



# JetViewSoft

Versionsupdate von V. 5.2.2 auf V. 5.3.0

We automate your success

Jetter AG  
Gräterstrasse 2  
71642 Ludwigsburg

**Telefon:**

Zentrale	+49 7141 2550-0
Vertrieb	+49 7141 2550-531
Technische Hotline	+49 7141 2550-444

**E-Mail:**

Hotline	hotline@jetter
Vertrieb	vertrieb@jetter

Version 1.00

19.12.2017

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Inhalt	2
1 Erweiterungen	1
1.1 Editor	1
1.1.1 Automatisches Scrollen der Maske am Fensterrand	1
1.1.2 Funktion Zoom Normal	1
1.1.3 Funktionen Zoom Percent und Custom bei VG-Plattformen	1
1.1.4 Einzelne Auswahlpunkte mit Pfeiltasten verschieben	1
1.2 Dynamik	1
1.2.1 Update-Intervall einer Variablen	1
1.2.2 Animation Duration	1
1.2.3 Bei einem Pointer das referenzierte Objekt über DynIO umschalten	1
1.3 Objekte	2
1.3.1 Änderungsmöglichkeit der Größe von Scrollbalken	2
1.3.2 Füllfarbe eines Bargraph-Objekts über Punktnotation ändern	2
1.3.3 Füllfarbe eines Bargraph-Objekts über VisuCommand ändern	2
1.3.4 Neue Linien-Eigenschaft „End Cap Style“	2
1.3.5 Neue Einstellmöglichkeit Font-Attributes für Alarm-Objekt	2
1.3.6 Neue Eigenschaft "Join Style" für Line, Polylinie und Polygon	2
1.3.7 Bilddatei eines Image-Objects zur Laufzeit auslesen und ändern	3
1.4 STX	3
1.4.1 Optimierung der VisuCommands (VG-Plattform)	3
1.5 Allgemein	3
1.5.1 Optimierung der Versionskontrolle	3
1.5.2 Wert-Übernahme im ColorChangeLimits- und AlarmService-Dialog vereinheitlicht	3
1.5.3 Image-Caching in der STX-Laufzeit für Speicherauslastung optimiert	4
1.5.4 Image-Caching in der VG-Plattform wurde optimiert	4
1.5.5 Inkonsistenzen zwischen VisuCommands und Punktnotation beseitigt	5
2 Beseitigte Software-Bugs	6
2.1 Dynamiken und Events	6
2.1.1 Fehler beim dynamischen Farbwechsel in einer Grid-Zelle	6
2.1.2 Dyn-IO und Dyn-Rotation mit Kombination aus Slider und Gruppe	6
2.1.3 Reihenfolge der ausgelösten Events beim Laden	6
2.1.4 Falsche Fehlermeldung bei Editbox mit DynIO bei der Werteeingabe	6
2.1.5 Bei DynMove- und DynScale-Dynamik von Arc und Ellipse wurden Bargraph-Eigenschaften nicht aktualisiert	6
2.2 Hardware	6
2.2.1 Display-Hardwareänderung von Width/Height bei großen Projekten benötigte lange	6
2.3 Laufzeit	6
2.3.1 Laufzeit zeigte falschen Farbverlauf bei LinearGradient	6
2.3.2 Aussehen des Line-Objekts (ER-STX (CE)/(PC)	6

---

2.4	Variablen .....	7
2.4.1	Im Auswahldialog für Variablen konnten Spalten entfernt werden.....	7
2.5	Alarm .....	7
2.5.1	Kopieren von Alarm Controls setzte die Properties auf Default-Werte .....	7
2.5.2	Löschen von Oberkategorien im Alarmservice-Dialog.....	7
2.6	STX.....	7
2.6.1	VisualInterface-Datei enthielt VisuAlarmControl-Objekte für Alarm-Dialoge.....	7
2.7	Objekte .....	7
2.7.1	Property ReadOnly bei Editbox funktionierte nicht (VG-Plattform).....	7
2.8	Allgemein.....	7
2.8.1	Ein geöffnetes Kontextmenü schloss sich nicht.....	7
2.8.2	Skalieren von großen Objekten .....	7
2.9	Editor .....	8
2.9.1	Gruppen wurden im Component-Editor nicht richtig dargestellt.....	8

# 1 Erweiterungen

Nachfolgend sind alle bei dieser Version hinzugekommenen Features sowie Erweiterungen aufgelistet.

