

Visualisierungskomponente

AID 027 DE für ADITO4



© 2014 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: FA, MN, MW, KN. Version 10.1. Zuletzt geändert 04.09.2017

Version	Kommentar
10.1	Anpassung der Formatierungen
10.0	Anpassung an ADITO4.6, Beispiele verändert
5.4	Vorbereitung zur Verwendung des JavaFX-Browsers entfernt. Kapitel ist nicht mehr notwendig mit ADITO 4.4 und 4.5
5.3	Vorversion vor Umstellung auf Versionshistorie im Dokument.

Inhaltsverzeichnis


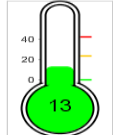



1.	Arbeiten mit der Visualisierungskomponente	4
1.1.	Visualisierungstypen	4
2.	Einbinden der Komponente	6
2.1.	Ablegen der Komponente auf dem Frame	6
2.2.	Einstellen der Eigenschaften	6
2.2.1.	Grundeigenschaften	6
2.2.2.	Setzen der Werte	7
2.2.3.	Diagrammeigenschaften	8
3.	Visualisierung als Diagramm	9
3.1.	Liste der Eigenschaften	10
4.	Visualisierung als Browser	12
4.1.	Übergabe einer URL an den Browser	12
4.2.	Auslesen einer URL aus dem Browser	12
4.3.	Eigenschaften der Browser-Visualisierung	12
5.	Visualisierung als Tachometer/ Thermometer	14
5.1.	Setzen der Werte	14
5.2.	Eigenschaften der Visualisierung als Tachometer/ Thermometer	14
5.3.	Beispiel	15




1. Arbeiten mit der Visualisierungskomponente

Dieses Dokument beschreibt die Nutzung der Visualisierungskomponente, mit der Sie Daten in ADITO online grafisch aufbereiten können.

1.1. Visualisierungstypen

Für die Visualisierung von Daten stehen Ihnen die in der nachfolgenden Tabelle gezeigten Diagrammtypen zur Verfügung.

ADITO4 online Komponente	Diagramm	Beschreibung
MeterChart		Tachometerdiagramm zur Anzeige eines Wertes. Hier können optional verschiedene Schwellenwerte definiert werden, die kritische Bereiche definieren. Der aktuelle Datenwert lässt sich bei Bedarf im Instrument anzeigen.
MeterChart		Thermometerdiagramm zur Anzeige eines Wertes. Hier können optional verschiedene Schwellenwerte definiert werden, die kritische Bereiche definieren. Der aktuelle Datenwert lässt sich bei Bedarf im Instrument anzeigen.
Chart		Ein Balkendiagramm für die Visualisierung von einem oder mehreren Werten in Form von Balken. Diese können optional in 3D-Optik dargestellt werden, der jeweilige Datenwert lässt sich auf dem Balken darstellen und die Grafik kann um 90° gedreht werden.
Chart		Ein Tortendiagramm für die Visualisierung von einem oder mehreren Werten. Diese können optional in 3D-Optik dargestellt werden, der jeweilige Datenwert lässt sich innerhalb des Tortenstücks darstellen.
Chart		Ein Liniendiagramm für die Visualisierung von einem oder mehreren Werten in Form einer Linie, die die einzelnen Datenwerte verbindet. Der jeweilige Datenwert lässt sich auf dem Balken darstellen.

ADITO4 online Komponente	Diagramm	Beschreibung
Chart		<p>Ein Flächendiagramm für die Visualisierung von einem oder mehreren Werten in Form einer Fläche. Diese kann optional in 3D-Optik dargestellt werden, der jeweilige Datenwert lässt sich einblenden und die Grafik kann um 90° gedreht werden.</p>
Chart		<p>Ein Punktediagramm für die Visualisierung von einem oder mehreren Werten in Form von Datenpunkten. Der jeweilige Datenwert lässt sich auf dem Balken darstellen.</p>
Browser		<p>Im Internet-Browser können Sie Websites aufrufen. Verwendet wird (auf Microsoft Windows-Betriebssystemen) die Internet Explorer Rendering Engine.</p>




2. Einbinden der Komponente

2.1. Ablegen der Komponente auf dem Frame

Starten Sie den ADITO4 Designer und erstellen oder öffnen Sie im Fenster Projekte einen Frame, auf dem Sie eine Visualisierungskomponente platzieren möchten.

