakiet As2HTML do prawidłowej pracy wymaga użycia po stronie klienta przeglądarki Internet Explorer 6 / 7.

Poniższa tabela zawiera listę dostępnych modułów pakietu:

Nazwa modułu	Opis
XConnect.htc	<p>Moduł działa w przeglądarce internetowej po stronie klienta. Jego działanie polega na okresowym pobieraniu z aplikacji systemu asix informacji o danych bieżących oraz o alarmach aktywnych. Dane te są następnie wpisywane do obiektów na stronie HTML, zgodnie z deklaracjami projektanta.</p> <p>Dane z aplikacji systemu asix pobierane są za pośrednictwem serwera <i>Web Service</i>.</p>

	Moduł obsługuje również obiekty LICZBA.
<i>XConnectAL.htc</i> <i>XConnectBar.htc</i> <i>XConnectButton.htc</i> <i>XConnectChart.htc</i> <i>XConnectPictures.htc</i> <i>XConnectText.htc</i> <i>XConnectTexts.htc</i> <i>XConnectWatch.htc</i>	Moduły obsługują obiekty ALARM, SŁUPEK, PRZYCISK, OBRAZKI, TEKST, TEKSTY i ZEGAR.
<i>XConnectChart.htc</i>	<p>Moduł obsługuje obiekt WYKRES.</p> <p>Moduł działa w przeglądarce internetowej po stronie klienta. Działanie modułu polega na pobraniu danych historycznych i wpisaniu ich do obiektu <i>ActiveX</i>, przedstawiającego wykres na stronie w przeglądarce internetowej. Obiekt <i>ActiveX</i> o nazwie <i>TeeChart</i> dostarczany jest w cenie pakietu <i>AsixConnect Server</i>.</p> <p>Dane historyczne mogą być pobierane z aplikacji systemu asix i dostarczone do modułu <i>XConnectHT.htc</i> na jeden z trzech sposobów:</p> <ul style="list-style-type: none"> • Przeczytane jednorazowo po stronie serwera i włączone do strony HTML jako wyspa danych XML (ang. <i>XML Data Island</i>). Dane mogą być przeczytane ponownie po odświeżeniu strony lub wykonaniu operacji <i>postback</i>. • Przeczytane jednorazowo po stronie klienta za pośrednictwem serwera <i>Web Service</i> i następnie automatycznie uzupełniane tą samą drogą o nowe dane – „żywy” wykres. • Przeczytane jednorazowo po stronie klienta za pośrednictwem serwera <i>Web Service</i>. Dane mogą być przeczytane ponownie po odświeżeniu strony lub wywołaniu odpowiednich funkcji modułu <i>XConnectHT.htc</i>.
<i>XConnect.css</i>	Moduł zawiera pomocnicze style CSS pakietu <i>AsDHTML</i> .
<i>Number.css</i> , <i>Bar.css</i> , <i>Alarm.css</i> , <i>Button.css</i> , <i>Text.css</i> , <i>Texts.css</i>	Moduły zawiera style CSS określające domyślny wygląd obiektów LICZBA, SŁUPEK, ALARM, PRZYCISK, TEKST I TEKSTY
<i>XConnect.js</i>	Moduł zawiera stałe wykorzystywane przy definiowaniu obiektów SŁUPEK, WYKRES i PRZYCISK
<i>XConnectNetCS.dll</i>	Moduł ułatwiający przygotowanie po stronie serwera internetowego danych dla wykresów.
<i>webservice.htc</i>	Moduł autorstwa firmy Microsoft. Moduł ten umożliwia dostęp do usług serwerów typu <i>WebService</i> z poziomu skryptu na stronie HTML w przeglądarce internetowej.

6.2 Przygotowanie katalogu projektu

W katalogu projektu należy założyć katalog *bin* i do katalogu skopiować plik *C:\Program Files\Askom\Asix\XConnectNetCS.dll*.

Aby udostępnić katalog w sieci, należy uruchomić program *Internetowe usługi informacyjne*. Program znajduje się w menu *Start/Panel sterowania/Narzędzia administracyjne*. W oknie programu należy podświetlić element *Domyślna witryna sieci Web*, z menu *Akcja/Nowy* wybrać polecenie *Katalog wirtualny*. Uruchamia się wizard, w którym jako *Alias* należy podać nazwę, pod jaką katalog ma być widoczny w Internecie, a jako *Katalog* pełną ścieżkę do katalogu projektu. Pozostałe opcje należy pozostawić bez zmian.

Domyślnie przy próbach dostępu do serwera używana jest autentyfikacja Windows. Aby włączyć dostęp anonimowy należy podświetlić nowo utworzony katalog wirtualny, z menu *Akcja* wybrać polecenie *Właściwości*, wybrać zakładkę *Zabezpieczenia katalogów*, w polu *Dostęp anonimowy* nacisnąć przycisk *Edytuj* i włączyć opcję *Dostęp anonimowy*.

6.3 Przygotowanie projektu

Obiekt WYKRES może być umieszczany tylko na stronach *aspx*. Pozostałe obiekty mogą być umieszczane na stronach *aspx* lub *html*.

Przygotowanie aplikacji Visual Studio 2003, która ma korzystać z jednego z obiektów, polega na dodaniu do projektu aplikacji i plików odwołań aplikacji do modułów pakietu AsixConnect Server.

Projekt i pliki źródłowe aplikacji

Punkt dotyczy tylko aplikacji, które korzystają z obiektu WYKRES.

Po wygenerowaniu projektu należy:

Podświetlić w drzewie projektu folder *References*,

Z menu *Project* wybrać polecenie *Add Reference*,

Nacisnąć przycisk *Browse* i z podkatalogu *C:\Program Files\Askom\Asix* wybrać pliki *XConnectNet.dll* i *XConnectNetCS.dll*,

Nacisnąć przycisk *OK* i jeszcze raz nacisnąć przycisk *OK*, aby zamknąć okno *Add Reference*.

W każdym pliku z kodem źródłowym C# skojarzonym ze stroną *aspx* należy w regionie deklaracji *using* dodać linię:

```
using XConnectNet;
```

Sekcja HEAD

Obowiązkowe elementy w sekcji HEAD pliku HTML:

```
<LINK href="/WebService/Visualization/XConnect.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Number.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Bar.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Button.css" type="text/css" rel="stylesheet">
<script src="/WebService/Visualization/XConnect.js"></script>
<script>
    function init()
    {
        // Region 1 - inicjalizacja obiektow LICZBA, SLUPEK i innych

        xConnectWS.start (2);

        // Region 2 - inicjalizacja obiektow WYKRES
    }
</script>
```

Skrypt *init* będzie wykonany po załadowaniu strony. W skrypcie tym przeprowadzona zostanie inicjalizacja strony.

Funkcji *start* służy do nawiązania połączenia z serwerem *Web Service* i rozpoczęcia procesu odświeżania danych na stronie. Deklaracja funkcji *start* jest następująca:

```
[JScript]
function start (refreshPeriodS);
```

Jako parametr *refreshPeriodS* należy podać okres odświeżania danych bieżących na stronie. Jednostką okresu odświeżania jest sekunda.

Element *BODY*

W elemencie *BODY* należy umieścić atrybut *onload* odwołujący się do skryptu *init*:

```
<body onload="init()">
```

Sekcja *BODY*

Obowiązkowy element w sekcji *BODY* pliku HTML:

```
<DIV id="xConnectWS">AsixConnect</DIV>
```

Obiekt *DIV* o nazwie *xConnectWS*, służy do załadowania modułu *XConnect.htc* i udostępnienia jego funkcji. Moduł *XConnect.htc* jest ładowany w wyniku nadania atrybutowi *id* wartości *xConnectWS*.

Źródło danych

Po przygotowaniu projektu i plików projektu należy skonfigurować źródło danych *Web Service*, co jest opisane w rozdziałach *Instalacja i Plik konfiguracyjny Web.Config*.

6.4 Obiekt KOMUNIKATY

Obiekt *KOMUNIKATY* umożliwia wyświetlenie jednego z wielu wyspecyfikowanych komunikatów tekstowych.

Wybrany do wyświetlenia komunikat zależy od wartości zmiennej monitorowanej. Aby pokazać obiekt na stronie *aspx/html* tekst, należy:

- Utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* - na przykład:

```
<DIV id="KW_A112_Messages"></DIV>
```

- W skrypcie inicjującym *init*, za pomocą funkcji *div2messagess* powiązać obiekt *DIV* ze zmienną:

```
xConnectWS.div2messages (window.document.all.KW_A112_Messages, "KW_A112",
    ['*', 310, 311, [300, 309]],
    ['Tekst niezdefiniowany', 'Tekst 1', 'Tekst 2', 'Tekst 3']);
```

```
[JScript]
function div2messages(element, itemID, numbersArray, textsArray, userStyles)
```

Funkcja *div2messages* służy do przekształcenia obiektu *DIV* w obiekt *KOMUNIKATY*.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.T_11* zwraca referencję do obiektu o identyfikatorze *T_11*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być użyta jako numer komunikatu do wyświetlenia.