## 1.1 Editor

### 1.1.1 Automatisches Scrollen der Maske am Fensterrand

Wird ein Objekt per Drag & Drop innerhalb einer Maske verschoben, dann wird jetzt automatisch gescrollt, wenn man den Rand des Maskenfensers erreicht.

### 1.1.2 Funktion Zoom Normal

Die Funktion zoomt eine Maske auf 100 % und verschiebt die linke obere Ecke auf Position **0,0**. Diese Funktion hat nun die Tastaturkombination **STRG + 0**.

### 1.1.3 Funktionen Zoom Percent und Custom bei VG-Plattformen

Wenn Sie über die Menüeinträge Zoom Percent und Zoom Custom zoomen wird die Mitte der Maske beibehalten.

### 1.1.4 Einzelne Auswahlpunkte mit Pfeiltasten verschieben

Sie können bei den Objekten Line, Polyline und Polygon einzelne Auswahlpunkte mit der Tastatur verschieben. Näher Informationen finden Sie bei der Beschreibung des einzelnen Objekts.

## 1.2 Dynamik

### 1.2.1 Update-Intervall einer Variablen

Sie können bei jedem Objekt einstellen, wie hoch das Update-Intervall der zugehörigen Variable bei einer **Dynamic IO** ist.

Dies geschieht über die neuen Eigenschaften **Overwrite Update Interval** und **Update Interval** bei der **Dynamic IO**. Dadurch wird das normale Update Intervall einer Variable überschrieben.

Es können lediglich Update Intervalle von Steuerungs-Variablen überschrieben werden, da lokale Variablen immer sofort aktualisiert werden.

Wenn Sie ein geringeres Update-Intervall als in der Variablen definiert ist wählen, dann gibt der Compiler eine Warnung aus.

### 1.2.2 Animation Duration

Dieser Wert beeinflusst die Animation bei einer DynIO, z. B. bei einem Meter-Objekt.

Dadurch erreichen Sie eine flüssigere Animation und verhindern z. B. Sprünge des Zeigers bei einem Meter-Objekt.

### 1.2.3 Bei einem Pointer das referenzierte Objekt über DynIO umschalten

Es gab keine Möglichkeit Pointer, z. B. auf Gruppen, über eine DynIO umzuschalten.

Dies ist nun mit einer DynIO möglich, wenn Sie als **Target Property** die Eigenschaft **ReferencedObject** wählen.

## 1.3 Objekte

### 1.3.1 Änderungsmöglichkeit der Größe von Scrollbalken

Neue Einstellmöglichkeit **Scrollbar Size** in Prozent (Standard = 100) unter **Projekt/Properties /ER/Components**.

Eine Änderung des Werts bedeutet, dass die Breite und Höhe der Scrollbar, je nachdem ob vertikal oder horizontal, als auch die Pfeilsymbole in ihrer Größe verändert werden.

Beispiel: Wenn Sie 200 als Wert eingeben, dann verdoppelt sich die Breite und Höhe der Scrollbar.

### 1.3.2 Füllfarbe eines Bargraph-Objekts über Punktnotation ändern

Die Füll- und Hintergrundfarbe eines Bargraph-Objekts kann nun per Punktnotation geändert werden. Möglich ist dies für die Objekte **Arc**, **Rectangle**, **Ellipse** und **Polygon**.

Eingebaut in Visu-Library ab Version 4.0.0.0.

### 1.3.3 Füllfarbe eines Bargraph-Objekts über VisuCommand ändern

Die Füll- und Hintergrundfarbe eines Bargraph-Objekts kann nun per **VisuCommandAttribute** geändert werden. Möglich ist dies für die Objekte Arc, Rectangle, Ellipse und Polygon.

Eingebaut in Visu-Library ab Version 4.0.0.0.

