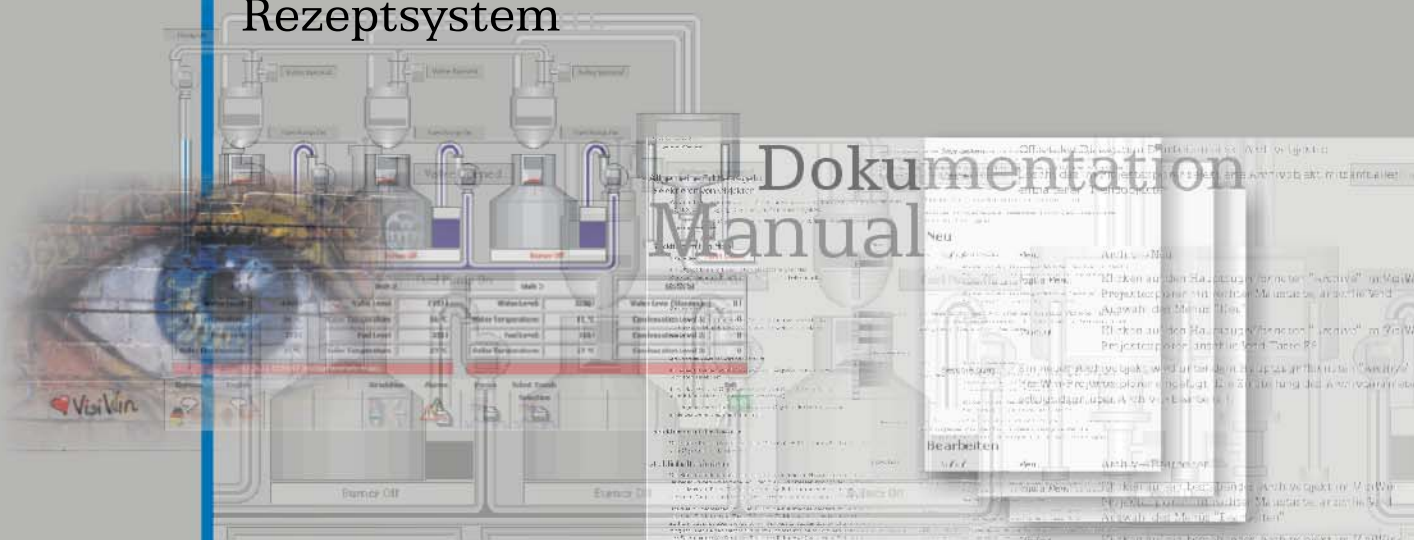


# VisiWinNET 2005

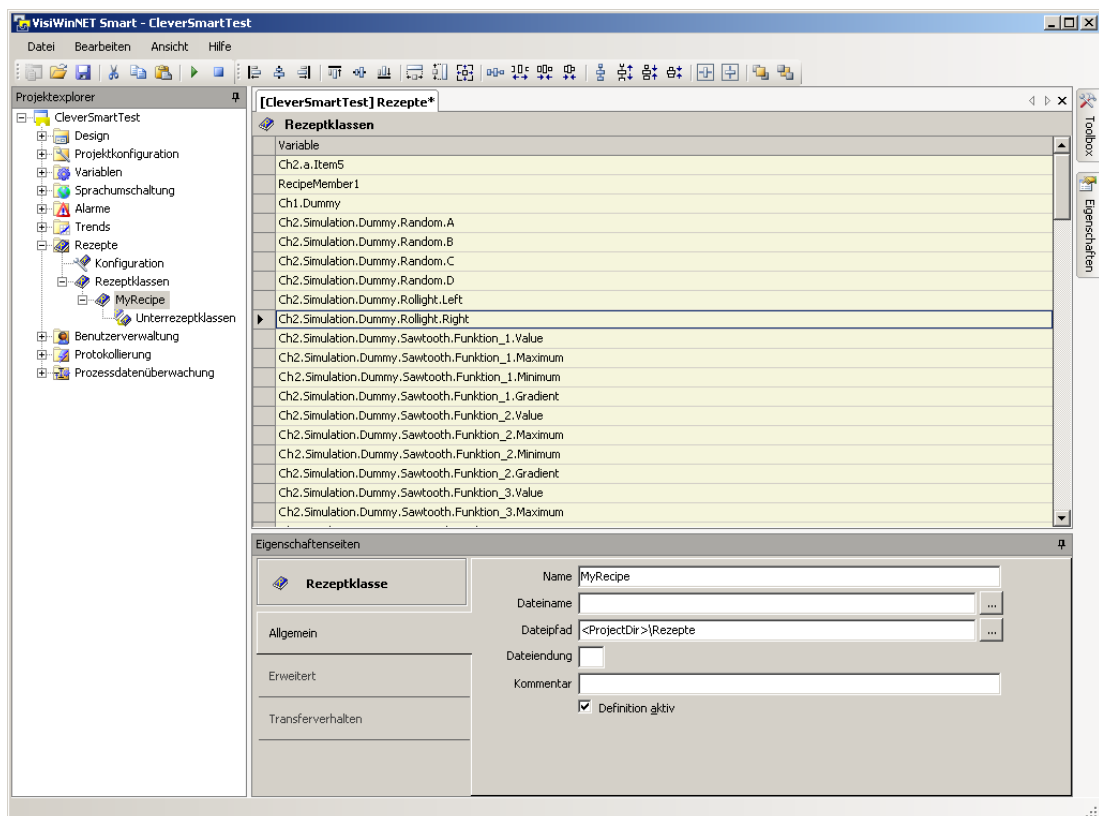
## Rezeptsystem



- VisiWin
- VisiWinNET 2005
  - Allgemeines
  - Klassenbibliothek
- Systeme
  - Werkzeuge
  - Technische Infos
- Inosoft OPCServer
  - Grundlagen und Hilfsmittel
  - Protokolle

# VisiWinNET 2005







## Rezeptsystem



Ohne ausdrückliche schriftliche Erlaubnis der INOSOFT GmbH darf der Inhalt des Handbuches nicht anderweitig verwendet werden.  
Den Inhalt des Handbuches haben wir auf Übereinstimmung mit der beschriebenen Software geprüft. Trotzdem können Abweichungen nicht ausgeschlossen werden. Aus diesem Grund können wir für eine völlige Übereinstimmung keine Gewähr geben. Der Inhalt des Handbuches wird regelmäßig auf Änderungen geprüft. Korrekturen sind in der nachfolgenden Ausgabe vorhanden. Verbesserungsvorschläge werden dankbar angenommen.

## Legende

Um auf besondere Einstellungen aufmerksam zu machen, werden in den INOSOFT-Dokumentationen folgende Symbole verwendet:

	<b>Achtung</b>	Absätze mit diesem Zeichen sollten Sie besonders aufmerksam lesen - und beachten.
	<b>Hinweis</b>	Wichtige Einstellung "additional informations".
	<b>Tipp</b>	Viele Wege führen nach Rom, hier finden Sie eine Abkürzung.
	<b>In Arbeit</b>	Funktionen in Vorbereitung, bzw. bereits implementiert, jedoch noch nicht für die Dokumentation aufbereitet.
	<b>Beispiel ausführen</b>	Anweisungen, die in einem Beispiel auszuführen sind.
	<b>Beispiel beobachten</b>	Ergebnisse, die durch die Ausführung der Beispielanweisungen zu beobachten sind.

© / ™ / ®

Windows®, Windows NT®, Windows 2000®, Windows XP®  
sind eingetragene Warenzeichen der Firma Microsoft.

Weitere mit ® erwähnte Produktnamen sind Warenzeichen der betreffenden  
Herstellerfirmen.