Jako parametr *numbersArray* należy podać tablicę numerów komunikatów. Jako jeden element tablicy można podać:

- Liczbę – numer komunikatu.
- Parę liczb – zakres numerów komunikatów. Parę liczb należy przesłać jako podtablicę.
- Znak specjalny '*'. Znak ten reprezentuje komunikat domyślny.

Jako parametr *textsArray* należy podać tablicę komunikatów, które mają być wyświetlane w obiekcie. Dla każdego numeru komunikatu (lub każdego zakresu numerów) w tablicy *numbersArray* należy podać w tablicy *textsArray* jeden komunikat.

Jako parametr *userStyles* należy podać tablicę nazw stylów używanych do formatowania komunikatów. Dla każdego numeru komunikatu (lub każdego zakresu numerów) w tablicy *numbersArray* należy podać jedną nazwę stylu w tablicy *userStyles*. Przy błędnie komunikacyjnym wyświetlany tekst jest przekreślany.

6.5 Obiekt LICZBA

Obiekt LICZBA służy do wyświetlania w postaci liczbowej wartości zmiennej z aplikacji systemu **asix**. Aby pokazać na stronie *aspx/html* wartość zmiennej, należy:

- utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* - na przykład:

```
<DIV id="KW_A084"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *div2number* powiązać obiekt *DIV* ze zmienną:

```
xConnectWS.div2number (window.document.all.KW_A084, "KW_A084");
```

```
[JScript]
function div2number (element, itemID, styles, values);
```

Funkcja *div2number* służy do przekształcenia obiektu *DIV* w obiekt LICZBA.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.KW_A108* zwraca referencje do obiektu o identyfikatorze *KW_A108*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być wyświetlana przez element *DIV*. Parametrem *itemID* może być tylko identyfikator oznaczający bieżącą wartość zmiennej.

Parametry *styles* i *values* są parametrami opcjonalnymi i służą do modyfikacji wyglądu i działania obiektu LICZBA. Ich opis znajduje się w rozdziale *Modyfikowanie wyglądu obiektów LICZBA i SŁUPEK*.

Szczegóły dotyczące obiektu LICZBA znajdują się w *dokumentacji asix, rozdz. OBIEKTY*.

6.6 Obiekt SŁUPEK

Obiekt SŁUPEK służy do wyświetlania w postaci słupka wartości zmiennej z aplikacji systemu **asix**. Aby pokazać na stronie *aspx/html* obiekt SŁUPEK, należy:

- utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* na przykład:

```
<DIV id="KW_A084"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *div2bar* powiązać obiekt *DIV* ze zmienną:

```
xConnectWS.div2bar (window.document.all.KW_A084, "KW_A084", xConnectWS.BarDirN);
```

```
[JScript]
function div2bar (element, itemID, flags, styles, values);
```

Funkcja *div2bar* służy do przekształcenia obiektu *DIV* w obiekt *SŁUPEK*.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.KW_A108* zwraca referencje do obiektu o identyfikatorze *KW_A108*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być wyświetlana przez element *DIV*. Parametrem *itemID* może być tylko identyfikator oznaczający bieżącą wartość zmiennej.

Parametr *flags* służy do określenia wyglądu słupka i powinien on być sumą logiczną następujących flag:

Tabela: Flagi wykorzystywane do parametryzacji wyglądu słupka.

Nazwa flagi	Opis
<i>BarDirE</i>	Kierunek wzrostu słupka. Odpowiednio: E (w prawo), W (w lewo), S (w dół) oraz N (w górę). Należy użyć jednej z tych flag.
<i>BarDirW</i>	
<i>BarDirS</i>	
<i>BarDirN</i>	
<i>BarUseBase</i>	Należy użyć tej flagi, jeżeli słupek ma mieć inną bazę niż minimum wartości zmiennej. W bazie zmiennych musi być zdefiniowany atrybut określający wartość bazy słupka.
<i>BarHiLimit</i>	Użycie tej flagi powoduje dodanie do słupka poprzecznej linii określającej wartość górnego limitu ostrzegawczego.
<i>BarLoLimit</i>	Użycie tej flagi powoduje dodanie do słupka poprzecznej linii określającej wartość dolnego limitu ostrzegawczego.
<i>BarHiHiLimit</i>	Użycie tej flagi powoduje dodanie do słupka poprzecznej linii określającej wartość górnego limitu alarmowego.
<i>BarLoLoLimit</i>	Użycie tej flagi powoduje dodanie do słupka poprzecznej linii określającej wartość dolnego limitu alarmowego.
<i>BarAllLimits</i>	Użycie tej flagi powoduje dodanie do słupka poprzecznych linii określających wartość wszystkich limitów.

Parametry *styles* i *values* są parametrami opcjonalnymi i służą do modyfikacji wyglądu i działania obiektu. Ich opis znajduje się w rozdziale *Modyfikowanie wyglądu obiektów LICZBA i SŁUPEK*.

Szczegóły dotyczące obiektu *SŁUPEK* znajdują się w *dokumentacji asix, rozdz. SŁUPEK*.

6.7 Obiekt PRZYCISK

Obiekt PRZYCISK służy do wyświetlania elementu sterującego, umożliwiającego zamykanie i otwieranie masek aplikacji internetowej. Używany jest głównie przez moduł As2WWW przy konwersji masek aplikacji systemu **asix** na aplikację internetową, ale może też być użyty we własnych aplikacjach internetowych.

Aby wyświetlić na stronie *aspx/html* obiekt PRZYCISK, należy:

- utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* na przykład:

```
<DIV id=" BUTTON_10"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *div2button* przekształcić obiekt *DIV* w przycisk:

```
xConnectWS.div2button (
  window.document.all.BUTTON_10,
  'Diagn.',
  ['img/F3C.png'],
  Button3D + ButtonLeftImage,
  function() {xConnectWS.closeAllNotBlockedMask ('');
    xConnectWS.openNewMask ('xkwdiag.aspx', 0, 87, 1024, 681, 0); },
  ['BUTTON_10_Text_Normal', 'BUTTON_10_Text_Inset', 'BUTTON_10_Text_Outset',
  'Button_Frame_Normal', 'Button_Frame_Inset', 'Button_Frame_Outset']);

[JavaScript]
function div2button (element, text, images, userFlags, userFunction, userStyles);
```

Funkcja *div2button* służy do przekształcenia obiektu *DIV* w obiekt PRZYCISK.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.BUTTON_10* zwraca referencje do obiektu o identyfikatorze *BUTTON_10*.

Jako parametr *text* należy podać tekst, który ma być wyświetlany na przycisku.

Jako parametr *images* należy podać tablicę nazw plików graficznych, które mają być wyświetlane przez przycisk. Pierwszy plik jest wyświetlany, gdy przycisk jest w pozycji *wyciśnięty*, drugi w pozycji *plaski*, a trzeci w pozycji *wciśnięty*. Drugi i/lub trzeci pik może być pominięty i wtedy wyspecyfikowane pliki zastępują opuszczone.

Parametr *flags* służy do określenia wyglądu przycisku i powinien on być sumą logiczną następujących flag:

Tabela: Flagi do określania wyglądu przycisku.

Nazwa flagi	Opis
<i>Button3D</i>	Przycisk wykorzystuje elementy graficzne sprawiające wrażenie obecności trzeciego wymiaru w wyglądzie obiektu.
<i>Button3States</i>	Bez tej flagi przycisk wykorzystuje dwa obrazki wyświetlane w stanie <i>wyciśniętym</i> i <i>wciśniętym</i> . Użycie tej flagi powoduje wykorzystanie trzeciego obrazka wyświetlanego w przypadku, gdy nad przyciskiem nie znajduje się kursor myszy.
<i>ButtonLeftImage</i>	Użycie tej flagi powoduje wyświetlenie obrazka z lewej strony przycisku a tekstu z prawej.
<i>ButtonTransparent</i>	Użycie tej flagi powoduje, że przycisk nie jest widoczny, ale reaguje na naciśnięciu przycisku myszy i wykonuje funkcję użytkownika. Flaga ta jest zwykle wykorzystywana w przycisku nakładanym na obiekt LICZBA i mającym za zadanie wyświetlenie stacyjki.

Parametry `userFunction` służy do przekazania funkcji, która będzie wykonana w reakcji na naciśnięcie przycisku. Jako parametr można przekazać referencję do funkcji globalnej lub wprost funkcję anonimową języka JavaScript, jak w przykładzie powyżej.

W funkcji użytkownika można korzystać z funkcji do zarządzania maskami udostępnianymi przez obiekt `xConnectWS`.

Tabela: Funkcje do zarządzania maskami udostępnianymi przez obiekt `xConnectWS`.

Nazwa funkcji	Opis
<code>openNewMask</code>	<p>Składnia wywołania funkcji:</p> <pre>function openNewMask (maskName, x, y, w, h, blokada)</pre> <p>Parametry funkcji to odpowiednio nazwa otwieranej strony <code>aspx/html</code>, położenie strony i jej rozmiar oraz parametr logiczny określający, czy maska jest zablokowana.</p> <p>Otwierana strona jest nakładana na stronę aktualnie załadowaną przy pomocy elementu HTML o nazwie <code>IFRAME</code>.</p>
<code>closeMask</code>	<p>Składnia wywołania funkcji:</p> <pre>function closeMask (maskNameMask)</pre> <p>Parametrem jest nazwa strony <code>aspx/html</code>, która ma zostać zamknięta. W nazwie można używać znaków <code>*</code> i <code>?</code>, aby zamknąć naraz wiele stron.</p> <p>Funkcja <code>closeMask</code> zamyka również strony zablokowane.</p>
<code>closeAllNotBlockedMask</code>	<p>Składnia wywołania funkcji:</p> <pre>function closeAllNotBlockedMask (exceptThisMask)</pre> <p>Funkcja zamyka wszystkie otwarte strony, które nie są zablokowane. Jako parametr można podać nazwę strony, która ma pozostać otwarta.</p>

Parametr `userStyles` służy do modyfikowania wyglądu obiektu. Jako jego wartość należy przekazać tablicę sześcioelementową, zawierającą nazwy stylów stosowanych do określenia wyglądu przycisku i wyglądu ramki przycisku.

6.8 Obiekt TEKST

Obiekt TEKST służy do wyświetlania wartości atrybutu zmiennej z aplikacji systemu **asix**. Aby pokazać na stronie *aspx/html* wartość atrybutu zmiennej, należy:

- utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* - na przykład:

```
<DIV id="KW_A084_Opis"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *div2text* powiązać obiekt *DIV* ze atrybutem zmiennej:

```
xConnectWS.div2text (window.document.all.KW_A084_Opis, "KW_A084", "Opis");

[JavaScript]
function div2text (element, itemID, attributeName, styles, values)
```

Funkcja *div2text* służy do przekształcenia obiektu *DIV* w obiekt TEKST.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.KW_A108_Opis* zwraca referencje do obiektu o identyfikatorze *KW_A108_Opis*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być wyświetlana przez element *DIV*. Parametrem *itemID* może być tylko identyfikator oznaczający bieżącą wartość zmiennej.

Jako parametr *attributeName* należy podać nazwę atrybutu zmiennej procesowej.

Parametry *styles* i *values* są parametrami opcjonalnymi i służą do modyfikacji wyglądu i działania obiektu TEKST.

6.9 Obiekt OBRAZKI

Obiekt OBRAZKI umożliwia wyświetlenie jednej z wielu wyspecyfikowanych map bitowych. Wybrany do wyświetlenia obrazek zależy od wartości zmiennej monitorowanej. Aby pokazać na stronie *aspx/html* wartość zmiennej, należy:

- utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* - na przykład:

```
<DIV id="PICUTRES_104"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *div2number* powiązać obiekt *DIV* ze zmienną:

```
xConnectWS.div2pictures (window.document.all.PICTURES_104, "KW_N11", 'BINARY', 8, 4,
['img/KL_WY.png', 'img/KL_H_Y.png', 'img/KL_Z.png', 'img/KL_A.png']);

[JavaScript]
function div2pictures(element, itemID, coding, firstBit, statesCount, pictArray,
bitMasks)
```

Funkcja *div2pictures* służy do przekształcenia obiektu *DIV* w obiekt OBRAZKI.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.PICTURES_104* zwraca referencje do obiektu o identyfikatorze *PICTURES_104*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być użyta przy wyznaczaniu bitmapy do wyświetlenia.

Jako parametr *coding* należy podać tekst określający sposób kodowania. Można użyć jednego z napisów: "BINARY", "NATURAL" lub "UNDEFINED";

Jako parametr *firstBit* należy podać numer pierwszego (najmłodszego) bitu danej monitorowanej.

Jako parametr *statesCount* należy podać liczbę stanów obiektu.

Jako parametr *pictArray* należy podać tablicę nazw bitmap, które mają być wyświetlane w obiekcie OBRAZKI. Jeżeli liczba bitmap jest równa liczbie stanów obiektu to w przypadku błędów komunikacyjnych na tle obiektu rysowane jest czerwone przekreślenie. Jeżeli liczba bitamp jest o jedno większa od liczby stanów, to w przypadku błędów komunikacyjnych wyświetlana jest ostatnia bitmapa.

Parametr *bitMasks* jest używany tylko przy kodowaniu dowolnym (parametr *coding* równy "UNDEFINED"). Jako jego wartość należy podać tablicę tekstów zawierający mapy bitowe poszczególnych stanów. Szczegóły opisane są w dokumentacji systemu *asix* w rozdziale dotyczącym obiektu OBRAZKI.

6.10 Obiekt TEKSTY

Obiekt TEKSTY umożliwia wyświetlenie jednego z wielu wyspecyfikowanych tekstów. Wybrany do wyświetlenia tekst zależy od wartości zmiennej monitorowanej. Aby pokazać na stronie *aspx/html* tekst, należy:

- Utworzyć na stronie element *DIV* zawierający zdefiniowany atrybut *id* - na przykład:

```
<DIV id="TEXTS_11"></DIV>
```

- W skrypcie inicjującym *init*, za pomocą funkcji *div2texts* powiązać obiekt *DIV* ze zmienną:

```
xConnectWS.div2texts (window.document.all.TEXTS_11, "KW_B000", 'NATURAL', 3, 2,
['F31', 'F31'], ['TEXTS_11_0', 'TEXTS_11_1']);

[JavaScript]
function div2texts (element, itemID, coding, firstBit, statesCount, textsArray,
userStyles, bitMasks)
```

Funkcja *div2texts* służy do przekształcenia obiektu *DIV* w obiekt TEKSTY.

Jako parametr *element* należy przesłać referencję obiektu *DIV*, który ma pokazywać wartość zmiennej. Przykładowo, wyrażenie *window.document.all.TEXTS_11* zwraca referencje do obiektu o identyfikatorze *TEXTS_11*.

Jako parametr *itemID* należy podać nazwę zmiennej procesowej, której wartość ma być użyta przy wyznaczaniu tekstu do wyświetlenia.

Jako parametr *coding* należy podać tekst określający sposób kodowania. Można użyć jednego z napisów: "BINARY", "NATURAL" lub "UNDEFINED";

Jako parametr *firstBit* należy podać numer pierwszego (najmłodszego) bitu danej monitorowanej.

Jako parametr *statesCount* należy podać liczbę stanów obiektu.

Jako parametr *textsArray* należy podać tablicę tekstów, które mają być wyświetlane w obiekcie TEKSTY.

Jako parametr *userStyles* należy podać tablicę nazw stylów używaną do formatowania obiektu, gdy wartość zmiennej jest dostępna i gdy występuje błąd komunikacyjny. Przy błędzie komunikacyjnym dodatkowo wyświetlany tekst jest zawsze przekreślany.

Parametr *bitMasks* jest używany tylko przy kodowaniu dowolnym (parametr *coding* równy "UNDEFINED"). Jako jego wartość należy podać tablicę tekstów zawierających mapy bitowe poszczególnych stanów. Szczegóły opisane są w dokumentacji systemu *asix* w rozdziale dotyczącym obiektu TEKSTY.

6.11 Obiekt ALARM

Aby pokazać na stronie *html* lub *aspx* wartość ostatniego aktywnego alarmu aplikacji systemu **asix**, należy:

- utworzyć na stronie HTML dwa elementy *DIV* zawierające zdefiniowany atrybut *id*:

```
<DIV id="LastAlarm"></DIV> <DIV id="LastAlarmDateTime"></DIV>
```

- w skrypcie inicjującym *init*, za pomocą funkcji *divs2alarms* powiązać obiekty *DIV* z systemem alarmów:

```
xConnectWS. divs2alarms (window.document.all.LastAlarm,
                        window.document.all.LastAlarmDateTime,
                        true, false, true, true);
```

```
[JScript]
function divs2alarms (alarmTextBoundElement,
                    alarmTimeBoundElement,
                    showTime,
                    showTimeShort,
                    showDate,
                    showDateShort);
```

Funkcja *divs2alarms* umożliwia wyświetlanie informacji o ostatnim, niepotwierdzonym alarmie aktywny w dwu obiektach *DIV* na stronie HTML. Pierwszy element *DIV* służy do wyświetlania tekstu alarmu, a drugi do wyświetlania stempla czasu alarmu. Pozostałe parametry funkcji *divs2alarms* służą do określenia sposobu wyświetlania stempla czasu alarmu.

Do formatowania wyglądu obiektów *DIV* używane są style zdefiniowane w pliku *Alarm.css*. Styl *Alarm_QualityGood* używany jest, gdy działa komunikacja z serwerem alarmów, a *Alarm_QualityBad*, gdy wystąpił błąd komunikacji.

Aby zmienić wygląd obiektu ALARM należy utworzyć dwa własne style o nazwach *Alarm_QualityGood* i *Alarm_QualityBad* w nowym pliku CSS i włączyć ten plik do strony *aspx/html*.