### 1.3.4 Neue Linien-Eigenschaft „End Cap Style“

Mit dieser Eigenschaft bestimmen Sie das Aussehen der Endpunkte einer Linie oder Polylinie. Es gibt folgende Möglichkeiten:



### 1.3.5 Neue Einstellmöglichkeit Font-Attributes für Alarm-Objekt

- Erweiterung der Dialog Properties des Alarm Service um die zentrale Einstellmöglichkeit von Font Attributes für alle Alarm-Dialoge und Controls.
- Erweiterung der Punkt-Notation um die Möglichkeit das Font-Attribute über das Application-object (WorkingSet).

### 1.3.6 Neue Eigenschaft "Join Style" für Line, Polylinie und Polygon

Die Objekte Line, Polyline und Polygon wurden um die Eigenschaft **Join Style** erweitert (nur VG-Plattformen).

### 1.3.7 Bilddatei eines Image-Objects zur Laufzeit auslesen und ändern

**Voraussetzung:** Visu-Library 4.0.0.0.

Die Klasse `VisuImage` wurde um die Eigenschaft **FilePath** erweitert, um zur Laufzeit die Bilddatei eines Image-Objekts auszulesen und zu ändern.

Das Image-Objekt wurde um die Befehle **VisuCmdString** und **VisuCmdGetString** erweitert, da diese intern den Dateipfad setzen und lesen.

Setzen des `FilePath`:

```
_english_flag_9000.FilePath := 'C:\Users\j\h\Pictures\russland_flag.bmp';
```

Abfrage des `FilePath`:

```
Trace(_english_flag_9000.FilePath);
```

Handelt es sich bei dem Image-Objekt um ein `ImageList`-Objekt, dann bezieht sich `FilePath` immer auf das aktuell angezeigte Bild.

## 1.4 STX

### 1.4.1 Optimierung der `VisuCommands` (VG-Plattform)

Die `VisuCommands` werden nun doppelt bis dreimal so schnell abgearbeitet.

## 1.5 Allgemein

### 1.5.1 Optimierung der Versionskontrolle

Unterhalb von **Projekt/Properties/Deployment** wurde das neue **Property List Of Files And Folders** in der Kategorie **Excluded Files And Folders** eingeführt. Die hier aufgeführten Einträge werden nicht deployed.

Unterhalb von **Projekt/Properties/Compiler** wurde das neue **Property List Of Files And Folders** in der Kategorie **Protected Files And Folders** eingeführt. Die hier aufgeführten Einträge werden nicht gelöscht, wenn `Clean Workspace` ausgeführt wird.

### 1.5.2 Wert-Übernahme im `ColorChangeLimits`- und `AlarmService`-Dialog vereinheitlicht

Wurde bei einem `ColorChangeLimits`-Dialog ein Limit oder Blinkintervall geändert und `Return` für die Übernahme des geänderten Werts gedrückt, dann wurde der Dialog sofort geschlossen. Beim `EditColumns`-Dialog des `AlarmService` trat das gleiche Verhalten auf.

### 1.5.3 Image-Caching in der STX-Laufzeit für Speicherauslastung optimiert

Das **Vorab - Bildercaching** wird jetzt automatisch justiert, wenn die aktuelle Speicherauslastung eine bestimmte Grenze, Standard 90%, überschreitet. Es wird dann versucht Bilder, welche längere Zeit (Standard 30 Min.) nicht mehr benötigt wurden, aus dem Speicher zu entfernen. Es werden solange Bilder aus dem Speicher genommen, bis eine Untergrenze (Standard 70%) erreicht wird. Das Prüfen der Speicherauslastung erfolgt hierbei in einem festen Intervall (Standard 5sek)

Die Standardwerte können in der Datei **JVER.xml** im Bereich **General** wie folgt überschrieben werden:

```
<General
.
.
.
.
ImagePreCachingEnabled      = "1"           ein/ausschalten des vorab Bildercachings
ImageGCUpperMemBoundary    = "90"           90% obere Speichergrenze
ImageGCLowerMemBoundary    = "70"           70% untere Speichergrenze
ImageGCRemoveTimespan      = "1800"         30Min. nicht verwendet
ImageGCCheckInterval       = "5000"         Prüfintervall in ms
/>
```

### 1.5.4 Image-Caching in der VG-Plattform wurde optimiert

Beim Caching von Bilder für die VG-Plattform wird jetzt der verfügbare Hauptspeicher (RAM) auch berücksichtigt. Bei großen Projekten mit sehr vielen Bildern wird so sichergestellt, dass das Caching nicht den RAM-Speicher zu sehr in Anspruch nimmt.