INOSOFT GmbH

VisiWinNET Version: ab 6.04.0000

erstellt am 07.06.2010

# Inhalt

<b>1</b>	<b>Vorwort</b>	<b>1</b>
<b>2</b>	<b>Einführung in das VisiWinNET Rezeptsystem</b>	<b>2</b>
<b>3</b>	<b>Projektierung</b>	<b>5</b>
<b>4</b>	<b>Definitionen des Rezeptsystems</b>	<b>6</b>
4.1	Rezept	6
4.1.1	Rezepte bearbeiten	8
4.2	Rezeptelement	8
4.2.1	Rezeptelemente bearbeiten	8
<b>5</b>	<b>Parameter der Rezeptdefinitionen</b>	<b>10</b>
5.1	Parameter in alphabetischer Reihenfolge	10
5.1.1	Änderungshistorie aktivieren	11
5.1.2	Beschreibungsvariable	11
5.1.3	Dateiendung	11
5.1.4	Definition aktiv	11
5.1.5	geänderte Variablenkernwerte direkt speichern	12
5.1.6	geänderte Werte direkt schreiben	12
5.1.7	geänderte Werte direkt speichern	12
5.1.8	Kommentar	12
5.1.9	Ladetrigger	13
5.1.10	Lesetrigger	13
5.1.11	Letzte Rezeptdatei beim Start laden	13
5.1.12	Max. Anzahl Dateien	14
5.1.13	Name	14
5.1.14	Normfaktor	14
5.1.15	Rezeptdateipfad	14
5.1.16	Rezeptwerte absolut/relativ	14
5.1.17	Schreibtrigger	15
5.1.18	Sollfaktorvariable	15
5.1.19	Speichertrigger	15
5.1.20	Unterrezept	16
5.1.21	Unterrezeptklassenwerte hier speichern	18
5.1.22	Variable	18
5.1.23	Variable für Dateiname	18
<b>6</b>	<b>Konfiguration des Rezeptservers</b>	<b>19</b>

# 1 Vorwort

## Zu diesem Handbuch

Dieses Handbuch enthält spezifische Informationen zum VisiWinNET Rezeptsystem. Enthalten sind unter anderem die Erklärung der beteiligten Komponenten, die Bedienreferenz des Editors und die Beschreibung der Definitionen des Rezeptsystems.

## Fragen und Probleme

Bei technischen Fragen und Problemen wenden Sie sich bitte an Ihre zuständige INOSOFT Vertretung oder an den Support der INOSOFT GmbH unter Tel.: +49 (5221) 16 66 02 bzw. E-Mail: [Support@INOSOFT.com](mailto:Support@INOSOFT.com)

Häufige Fragen und Probleme werden auch auf unserer Homepage unter [www.inosoft.com](http://www.inosoft.com) behandelt.

Dort finden Sie auch einen Supportbereich, in dem Sie sich mit Fragen direkt an unsere Zentrale wenden können.