Klicken Sie in der Palette mit den verfügbaren Komponenten auf eines der Symbole für die Visualisierungskomponenten und ziehen Sie die Komponente durch Anklicken der Ziehpunkte mit der Maus und bei gehaltener Maustaste auf die gewünschte Größe.

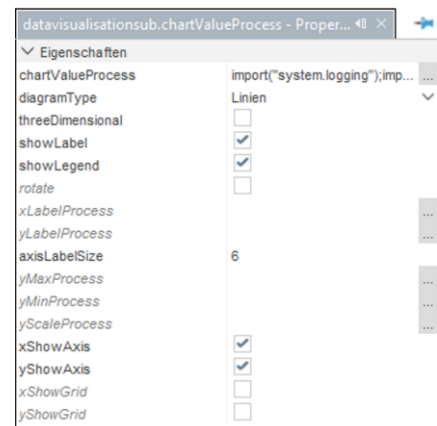
Es stehen in ADITO4 folgende Symbole zur Verfügung:

- Symbol  **Chart**
 - Balken
 - Tortendiagramm
 - Linien
 - Fläche
 - Punkte
- Symbol  **MeterChart**
 - Tachometer
 - Thermometerdiagramm
- Symbol  **Browser**
 - Browser

2.2. Einstellen der Eigenschaften



2.2.1. Grundeigenschaften

Zuerst legen Sie für die Komponente fest, ob die Darstellung als Thermometer/Rundinstrument oder als Wertediagramm gewählt wird. Die Auswahl als Thermometer oder Rundinstrument bewirkt gleichzeitig, dass Sie nur einen einzelnen Datenwert pro Visualisierungskomponente darstellen können. Setzen Sie dazu in den Eigenschaften der Komponente auf dem Reiter mit der Bezeichnung *Datenvisualisierung (dv)* die Eigenschaft *Darstellung* auf den Wert „Diagramm“ oder „Instrumentenanzeige“ (s. Grafik rechts).



Bei Bedarf können Sie nun noch im Reiter *Layout* Eigenschaften wie die Position, die Hintergrundfarbe und andere Standardeigenschaften verändern.



In ADITO4 online in der Visualisierungskomponente  **Chart** ist nur die Darstellung als Wertediagramm möglich. Für das Thermometer oder das Rundinstrument steht die Visualisierungskomponente  **MeterChart** zur Verfügung.

2.2.2. Setzen der Werte

Für die Diagramme vom Typ „Instrument“ wird der anzuzeigende Wert über die Eigenschaft `valueProcess` gesetzt. Der dort enthaltene JDito-Code muss einen Wert zurückliefern, der dann visualisiert wird. Für die Rückgabe stehen Ihnen die normalen Rückgabemethoden zur Verfügung (`result.string()`, `result.object()` und `result.query()`).

Um folgende Anzeige zu erhalten, werden diese Eigenschaften benötigt:



`valueProcess` (Formel):

```
result.string(34);
```

`thresholdsProcess` (Festlegung der Schwellenwerte):


```
import("system.logging");
import("system.result");
var base = 0;
var retArr = [base];

for ( var i = 0; i < 10; i ++ ) {
    base = base + 10;
    retArr.push(base);
}
result.object(retArr);
```

Es würde also auf dem Thermometer bzw. dem Tacho der Wert 34 angezeigt.

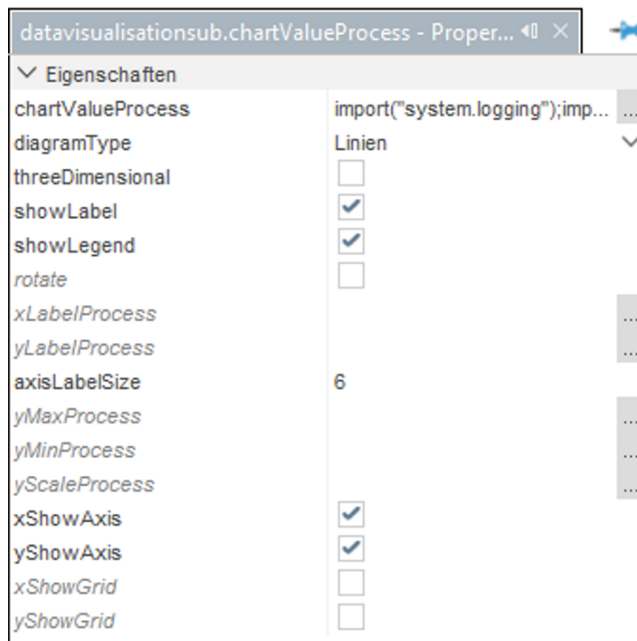
Für die Diagramme vom Typ „Chart“ kann mehr als ein Wert visualisiert werden. Daher erfolgt das Setzen der Werte nicht über die Formel, sondern über eine gesonderte Eigenschaft des Diagramm-Eigenschaftsdialogs. Bitte lesen Sie dazu den nächsten Absatz.