Eine weitere optionale Optimierung ermöglicht eine intelligente Speicherung der Caching-Reihenfolge von Bildern. Wird diese Option aktiviert (manuell in der Datei JVER.xml), dann speichert das Programm eine sortierte Liste mit Bildern, die zuletzt verwendet wurden. Diese Bilder werden dann beim nächsten Start als erstes in den Cache geladen und stehen dadurch bereits frühzeitig zur Verfügung.

In der Datei JVER.xml im XML-Knoten General können folgende Einstellungen gemacht werden:

- **ImagePreCachingEnabled="1"**  
Aktivieren/Deaktivieren des Cachings beim Start der Applikation im Hintergrund
- **ImagePreCachingWriteToDiskInterval="0"**  
Intervall in Sekunden, in dem die PreCaching Informationen aktualisiert werden. Der Wert 0 deaktiviert diese Funktion.
- **ImagePreCachingWriteToDiskFilePath="\Storage Card\JVER\_PreCaching.txt"**  
Vollständiger Dateipfad zur Datei für die PreCaching info. Diese Datei sollte nicht auf dem Flash-Speicher des Displays abgelegt werden, da dieser nur eine begrenzte Anzahl von Scheib Zyklen zulässt.
- **ImageGCUpperMemBoundary = "90"**  
Speichergrenze in % ab der die Bildverwaltung Bilder aus dem Cache wieder freigibt
- **ImageGCLowerMemBoundary = "70"**  
Untere Speichergrenze, bis zu der die Bildverwaltung Bilder aus dem Cache freigibt



### **1.5.5 Inkonsistenzen zwischen VisuCommands und Punktnotation beseitigt**

Es konnte zu abweichenden Ergebnissen führen, wenn VisuCmdAttribute, bzw. VisCmdGetAttribute zusammen mit der Punktnotation benutzt wurde. Dieses Fehlverhalten wurde beseitigt.

Zusätzlich kann nun auch über STX mit den VisuCommands und mit der Punktnotation die aktuelle Datenmaske ermittelt oder gesetzt werden. Die Softkeymaske einer beliebigen Datenmaske kann auch gesetzt und ermittelt werden.

## 2 Beseitigte Software-Bugs

Dieses Kapitel beschreibt die in der neuen Softwareversion beseitigten Bugs.

### 2.1 Dynamiken und Events

#### 2.1.1 Fehler beim dynamischen Farbwechsel in einer Grid-Zelle

Beim Wechsel von einer blinkenden Farbe in eine statisch wurde der Farbwert nicht immer korrekt angezeigt. Der Fehler trat nur in Projekten mit ER-STX-VG-Plattform auf.

#### 2.1.2 Dyn-IO und Dyn-Rotation mit Kombination aus Slider und Gruppe

Die Dyn-IO und Dyn-Rotation mit Kombination aus Slider und Gruppe funktionierte nicht mehr in der VG-Laufzeit.

#### 2.1.3 Reihenfolge der ausgelösten Events beim Laden

Die Events OnLoad, OnBeforeLoad, OnUnload und OnBeforeUnload der Maske und des Workingsets wurden zeitlich nicht korrekt abgearbeitet.

#### 2.1.4 Falsche Fehlermeldung bei Editbox mit DynIO bei der Werteeingabe

Enthielt eine numerische Editbox bei der Formatierung der DynIO einen Punkt, dann bekam man dauernd bei der Eingabe die Fehlermeldung **Ungültige Eingabe! Geben Sie einen gültigen Wert ein!** Die Editbox konnte dann nicht mehr verlassen werden.