```
<LINK href="MyStyles.css" type="text/css" rel="stylesheet">
```

6.12 Obiekt ZEGAR

Obiekt ZEGAR służy do wyświetla na stronie *html/aspx* daty i/lub czasu lokalnego. Aby umieścić na stronie obiekt ZEGAR należy wstawić na stronę element *DIV*, w którym zostanie umieszczony zegar. Element *DIV* nie może zawierać żadnych elementów wewnętrznych. Do wstawienia elementu *DIV* można użyć komponentu *FlowLayoutPanel* z palety HTML; komponent ten jest renderowany na stronie jako element *DIV*. Element musi mieć zdefiniowany atrybut *class*:

```
class=" XConnectWatch"
```

PRZYKŁAD

```
<DIV class="XConnectWatch"></DIV>
```

6.13 Obiekt WYKRES

6.13.1 Deklaracja wykresu

Obiekt WYKRES służy do wyświetlenia w postaci wykresu wartości archiwalnych zmiennych procesowych z aplikacji systemu **asix**. Obsługiwany są tylko strony *aspx*, ponieważ część tworzenia wykresu związana z włączeniem projektu wykresu do strony internetowej może odbywać się tylko na serwerze.

Obiekt WYKRES może pracować w dwóch wariantach:

1. Wykres statyczny generowany po stronie serwera.
2. Wykres generowany po stronie klienta.

Wykres statyczny generowany po stronie serwera służy do wyświetlenia wykresu z dowolnego okresu czasu. Dane historyczne są czytane z wybranego okresu czasu i wstawiane na stronę internetową w postaci wyspy danych XML (ang. XML Data Island). Dane te są później używane przez skrypt po stronie klienta do stworzenia wykresu. Po wyświetleniu wykresu nie jest on już automatycznie modyfikowany. W większości przypadków ten rodzaj wykresu jest generowany najszybciej. W celu przesuwania okresu wykresu można utworzyć na stronie przyciski, powiązać je procedurami obsługi po stronie serwera i w procedurach tych oprogramować pobieranie nowych danych historycznych.

Wykres generowany po stronie klienta służy do wyświetlenia wykresu danych historycznych z okresu obejmującego chwilę bieżącą i pewien odcinek czasu wstecz. Automatycznie na końcu wykresu dodawane są nowe dane, które pojawiają się w archiwum aplikacji systemu **asix**. Zastępowane przez nie dane z początku wykresu są usuwane. Strona pobierana jest z serwera raz, a odświeżaniem danych zajmuje się skrypt pracujący po stronie klienta.

Wykres generowany po stronie klienta służy również do wyświetlenia wykresu z dowolnego okresu czasu. Dane historyczne wyspecyfikowanych zmiennych są czytane po stronie klienta już po załadowaniu strony. Po wyświetleniu wykresu nie jest on już automatycznie modyfikowany. W celu przesuwania okresu wykresu można utworzyć na stronie przyciski, powiązać je procedurami obsługi po stronie klienta i w procedurach tych oprogramować przesuwanie okresu wykresu.

Niezależnie od rodzaju wykresu zawsze należy wykonać następujące kroki:

1. Przygotować projekt wykresu i zapisać go w pliku; plik umieścić w katalogu aplikacji.
2. Wstawić na stronę *aspx* element *DIV*, w którym zostanie umieszczony wykres.

Element *DIV* nie może zawierać żadnych elementów wewnętrznych. Element musi mieć zdefiniowane następujące atrybuty:

```
- runat="server"
- class="XConnectChart"
- id=<identyfikator>
```

PRZYKŁAD

```
<DIV runat="server" id="Chart1" class="XConnectChart" style="Z-INDEX:
101; LEFT: 16px; WIDTH: 768px; POSITION: absolute; TOP: 16px;
HEIGHT: 192px" ms_positioning="FlowLayout"></DIV>
```

3. W pliku kodu należy dodać deklarację pola klasy:

```
protected System.Web.UI.HtmlControls.HtmlGenericControl Chart1;
```

4. W pliku kodu, w funkcji *Page_Load*, należy dodać linię

```
ChartXMLData.InsertChartHTML ("chart.tee", Chart1);
```

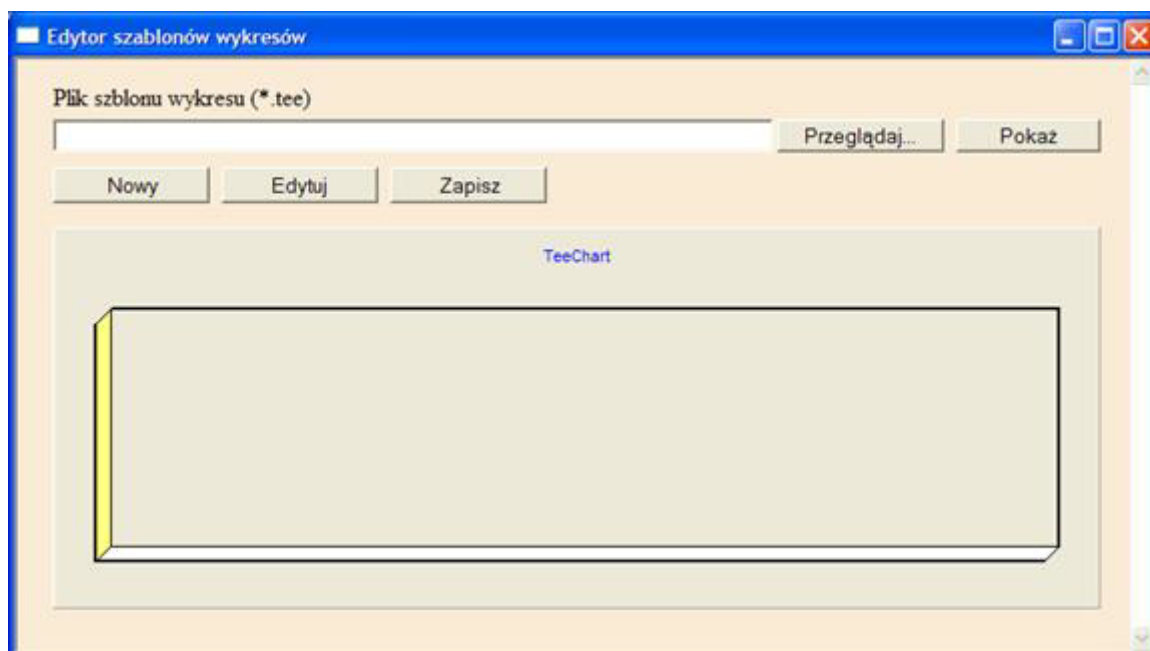
Zamiast nazwy pliku projektu wykresu *chart.tee* należy użyć nazwę pliku utworzonego w punkcie 1.

Dalsze kroki zostały opisane przy poszczególnych rodzajach wykresów.

6.13.2 Projekt wykresu

Do projektowania wykresów służy aplikacja *ChartDesigner.hta*. Podstawowym elementem projektu wykresu są serie. Do projektu wykresu należy dodać jedną serię dla każdej zmiennej z aplikacji systemu **asix**, której wykres chcemy wyświetlić. Jako typ serii należy użyć *Liniowy*, *Słupkowy* lub *Powierzchniowy*. Pozostałe wykresy XY mogą działać, ale nie były testowane. Nie można używać wykresów XYZ. Oprócz serii zaleca się sparametryzowanie osi wykresu i legendy.

Po uruchomieniu aplikacji *ChartDesigner.hta* pojawia się okno główne:

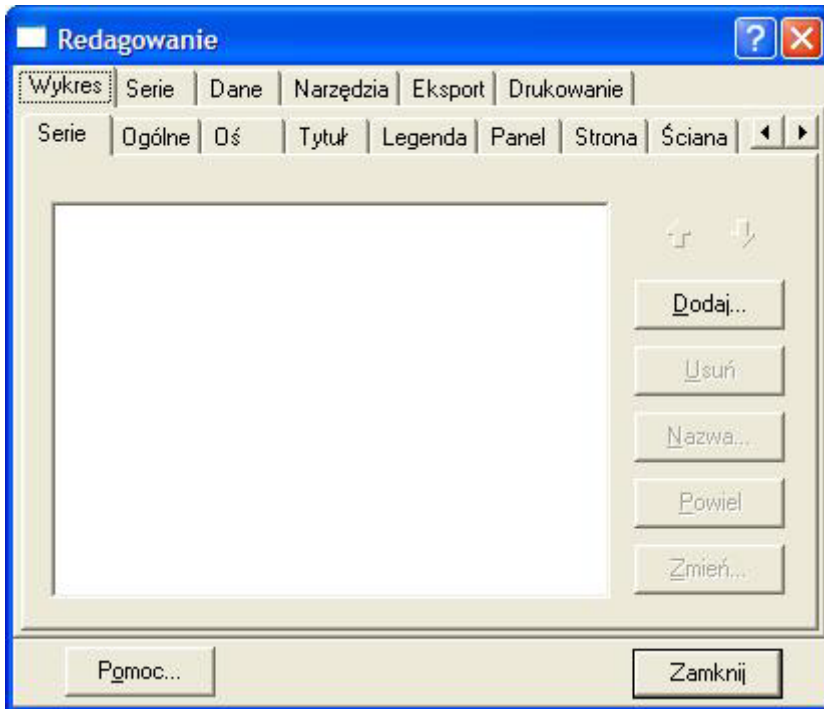


Rysunek: Edytor szablonów wykresów - okno główne.

Opis elementów w oknie:

- | | |
|------------------------------|---|
| <i>Plik szablonu wykresu</i> | - w polu tym należy podać nazwę pliku projektu wykresu. Nazwę można wpisać lub wybrać naciskając przycisk <i>Przełączaj</i> . |
| <i>Przełączaj</i> | - wyświetlenie okna pozwalającego wybrać nazwę pliku projektu wykresu. |
| <i>Pokaż</i> | - załadowanie pliku projektu wykresu o nazwie znajdującej się w polu <i>Plik szablonu wykresu</i> . |
| <i>Nowy</i> | - utworzenie nowego projektu wykresu. |
| <i>Edytuj</i> | - edycja aktualnego projektu wykresu – uruchamiany jest edytor projektu. Po zakończeniu edycji wykres należy zapisać w pliku – plik ten będzie używany razem ze stronami <i>aspx</i> do wyświetlenia wykresu. |
| <i>Zapisz</i> | - zapisanie stworzonego projektu w pliku. |

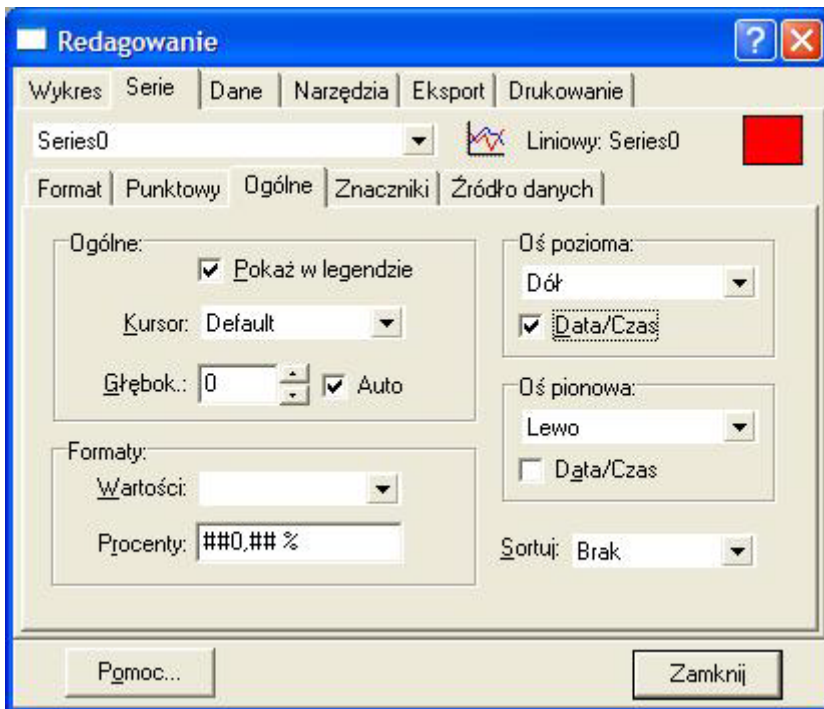
Po naciśnięciu przycisku *Edytuj* uruchamiany jest edytor projektu.



Rysunek: Edytor szablonów wykresów - okno edycyjne.

Aby dodać serię do wykresu należy nacisnąć przycisk *Dodaj* i wybrać wykres *Liniowy*, *Słupkowy* lub *Powierzchniowy*.

Po utworzeniu każdej serii należy wybrać zakładkę *Serie*, wybrać utworzoną serię, wybrać zakładkę *Ogólne* i w grupie *Oś pozioma* zaznaczyć pole wyboru *Data/Czas*.



Rysunek: Edytor szablonów wykresów - okno edycyjne serii.

Pozostałe opcje serii umożliwiają dopasowanie wyglądu wykresu do potrzeb projektanta.

6.13.3 Wykres statyczny generowany po stronie serwera

Wykres statyczny generowany po stronie serwera

Procedura umieszczania wykresu na stronie *aspx*:

1. Przygotować stronę – patrz: [6.12.1 Deklaracja wykresu](#).
2. Do strony *aspx* dodać komponent typu *WebControl* o nazwie *Literal* i nadać mu identyfikator. W przykładzie będzie używany identyfikator *XMLIsland1*.
3. W pliku kodu, w funkcji *Page_Load*, dodać linie:

```
ChartXMLData chartXMLData = new ChartXMLData();
chartXMLData.Add ("K11_Para", Aggregate.Start, "K11_Para",
    HTReadFlags.Default, "1M");
chartXMLData.SetChartPeriod "DAY-24H", "24H");
chartXMLData.ReadData (XMLIsland1);
Session.Add ("chartXMLData", chartXMLData);
```

4. Na stronie *aspx*, w funkcji *init*, w miejscu oznaczonym *Region2* wstawić linię:

```
window.document.all.Chart1.div2chartXML (null, "XMLIsland1");
```

Fragment programu w języku C# zamieszczony w punkcie 5 wykonuje następujące operacje:

- Tworzy obiekt klasy *ChartXMLData*, którego zadaniem jest pobranie danych archiwalnych i umieszczenie ich na stronie.
- Za pomocą funkcji *Add* dodaje do obiektu jedną zmienną *K11_Para*. Pozostałe parametry określają identyfikator agregatu, nazwę zmiennej w legendzie, flagi odczytu oraz okres próbkowania.
- Za pomocą funkcji *SetChartPeriod* ustawia początek okresu, z którego mają być pobrane dane oraz długość tego okresu.
- Za pomocą funkcji *ReadData* czyta dane archiwalne i umieszcza je w obiekcie *XMLIsland1*.
- Za pomocą funkcji *InsertChartHTML* wstawia definicję wykresu do wnętrza obiektu *Chart1*. W linii tej zamiast nazw pliku projektu wykresu *chart.tee* należy użyć nazwy pliku utworzonego w punkcie 1.
- Wstawia obiekt *chartXMLData* do pamięci podręcznej *Session* aplikacji ASP.NET, aby móc wykorzystać go przy kolejnych wywołaniach ze strony użytkownika przy obsłudze wywołań *postback*. Na przykład, można umieścić na stronie przycisk, którego zadaniem będzie przesunięcie okresu wykresu do przodu o godzinę. Kod wykonujący taką operację jest następujący:

```
ChartXMLData chartXMLData =
    (ChartXMLData)Session["chartXMLData"];
chartXMLData.MoveChartPeriod ("1H");
chartXMLData.ReadData (XMLIsland1);
```

Funkcja *div2chartXML*

```
[JScript]
function div2chartXML (teeFile, xmlDataIslandName)
```

Funkcja *div2chartXML* jest używana do utworzenia w elemencie *DIV* obiektu WYKRES, dla którego dane zostały przygotowane po stronie serwera i przekazane jako część strony *aspx*.

Jako parametr *teeFile* należy przekazać wartość *undefined*, a jako parametr *xmlDataIslandName* identyfikator obiektu typu *Literal*, do którego wstawiono dane po stronie serwera.

6.13.4 Wykres generowany po stronie klienta

Wykres generowany po stronie klienta

Procedura umieszczania wykresu na stronie *aspx*:

1. Przygotować stronę – patrz: [6.12.1 Deklaracja wykresu](#).
2. Na stronie *aspx*, w funkcji *init*, poniżej miejsca oznaczonego *Region2* wstawić linię:

```
window.document.all.Chart1.div2chart(
    undefined, ["KW_A000"], ["Start"], ["KW_A000"], 3600, 60);
```

Zamiast nazwy zmiennej *KW_A000* należy podać właściwą nazwę zmiennej. Zamiast nazwy atrybutu *Start* można użyć nazwy innego atrybutu.

3. Jeżeli na stronie ma się pojawić wykres żywy, to następnie należy wstawić linię zawierającą wywołanie funkcji *goLive*:

```
window.document.all.Chart1.goLive (5);
```

W przeciwnym wypadku, gdy na stronie ma się pojawić wykres z danymi z przeszłości, to należy wstawić linię zawierającą wywołanie funkcji *moveChartPeriodTo*:

```
window.document.all.Chart1.moveChartPeriodTo("DAY");
```

Funkcja div2chart

```
[JScript]
function div2chart (
    teeFile,
    itemNames,
    aggregateNames,
    itemLegends,
    periodLen,
    resampleInterval,
    readFlags)
```

Funkcja *div2chart* jest używana do utworzenia w elemencie DIV obiektu WYKRES.

Parametr *teeFile* w wersji finalnej aplikacji musi mieć wartość *undefined* i wykres musi być ładowany, zgodnie z opisem, w funkcji *Page_Load*. Podczas testów aplikacji jako wartość parametru *teeFile* można podać nazwę pliku projektu wykresu i odłożyć modyfikowanie funkcji *Page_Load*.

Jako parametr *itemNames* należy podać tablicę tekstów nazw zmiennych. Można użyć skróconego zapisu do tworzenia tablic języka *JScript*:

```
["Nazwa1", "Nazwa2", "Nazwa2"]
```

Jako parametr *aggregateNames* należy podać tablicę nazw agregatów zmiennych.

Jako parametr *itemLegends* należy podać tablicę tekstów, które mają się pojawić w legendzie wykresu i reprezentować poszczególne zmienne.

Jako parametr *periodLen* należy podać długość okresu wykresu. Długość można wyrazić w sekundach lub używając czasu względnego.

Jako parametr *resampleInterval* należy podać długość interwału agregatu. Długość można wyrazić w sekundach lub używając czasu względnego.

Parametr *readFlags* jest opcjonalny i został opisany w rozdziale [6.12.6 Flagi ReadFlags](#) oraz w dokumentacji *AsixConnect4*, rozdz. *Flagi ReadFlags*. W skrypcie `init` dostępne są dwie stałe, które można użyć jako wartość tego parametru:

```
ReadFlags_AddPointAtStartOfBadQualities
ReadFlags_AddPointAtEndOfBadQualities
```

Pierwszej stałej należy używać w systemach operujących na danych chwilowych. Drugiej można używać w systemach operujących na średnich (czyli na przykład w systemach kontroli emisji spalin).

Funkcja `addSeries`

```
[JScript]
function addLiveSeries(
    seriesSerialNumbers,
    itemNames,
    aggregateNames,
    itemLegends,
    periodLen,
    resampleInterval,
    readFlags)
```

Funkcja *addSeries* umożliwia umieszczenie na jednym wykresie serii różniących się okresem wyświetlania lub długością interwału agregatu. Chcąc skorzystać z tej możliwości należy najpierw wywołać funkcję *div2chart* bez żadnych parametrów lub przekazując w parametrze *itemNames* nazwy mniejszej liczby zmiennych niż liczba serii zdefiniowanych w pliku projektu wykresu. Następnie wywołując funkcję *addSeries* należy dodać pozostałe zmienne do wykresu.

Jako parametr *seriesSerialNumber* należy podać tablicę liczb oznaczających numery serii w pliku projektu serii, do których mają być przypisane zmienne z drugiego parametru funkcji *itemNames*.

Pozostałe parametry mają takie same znaczenie jak parametry funkcji *div2chart*.

Funkcja `goLive`

```
[JScript]
function goLive (
    refreshIntervals)
```

Funkcja *goLive* służy do przełączenia wykresu w tryb żywy, czyli wyświetlania ostatnio zarejestrowanych danych i periodycznego odświeżania tych danych.

Jako *refreshIntervals* należy podać okres odświeżania wykresu w sekundach.

Funkcja `moveChartPeriodTo`

```
function moveChartPeriodTo (
    periodStart)
```

Funkcja *moveChartPeriodTo* służy do ustalenia nowego początku danych na wykresie. Jeżeli wykres znajduje się w trybie „żywym” to tryb ten wyłączany i wyświetlane są dane historyczne

Jako parametr *periodStart* należy podać początek okresu czasu, z którego mają być pobrane dane. Datę można podać jako datę względną w postaci tekstu w formacie OPC, jako datę bezwzględną w postaci obiektu *Date*, lub jako datę bezwzględną w postaci tekstowej.

Funkcja `moveChartPeriodBy`

```
function moveChartPeriodBy (
    seconds)
```

Funkcja `moveChartPeriodBy` służy do przesunięcia początku okresu wykresu. Parametr `seconds` określa o ile sekund ma się przesunąć początek wykresu. Wartość dodatnie oznaczają przesunięcie w przód a wartości ujemne w tył.

6.13.5 Klasa `ChartXMLData`

Klasa `ChartXMLData` ułatwia tworzenie stron internetowych zawierających wykresy.

Funkcja `InsertChartHTML`

```
[C#]
static public void InsertChartHTML(
    String fileName,
    HtmlGenericControl div);
```

Funkcja statyczna używana we wszystkich rodzajach wykresów. Służy do wstawienia do wnętrza elementu `DIV` kodu HTML, który zawiera:

- Definicję obiektu HTML wyświetlającego wykres.
- Opis wykresu przeczytany z pliku projektu wykresu.

Jako parametr `fileName` należy podać nazwę pliku zawierającą definicję wykresu; w nazwie pliku można użyć ścieżki względnej. Jako parametr `div` należy podać obiekt reprezentujący obiekt `DIV` znajdujący się na stronie.

Funkcja `Add`

```
[C#]
public void Add(
    String itemID, Aggregate aggregate, String itemLegend, ReadFlags readFlags)
public void Add(
    String itemID, Aggregate aggregate, String itemLegend, ReadFlags readFlags,
    TimeSpan resampleInterval)
public void Add(
    String itemID, Aggregate aggregate, String itemLegend, ReadFlags readFlags,
    String resampleInterval)
```

Funkcja `Add` służy do dodania do tworzonego wykresu kolejnych zmiennych. Funkcję `Add` należy wywołać tyle razy, ile serii utworzono w pliku projektu wykresu.

Parametr `itemID` określa nazwę zmiennej, której próbki mają znaleźć się na wykresie. Parametr `aggregate` określa agregat, który ma być wyliczony z danych archiwalnych. Najczęściej używany jest agregat `Aggregate.Start` umożliwiający równomierne rozmieszczenie punktów na wykresie.

Jako parametr `itemLegend` należy podać tekst, który w legendzie wykresu będzie określał zmienną. Często parametr ten jest taki sam jak `itemID`.

Parametr `readFlags` został opisany w [6.12.6 Flagi `ReadFlags`](#).

Przy obliczaniu agregatu okres czasu dzielony jest na interwały o długości podawanej w parametrze `resampleInterval`. Agregat o identyfikatorze podanym w parametrze `aggregate` jest obliczany dla każdego interwału przy wykorzystaniu danych zarchiwowanych w czasie trwania tego interwału. Przyjmuje się, że każdy interwał jest przedziałem zamkniętym lewostronnie i otwartym prawostronnie. Wartość parametru `resampleInterval` może być podana jako obiekt typu `TimeSpan`

lub jak tekst zawierający długość interwału w formacie OPC. Jeżeli parametr *resampleInterval* jest opuszczony to wtedy przyjmuje się, że jest on równy okresowi archiwizacji zmiennej.

Funkcja SetChartPeriod

```
[C#]
public void SetChartPeriod(
    DateTime periodStart, TimeSpan periodLen, TimeSpan resampleInterval)
public void SetChartPeriod(
    DateTime periodStart, TimeSpan periodLen)
public void SetChartPeriod(
    DateTime periodStart)

public void SetChartPeriod(
    String periodStart, String periodLen, String resampleInterval)
public void SetChartPeriod(
    String periodStart, String periodLen)
public void SetChartPeriod(
    String periodStart)
```

Funkcja *SetChartPeriod* określa okres, z którego mają być przeczytane dane archiwalne.

Parametr *periodStart* jest obowiązkowy i określa początek okresu. Może być podany w postaci obiektu klasy *DateTime* (wartość bezwzględna czasu) lub w postaci tekstu zawierającego datę względną w formacie OPC.

Parametr *periodLen* określa długość okresu. Również może być podany jako wartość bezwzględna lub względna.

Parametr *resampleInterval* określa długość interwału agregacji. Jeżeli zostanie użyty to modyfikowany jest interwał wszystkich zmiennych wyświetlanych na wykresie.

Przy pierwszym wywołaniu funkcji *SetChartPeriod* parametry *dateTimeFrom* i *periodLen* są obowiązkowe. Przy następnych można podać tylko parametr *dateTimeFrom*, dzięki czemu uzyskujemy efekt zmiany początku wykresu bez modyfikacji pozostałych parametrów.

Funkcja MoveChartPeriod

```
[C#]
public void MoveChartPeriod (int period)
public void MoveChartPeriod (TimeSpan period)
public void MoveChartPeriod (String period)
```

Funkcja *MoveChartPeriod* pozwala na przesunięcie początku wykresu w przód (parametr *period* większy od zera) lub w tył (parametr *period* mniejszy od zera). Jako wartość parametru *period* można podać liczbę sekund, obiekt typu *TimeSpan* lub tekst zawierający długość okresu czasu w formacie OPC.

Funkcja ReadData