In ADITO4 steht für die Instrumentenanzeige (Thermometer oder Rundinstrument) die Visualisierungskomponente  `MeterChart` zur Verfügung.

2.2.3. Diagrammeigenschaften

Die Diagrammeigenschaften legen Sie im Eigenschaftfenster im Reiter *Andere* fest.



3. Visualisierung als Diagramm

Alle Diagrammeigenschaften werden im Eigenschaftfenster definiert. Der Prozess selbst ist im `chartValueProcess` anzugeben.

Der hier eingetragene Code muss ein Array zurückliefern. Das erste Element ist ein Array mit den Datenwerten und das zweite Element ist ein Array mit den Beschriftungen der Werte (den X-Labels).

Ein sehr einfaches Diagramm lässt sich also mit der folgenden Zeile erzeugen:

```
result.object( [ [ 1, 2, 3, 5, 8, 13 ], ["F1", "F2", "F3", "F4", "F5", "F6"] ] );
```

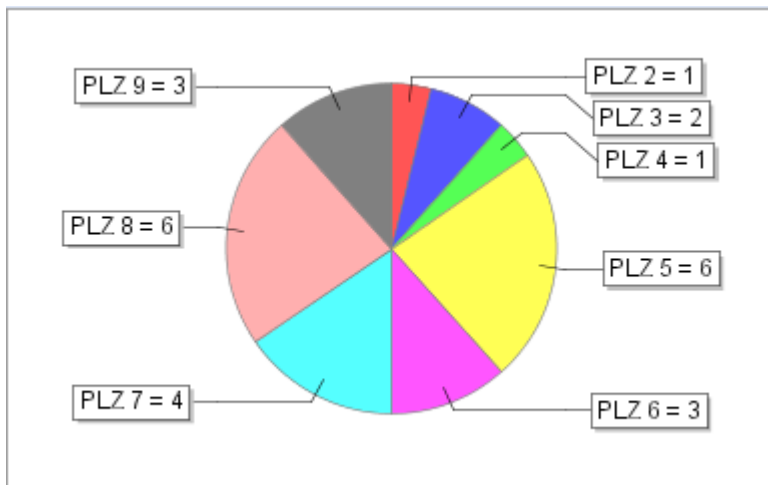
Für die Rückgabe stehen Ihnen die normalen Rückgabemethoden zur Verfügung (`result.string()`, `result.object()` und `result.query()`).

Das folgende Beispiel liefert die Verteilung der Postleitzahlen auf die einzelnen PLZ-Gebiete als Daten- und Beschreibungsarray zurück:

```
import("system.logging");
import("system.result")

var sql = "select substr(zip, 1, 1), count(*) from relation "
        + "join address on address_id = addressid "
        + "where country = 'DE' and zip is not null "
        + "group by substr(zip, 1, 1) ";
var rset = db.table(sql);
var werte = new Array();
var labels = new Array();
for(var i = 0; i < rset.length; i++)
{
    labels.push( "PLZ " + rset[i][0] );
    werte.push( rset[i][1] );
}
var res = new Array();
res[0] = werte;
res[1] = labels;
result.object( res );
```

Die folgende Grafik zeigt das daraus resultierende Diagramm:




3.1. Liste der Eigenschaften

Die nachfolgende Tabelle listet die derzeit verfügbaren Diagrammeigenschaften auf.

Eigenschaft (Bezeichnung ADITO4 online)	Beschreibung
threeDimensional	Stellt die Diagrammelemente dreidimensional dar. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
axisLabelSize	Legt die Größe der Labelbeschriftung fest. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
diagramType	Stellt den Diagrammtyp ein. Verfügbar sind derzeit die Typen BAR (nur ADITO online 3.1), PIE, AREA, LINE, POINT.
chartValueProcess	Definiert den Prozess für die Berechnung der Wertepaare.
rotate	Gibt an, ob das Diagramm um 90° gedreht dargestellt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
showLabel	Legt fest, ob der Datenwert im Diagramm angezeigt werden soll.
showLegend	Zeigt eine Legende für die Daten an. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
xShowAxis	Legt fest, ob die X-Achse angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
xShowGrid	Legt fest, ob das X-Gitter angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.