#### 2.1.5 Bei DynMove- und DynScale-Dynamik von Arc und Ellipse wurden Bargraph-Eigenschaften nicht aktualisiert

Bei DynMove- und DynScale-Dynamik wurden die Bargraphen einer Ellipse und eines Arc nicht neu gezeichnet. Das gleiche war auch beim Bargraph des Rectangles mit DynScale-Dynamik.

### 2.2 Hardware

#### 2.2.1 Display-Hardwareänderung von Width/Height bei großen Projekten benötigte lange

Aktualisieren aller Masken nach Änderungen an einer Display-Hardware konnte um das dreifache beschleunigt werden.

### 2.3 Laufzeit

#### 2.3.1 Laufzeit zeigte falschen Farbverlauf bei LinearGradient

VG-Laufzeit zeigte einen falschen Farbverlauf bei Angle 90 und 270, wenn dieser auf LinearGradient eingestellt wurde.

#### 2.3.2 Aussehen des Line-Objekts (ER-STX (CE))/(PC)

Line-Objekte wurden in der Entwurfszeit und Simulation anders gezeichnet, als auf dem Display. Ein Line-Objekt mit der Stärke >1 Point wurde auf dem Display mit flachen Linienenden gezeichnet, aber zur Entwurfszeit und Simulation mit runden Linienenden. Nun werden die Linienenden überall flach dargestellt.

## 2.4 Variablen

### 2.4.1 Im Auswahldialog für Variablen konnten Spalten entfernt werden

Wurde im Properties-Fenster das Fenster zur Auswahl einer Variablen geöffnet, dann konnte der Anwender die Anordnung der Spalten mit der Maus ändern. Es konnte hierbei passieren, dass eine Spalte vollständig entfernt wurde. Danach war es nicht mehr möglich, diese Spalte wieder sichtbar zu machen. Die Bearbeitung der Spaltenanordnung wurde deaktiviert, da es an dieser Stelle keinen besonderen Anwendernutzen hat. Dadurch tritt der Fehler nicht mehr auf.

## 2.5 Alarm

### 2.5.1 Kopieren von Alarm Controls setzte die Properties auf Default-Werte

Beim Kopieren von Alarm Controls wurden die Werte bei den Properties nicht mit kopiert.

### 2.5.2 Löschen von Oberkategorien im Alarmservice-Dialog

Bisher war es möglich Oberkategorien im Alarmservice-Dialog zu löschen. Das war aber nicht zulässig und wurde nun deaktiviert.

## 2.6 STX

### 2.6.1 VisualInterface-Datei enthielt VisuAlarmControl-Objekte für Alarm-Dialoge

In der Deklarations-Datei **VisualInterface.stxp** wurden fälschlicherweise die AlarmDialoge heineingeschrieben. Dieser Fehler wurde im STX(VG)-Compiler behoben.

## 2.7 Objekte

### 2.7.1 Property ReadOnly bei Editbox funktionierte nicht (VG-Plattform)

Das Property **ReadOnly** wurde bei der Editbox nicht gesetzt und es konnte weiterhin Text eingegeben werden.

## 2.8 Allgemein

### 2.8.1 Ein geöffnetes Kontextmenü schloss sich nicht

Ein geöffnetes Kontextmenü schloß sich nicht, wenn an anderer Stelle mit der Maus geklickt wurde. Das Verhalten trat besonders in Dialogen auf.

### 2.8.2 Skalieren von großen Objekten

Es war bisher nicht möglich Objekte zu skalieren, die größer als die Maske waren. Nun können Sie Objekte beliebiger Größe skalieren. Die Performance beim Skalieren von Bildern in VG-Plattformen wurde optimiert.

## **2.9 Editor**

### **2.9.1 Gruppen wurden im Component-Editor nicht richtig dargestellt**

Beim Öffnen einer Gruppe im Component-Editor wurde diese nicht korrekt oben links positioniert.

Wurde im Component-Editor ein Zoomfaktor  $> 1$  eingestellt und der Component-Editor wurde geschlossen und wieder geöffnet, dann wurde ein falscher Zoomfaktor verwendet. Das zu editierende Objekte befand sich außerhalb des sichtbaren Bereichs