```
[C#]
public void ReadData (System.Web.UI.WebControls.Literal XMLIsland)
```

Funkcja *ReadData* czyta dane historyczne wyspecyfikowanych zmiennych z wybranego okresu czasu i wstawia te dane na stronę HTML w postaci wyspy danych XML (ang. XML Data Island). Dane te są później używane przez skrypt po stronie klienta do utworzenia wykresu.

6.13.6 Flagi ReadFlags

Przeczytane próbki mogą być dodatkowo przetworzone dla ułatwienia tworzenia z nich wykresów. Do określenia sposobu przetwarzania służy parametr typu ReadFlags. Parametr ten powinien być sumą logiczną stałych opisanych poniżej. Jeżeli nie jest potrzebne żadne specjalne przetwarzania danych to należy użyć flagi ReadFlags.Default.

Aproksymacja wartości danych o jakości złej

Przy braku aproksymacji punkt o jakości złej ma pole value wypełnione wartością zero. Jeżeli użyta zostanie flaga ReadFlags.LinearAproximationOfValueOfBadQuality to takiemu punktowi nadana zostanie wartości najbliższego wcześniejszego punktu o jakości dobrej (jakość dalej pozostaje zła!). Jeżeli nie ma punktu wcześniejszego o dobrej jakości to użyta zostanie wartość jakiegokolwiek punktu o jakości dobrej z bieżącego okresu lub napotkanego podczas poprzednich odczytów tej samej zmiennej.

Dodanie punktu na granicy obszaru próbek o jakości złej

Możliwe jest dodawanie próbek na początku obszaru próbek o jakości złej lub na końcu. Można użyć jednej z flag ReadFlags.AddPointAtStartOfBadQualities lub ReadFlags.AddPointAtEndOfBadQualities lub też żadnej z nich.

Dodanie jednej próbki na prawej krawędzi obszaru

Dodanie jednej próbki na prawej krawędzi czytanego obszaru danych archiwalnych pozwala wypełnić wykresem cały wyspecyfikowany okres. W tym celu należy użyć flagi ReadFlags.AddPointAtRightBound.

Przykładowo, jeżeli czytamy dane godzinowe za jeden dzień, to otrzymamy 25 próbek poczynając od północy dnia bieżącego do północy dnia następnego.

Użycie flag w skryptach JScript

Aby umożliwić użycie stałych ReadFlags w języku JScript zdefiniowano w pliku XConnect.js stałe odpowiadające stałym ReadFlags języka C#.

Tabela: Stałe ReadFlags w języku JScript odpowiadające stałym ReadFlags języka C#.

Nazwa flagi w języku C#	Nazwa flagi w języku JScript
ReadFlags.Default	ReadFlags_Default
ReadFlags.AddPointAtStartOfBadQualities	ReadFlags_AddPointAtStartOfBadQualities
ReadFlags.AddPointAtEndOfBadQualities	ReadFlags_AddPointAtEndOfBadQualities
ReadFlags.AddPointAtRightBound	ReadFlags_AddPointAtRightBound

6.14 Modyfikowanie obiektów LICZBA i SŁUPEK

6.14.1 Stany zmiennej

Do celów modyfikacji obiektów rozróżnia się dziewięć stanów obiektów. Stany opisane są w tabeli poniżej. Stan obiektu określa się na podstawie jakości powiązanej z nim zmiennej.

Tabela: Dziewięć stanów obiektów LICZBA i SŁUPEK określanych na podstawie jakości powiązanej z nim zmiennej.

Nazwa stanu obiektu	Opis stanu obiektu
QualityGood	Jakość zmiennej jest dobra.
QualityGood_HIHi	Jakość zmiennej jest dobra, przekroczone limit górny alarmowy.
QualityGood_Hi	Jakość zmiennej jest dobra, przekroczone limit górny ostrzegawczy.
QualityGood_Lo	Jakość zmiennej jest dobra, przekroczone limit dolny

	ostrzegawczy.
<i>QualityGood_LoLo</i>	Jakość zmiennej jest dobra, przekroczono limit dolny alarmowy.
<i>QualityUncertain</i>	Jakość zmiennej jest niepewna.
<i>QualityBad</i>	Jakość zmiennej jest zła. Stan obejmuje wszystkie przypadki jakości złej z wyjątkiem <i>Last Known</i> i <i>Sensor Failure</i> .
<i>QualityBad_LastKnown</i>	Jakość zmiennej jest zła, wystąpił błąd komunikacji, dostępna jest ostatnia znana wartość zmiennej.
<i>QualityBad_SensorFailure</i>	Jakość zmiennej jest zła z powodu uszkodzenia czujnika. Stan sygnalizowany przez zmienną kontrolną lub przez drajwer urządzenia.

Podczas modyfikacji obiektu często wygląd czy zachowanie obiektu jest takie samo przy różnych stanach obiektów. W takim przypadku można pominąć definicje niektórych stanów i wtedy zamiast stanu pominiętego zostanie użyty inny stan zgodnie z tabelą poniżej. Definicje stanów *QualityGood* i *QualityBad* są zawsze wymagane.

Tabela: Zestawienie: nazwa stanu pominiętego - nazwa stanu zastępującego dla obiektów LICZBA i SŁUPEK.

Nazwa stanu pominiętego	Nazwa stanu zastępującego
<i>QualityGood_HiHi</i>	<i>QualityGood_Hi</i>
<i>QualityGood_Hi</i>	<i>QualityGood</i>
<i>QualityGood</i>	-
<i>QualityGood_Lo</i>	<i>QualityGood</i>
<i>QualityGood_LoLo</i>	<i>QualityGood_Lo</i>
<i>QualityUncertain</i>	<i>QualityBad_LastKnown</i>
<i>QualityBad</i>	-
<i>QualityBad_LastKnown</i>	<i>QualityBad</i>
<i>QualityBad_SensorFailure</i>	<i>QualityBad</i>

6.14.2 Modyfikowanie wyglądu obiektów LICZBA i SŁUPEK

Dla każdego stanu obiektu zdefiniowany jest styl CSS określający wygląd obiektu *DIV*. Style te są zdefiniowane w plikach *Number.css* i *Bar.css*.

Tabela: Style CSS obiektu LICZBA i SŁUPEK.

Nazwa stanu obiektu	Nazwa domyślnego stylu obiektu LICZBA	Nazwa domyślnego stylu obiektu SŁUPEK
<i>QualityGood</i>	<i>Number_QualityGood</i>	<i>Bar_QualityGood</i>
<i>QualityGood_HiHi</i>	<i>Number_QualityGood_HiHi</i>	<i>Bar_QualityGood_HiHi</i>
<i>QualityGood_Hi</i>	<i>Number_QualityGood_Hi</i>	<i>Bar_QualityGood_Hi</i>
<i>QualityGood_Lo</i>	<i>Number_QualityGood_Lo</i>	<i>Bar_QualityGood_Lo</i>
<i>QualityGood_LoLo</i>	<i>Number_QualityGood_LoLo</i>	<i>Bar_QualityGood_LoLo</i>
<i>QualityUncertain</i>	<i>Number_QualityUncertain</i>	<i>Bar_QualityUncertain</i>

<i>QualityBad</i>	<i>Number_QualityBad</i>	<i>Bar_QualityBad</i>
<i>QualityBad_LastKnown</i>	<i>Number_QualityBad_LastKnown</i>	<i>Bar_QualityBad_LastKnown</i>
<i>QualityBad_SensorFailure</i>	<i>Number_QualityBad_SensorFailure</i>	<i>Bar_QualityBad_SensorFailure</i>

Wygląd obiektów można zmodyfikować tworząc własny zestaw stylów w nowym pliku CSS i włączając go do strony *aspx/html* po plikach *css* pakietu *AsixConnect* lub zamiast nich.

```
<LINK href="/WebService/Visualization/XConnect.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Number.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Bar.css" type="text/css" rel="stylesheet">
<LINK href="/WebService/Visualization/Alarm.css" type="text/css" rel="stylesheet">
<LINK href="MyStyles.css" type="text/css" rel="stylesheet">
```

Nazwy stylów w nowym pliku CSS mogą być:

1. Niezmienione – należy wtedy zastąpić deklaracje użycia plików *css* pakietu *AsixConnect* deklaracjami użycia własnych plików *css*; zmiany wyglądu obejmą wszystkie obiekty na stronie,
2. Nowe – należy wtedy dodać deklaracje użycia własnych plików *css* po deklaracjach użycia plików *css* pakietu *AsixConnect*. Zmiany wyglądu obejmą wtedy tylko te obiekty, którym w fazie inicjalizacji przekazano nazwy nowych stylów. Nowe style mogą bazować na stylach zdefiniowanych w plikach *css* pakietu *AsixConnect* i modyfikować tylko niektóre atrybuty i/lub dodawać nowe.

Aby przekazać nazwy nowych stylów do obiektu *LICZBA* lub *SŁUPEK* należy w skrypcie *init* utworzyć obiekt zawierający nazwy nowych stylów umieszczone w odpowiednich własnościach i przekazać ten obiekt jako parametr *styles* funkcji *div2number* lub *div2bar*. Nazwy własności odpowiadające poszczególnym stanom są takie same jak nazwy stanów.

PRZYKŁAD

Przykład przekazania nowego zestawu stylów do obiektu *LICZBA*:

```
var engineNumberStyle = new Object;
engineNumberStyle.QualityGood = "Engine_QualityGood";
engineNumberStyle.QualityGood_HiHi = "Engine_QualityGood_HiHi";
engineNumberStyle.QualityGood_Hi = "Engine_QualityGood_Hi";
engineNumberStyle.QualityGood_Lo = "Engine_QualityGood_Lo";
engineNumberStyle.QualityGood_LoLo = "Engine_QualityGood_LoLo";
engineNumberStyle.QualityUncertain = "Engine_QualityUncertain";
engineNumberStyle.QualityBad = "Engine_QualityBad";
engineNumberStyle.QualityBad_LastKnown = "Engine_QualityBad_LastKnown";
engineNumberStyle.QualityBad_SensorFailure = "Engine_QualityBad_SensorFailure";
xConnectWS.div2number (window.document.all.KW_A084, "KW_A084", engineNumberStyle);
```

Do formatowania wyglądu obiektów *SŁUPEK* używane są dwa dodatkowe style zdefiniowane w pliku *Bar.css*.

Tabela: Dodatkowe style obiektu *SŁUPEK*.

Nazwa stylu	Opis stosowania stylu	Nazwa własności obiektu przy przekazywaniu stylu
<i>Bar_Background</i>	Styl służy do formatowania tła obiektu.	<i>Background</i>
<i>Bar_Marker</i>	Styl służy do formatowania poprzecznych linii pokazujących wartości limitów ostrzegawczych i alarmowych.	<i>Marker</i>

6.14.3 Modyfikowanie działania obiektu LICZBA

Dla każdego stanu obiektu LICZBA zdefiniowana jest funkcja, która na podstawie informacji o stanie zmiennej zwraca wartość i wartość ta jest wyświetlana w obiekcie *DIV*. Jeżeli dla jednego ze stanów funkcja zwracałaby zawsze tę samą wartość to zamiast funkcji można bezpośrednio użyć tą wartość.

Moduł XConnect.htc po pobraniu z aplikacji systemu **asix** nowej wartości zmiennej określa stan zmiennej. Jeżeli dla tego stanu jest zdefiniowana stała wartość to jest ona wpisywana do obiektu *DIV*. Jeżeli dla stanu jest zdefiniowana funkcja, to funkcja ta jest wywoływana i jej wynik jest wpisywany do obiektu *DIV*. Jako parametr funkcji przesyłany jest obiekt zawierający stan zmiennej.

Obiekt ten posiada następujące pola:

Tabela: Zestawienie pól obiektu będącego parametrem funkcji określającej stan obiektu LICZBA.

Nazwa własności	Opis
<i>succeeded</i>	Pole zawiera wartość <i>true</i> , gdy udało się odczytać stan zmiennej i ustawione są pola <i>timeStamp</i> , <i>quality</i> i <i>dataValues</i> . Pole zawiera wartość <i>false</i> w przeciwnym wypadku.
<i>errorString</i>	Jeżeli pole <i>succeeded</i> zawiera wartość <i>false</i> to pole <i>errorString</i> zawiera tekstowy opis błędu.
<i>timeStamp</i>	Stempel czasu wartości zmiennej.
<i>quality</i>	Jakość zmiennej.
<i>dataValues</i>	Wartość zmiennej w postaci tablicy. Gdy zmienna jest zmienną skalarną to tablica jest jednoelementowa.

Domyślne wartości funkcji lub stałych dla poszczególnych stanów zmiennej dla obiektu LICZBA są następujące:

Tabela: Domyślne wartości funkcji lub stałych dla poszczególnych stanów zmiennej dla obiektu LICZBA.

Nazwa stanu	Kod domyślnej funkcji lub stałej
<i>QualityGood</i>	<code>function (itemState) { return itemState.dataValues[0]; }</code>
<i>QualityBad</i>	<code>"?"</code>
<i>QualityBad_LastKnown</i>	<code>function (itemState) { return itemState.dataValues[0] + "?"; }</code>
<i>QualityBad_SensorFailure</i>	<code>"??"</code>

Dla pozostałych stanów przyjmowane są wartości zgodnie z algorytmem opisanym w rozdziale *Stany zmiennej*.

Aby zmodyfikować działanie obiektu LICZBA należy w skrypcie *init* utworzyć obiekt zawierający stałe lub funkcje dla stanów zmiennej i przekazać ten obiekt jako czwarty parametr funkcji *div2number*. Jako trzeci parametr funkcji *div2number* należy przekazać obiekt z nazwami styli lub stałą *undefined*. Nazwy własności odpowiadające poszczególnym stanom są takie same jak nazwy stanów.

PRZYKŁAD

Przykład modyfikacji działania obiektu LICZBA:

```
var engineNumberText = new Object;

engineNumberText.QualityBad_LastKnown =
    function (itemState) { return "?" + itemState.dataValues[0] + "?"; };

engineNumberText.QualityGood = function (itemState) { return itemState.dataValues[0];
};
```

```

engineNumberText.QualityBad = "?";
engineNumberText.QualityBad_SensorFailure = "???"

xConnectWS.div2number (
    window.document.all.KW_A084, "KW_A084", undefined, engineNumberText);

```

6.15 Wiązanie obiektu DHTML z wartością bieżącą zmiennej

Aby związać własność obiektu DHTML z wartością bieżącą zmiennej należy użyć funkcji *bindProperty*:

```

[JavaScript]
function bindProperty (
    element,
    propertyName,
    itemName,
    values)

```

Jako parametr *element* należy przekazać referencję do obiektu DHTML, którego własność ma być modyfikowana. Przykładowo, wyrażenie *window.document.all.KW_A108* zwraca referencję do obiektu o identyfikatorze *KW_A108*, a wyrażenie *window.document.all.KW_A108.style* zwraca referencję do obiektu zawierającego style obiektu o identyfikatorze *KW_A108*.

Jako parametr *propertyName* należy podać nazwę własności obiektu DHTML, która ma być modyfikowana. Własność ta będzie nazywana dalej własnością aktywną.

Jako parametr *itemName* należy podać nazwę zmiennej procesowej, której wartość ma wpływać na własność aktywną.

Jeżeli wartość zmiennej procesowej ma być bezpośrednio i zawsze przypisana własności aktywnej to jako parametr *values* należy przekazać wartość *null*.

Przetwarzanie wartości zmiennej procesowej przed przypisaniem jej do własności obiektu DHTML

W operacji wiązania dowolnej własności obiektu DHTML z wartością zmiennej procesowej wyróżnia się dziewięć stanów obiektu opisanych w rozdziale *Stany zmiennej*. Dla każdego stanu obiektu należy zdefiniować funkcję, która na podstawie informacji o stanie zmiennej zwraca wartość i wartość ta jest wpisywana do własności obiektu DHTML. Jeżeli dla jednego ze stanów funkcja zwracałaby zawsze tę samą wartość to zamiast funkcji można bezpośrednio użyć tą wartość.

Moduł *XConnect.htc* po pobraniu z aplikacji systemu **asix** nowej wartości zmiennej określa stan obiektu. Jeżeli dla tego stanu jest zdefiniowana stała wartość to jest ona wpisywana do własności obiektu DHTML. Jeżeli dla stanu jest zdefiniowana funkcja to funkcja jest wywoływana i jej wynik jest wpisywany do własności obiektu DHTML. Jako parametr funkcji przesyłany jest obiekt zawierający stan zmiennej.

Obiekt ten posiada następujące pola:

Tabela: Zestawienie pól obiektu będącego parametrem funkcji określającej własność obiektu DHTML.

Nazwa własności	Opis
<i>succeeded</i>	Pole zawiera wartość <i>true</i> , gdy udało się odczytać stan zmiennej i ustawione są pola <i>timeStamp</i> , <i>quality</i> i <i>dataValues</i> . Pole zawiera wartość <i>false</i> w przeciwnym wypadku.
<i>errorString</i>	Jeżeli pole <i>succeeded</i> zawiera wartość <i>false</i> to pole <i>errorString</i> zawiera tekstowy opis błędu.
<i>timeStamp</i>	Stempel czasu wartości zmiennej.
<i>quality</i>	Jakość zmiennej.
<i>dataValues</i>	Wartość zmiennej w postaci tablicy. Gdy zmienna jest zmienną skalarną to tablica jest jednoelementowa.

W skrypcie *init* należy utworzyć obiekt zawierający stałe lub funkcje dla poszczególnych stanów zmiennej i przekazać ten obiekt jako parametr *values* funkcji *bindProperty*. Nazwy własności odpowiadające poszczególnym stanom są takie same jak nazwy stanów.

PRZYKŁAD

Przykład dowiązania własności *left* obiektu DHTML do wartości zmiennej procesowej w wyniku czego obiekt porusza przy zmianach wartości tej zmiennej:

```
var movingNumberPos = new Object;
movingNumberPos.QualityBad = 10; // położenie domyślne
movingNumberPos.QualityGood =
    function (itemState) { itemState.dataValues[0]; };
xConnectWS.bindProperty (
    window.document.all.KW_A084.style, "left", "KW_A084", movingNumberPos);
```