Eigenschaft (Bezeichnung ADITO4 online)	Beschreibung
yShowAxis	Legt fest, ob die Y-Achse angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
yShowGrid	Legt fest, ob das Y-Gitter angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
xLabelProcess	Legt fest, ob die Beschriftung der X-Achse angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
yLabelProcess	Legt fest, ob die Beschriftung der Y-Achse angezeigt wird. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
yMaxProcess	Definiert den maximalen Wert für die Y-Achse. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
yMinProcess	Definiert den minimalen Wert für die Y-Achse. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.
yScaleProcess	Legt die Skalierung der Y-Achse fest. Diese Eigenschaft wird nicht für alle Diagrammtypen berücksichtigt.

4. Visualisierung als Browser

Die Visualisierungskomponente kann auch als Browser verwendet werden. Hierzu muss die Visualisierungskomponente  `Browser` verwendet werden.

Grundsätzlich wird die Komponente wie eine Visualisierungskomponente behandelt, es sei denn Sie klicken in einer geöffneten Website mit der rechten Maustaste auf einen Link und wählen „in neuem Fenster öffnen“. In diesem Fall öffnet sich ein eigenes ADITO-Fenster als Browser.

Über die Eigenschaft `BrowserType` im Komponenteninspektor kann eingestellt werden, welcher Browser verwendet wird:

- **Native** (Nur Windows): Verwendet den Browser mit der im Betriebssystem integrierten Internet Explorer Rendering Engine (Trident-Engine, abhängig von der Betriebssystemversion). Details zu Trident siehe hier: https://en.wikipedia.org/wiki/Trident_%28layout_engine%29
- **JavaFX** (Alle Systeme): Verwendet den JavaFX Browser, eine Engine basierend auf WebKit. Dieser kann auf allen Betriebssystemen verwendet werden.

4.1. Übergabe einer URL an den Browser

Um der als Browser funktionierenden Visualisierungskomponente eine URL zu übergeben, ist es nur notwendig, eine Formel anzugeben. In dieser Formel muss eine URL als String enthalten sein, beispielsweise

```
result.string("http://www.adito.de");
```

Wichtig ist hierbei, dass die URL vollständig übergeben wird, also inklusive dem http-Präfix. Sie können selbstverständlich auch die URL über `vars.set` übergeben.

4.2. Auslesen einer URL aus dem Browser

Damit eine übergebene bzw. gesetzte URL ausgelesen wird, reicht es, einen `vars.getString`-Befehl auf die Komponente zu setzen. Damit die jeweils angezeigte URL ausgelesen wird, ist es allerdings notwendig, einen Prozess zu definieren. Die möglichen Eigenschaften werden im Folgenden aufgelistet.


4.3. Eigenschaften der Browser-Visualisierung

Die Eigenschaften der Visualisierungskomponente als Browser werden in ADITO4 online ebenfalls im `Eigenschaftenfenster` (Siehe Kapitel 2.2.3) gesetzt.

In den meisten Prozessen steht Ihnen eine lokale Variable `$local.value` zur Verfügung. In den meisten Fällen handelt es sich um ein Objekt, das an Stelle [0] die UID des Browser-Fensters beinhaltet und in [1] den entsprechenden Wert.

Eigenschaft	Beschreibung
javascriptEnabled	Gibt an, ob JavaScript in Websites aktiviert werden soll oder nicht.
browserType	Gibt den zu verwendenden Browsertypen an. Mögliche Werte sind hier „Native, bevorzugt“, „JavaFX“ und „Native“
onLocationChange	Um die gerade angezeigte URL auszulesen, können Sie den <code>locationProcess</code> verwenden. Hier steht Ihnen die lokale Variable <code>\$local.value</code> zur Verfügung, hier als Objekt. An Stelle 0 steht die UID des aktuellen Browser-Fensters, Stelle 1 besitzt die aktuelle URL. Dieser Prozess wird bei Klick auf einen Link im Browser aufgerufen. Diesem Prozess können Sie zusätzlich ein <code>result.string("true")</code> oder <code>result.string("false")</code> zurückgeben, wenn Sie das Verfolgen eines Links verhindern möchten.
onOpenWindow	Gibt eine UID zurück, die für das aktuelle Browser-Fenster steht. Der Prozess wird ausgeführt, wenn das aktuelle Browser-Fenster geöffnet wird.
onCloseWindow	Wird ausgeführt, wenn das aktuelle Browser-Fenster geschlossen wird.
onTitleChange	Kann den Seitentitel auslesen. Dieser steht in der lokalen Variable <code>\$local.value</code> als Objekt zur Verfügung. Der Prozess wird aufgerufen, wenn sich der Seitentitel ändert.
onStatusChange	Zeigt den aktuellen Status, den der Browser zurückliefert über das Objekt <code>\$local.value</code> . Der Prozess wird aufgerufen, wenn sich der Status des Browsers ändert.
onProgressChange	Zeigt den aktuellen Fortschritt des Browsers. Der Prozess wird über das Objekt <code>\$local.value</code> ausgelesen. Er gibt den Ladestand in Prozent an.

5. Visualisierung als Tachometer/ Thermometer

Für die Tachometer oder Thermometerdarstellung muss die Visualisierungskomponente  **MeterChart** verwendet werden.

5.1. Setzen der Werte

Für die Diagramme vom Typ "Instrument" wird der anzuzeigende Wert über die Eigenschaft `thresholdsProcess` (ADITO4 online) im Reiter `Andere` in den Eigenschaften gesetzt. Der dort enthaltene JDito-Code muss einen Wert zurückliefern, der dann visualisiert wird. Für die Rückgabe stehen Ihnen die normalen Rückgabemethoden zur Verfügung (`result.string()`, `result.object()` und `result.query()`).

Im obigen Beispiel (siehe Kapitel 2.2.2.) ist als Formel der Ausdruck

```
result.string(34);
```

eingetragen. Es würde also auf dem Thermometer bzw. dem Tacho der Wert 34 angezeigt.

5.2. Eigenschaften der Visualisierung als Tachometer/ Thermometer

Die Eigenschaften der Visualisierungskomponente als Browser werden in ADITO4 online ebenfalls im `Eigenschaftenfenster` (Siehe Kapitel 2.2.3) gesetzt.

In den meisten Prozessen steht Ihnen eine lokale Variable `$local.value` zur Verfügung. In den meisten Fällen handelt es sich um ein Objekt, das an Stelle [0] die UID des Browser-Fensters beinhaltet und in [1] den entsprechenden Wert.

Eigenschaft	Beschreibung
thresholdsProcess	Prozess, mit dem die Werte für das Diagramm ermittelt werden.
diagramType	Stellt den Diagrammtyp ein. Verfügbar sind derzeit die Typen: <ul style="list-style-type: none"> • Tachometer • Thermometerdarstellung
colorsProcess	Prozess, in dem die Farbe eingestellt wird.
valueFontSize	Schriftgröße
tickSize	Gibt an, wann ein Unterteilungsstrich gemacht wird.
valueColor	Farbe, die in der Komponente verwendet wird.

5.3. Beispiel

Die Tachometerdarstellung unterscheidet sich in einem zentralen Punkt vom Diagramm: Es ist über die Eigenschaft `valueProcess` zusätzlich ein Wert anzugeben, der den aktuellen "Level" der Komponente (Wert des Thermometers, anzuzeigender Wert im Tachometer) visualisiert. Der folgende Prozess definiert ein Tachometer (Eigenschaft `Diagrammtyp` auf "Tachometer" eingestellt):

Schwellenwerte:

```
import("system.result");
var ret = new Array;
ret[0] = 0;
ret[1] = 1000;
ret[2] = 2000;
ret[3] = 3000;
ret[4] = 4000;
ret[5] = 5000;
ret[6] = 6000;
ret[7] = 7000;
ret[8] = 8000;

result.object(ret)
```

Farben:

```
import("system.result");
var color = new Array();
color[0] = "-1"; // - 1000
color[1] = "-1"; // - 2000
color[2] = "-1"; // - 3000
color[3] = "-52"; // - 4000
color[4] = "-103"; // - 5000
color[5] = "-205"; // - 6000
color[6] = "-13261"; // - 7000
color[7] = "-65536"; // - 8000

result.object(color);
```

Damit nun auch ein entsprechender Wert angezeigt werden kann, muss in `valueProcess` noch ein bestimmter Wert festgelegt werden, über die JDito-Funktion `result.string`, beispielsweise

```
import("system.result");  
result.string(3500);
```