## 2 Einführung in das VisiWinNET Rezeptsystem

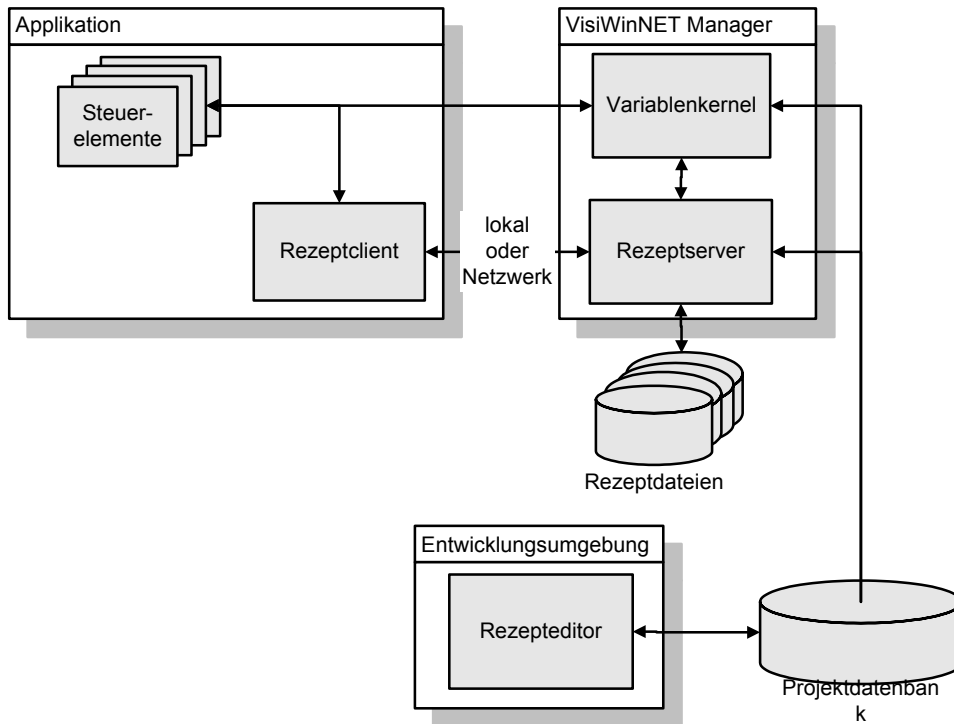
Visualisierungsrezepte beschreiben Sätze von Prozesswerten. Diese legen üblicherweise die Einstellungen einer Maschine zur Fertigung eines Produktes aus einer Produktpalette fest. Über das Schreiben von Rezepten zur SPS kann damit zwischen den einzelnen zu fertigenden Produkten umgeschaltet werden.

Das VisiWinNET Rezeptsystem erfüllt alle Anforderungen einer komfortablen Rezeptverwaltung:

- Über die Projektierung von Rezeptdefinitionen werden die für die Ausführung eines Rezeptes nötigen Prozessvariablen aus der Prozessanbindung ausgewählt.
- Der Transfer der Rezeptdaten zwischen SPS und Visualisierung ist sowohl schreibend als auch lesend steuerbar. Über den schreibenden Zugriff werden bestehende Rezepte in die SPS transferiert. Der lesende Zugriff ermöglicht es, an der Maschine optimierte Datensätze in das Visualisierungssystem zu übertragen. Der Transfer kann sowohl aus der Visualisierung als auch von der SPS gesteuert werden.
- Der Transfer der Rezeptdaten zwischen Visualisierungssystem und Rezeptdateien ist sowohl schreibend als auch lesend steuerbar. Über den lesenden Zugriff werden Daten aus einer Rezeptdatei in das Visualisierungssystem geladen. Der schreibende Zugriff ermöglicht es, an der Maschine optimierte Rezeptdaten persistent zu machen. Auch hier ist eine Steuerung sowohl manuell aus der Visualisierungsapplikation als auch automatisch über die SPS möglich.

Anhand des folgenden Blockschaltbildes werden die Komponenten des Rezeptsystems vorgestellt. Weiterführende Informationen über die Komponenten werden als Querverweise angegeben.

Komponenten des VisiWinNET Rezeptsystems



Komponente

Beschreibung

**Variablenkernel**

Der Variablenkern liefert die an den Rezepten beteiligten Prozesswerte. Weiterführende Informationen befinden sich im Handbuch "Prozessanbindung"

**Rezeptserver**

Der Rezeptserver puffert Rezeptdaten zentral. Er dient als Schnittstelle zwischen Rezeptclients, Variablenkernel und Rezeptdateien. Neben der Übertragung der Rezeptdaten zu den angeschlossenen Rezeptclients (und damit zu den Steuerelementen in der Visualisierung) unterstützt er im Wesentlichen vier Transferfunktionen:

- Lesen: Die Rezeptwerte werden aus dem VWOPC-Kernel in den Puffer des Rezeptservers übertragen.
- Schreiben: Die Rezeptwerte werden aus dem internen Puffer des Rezeptservers in den VWOPC-Kernel übertragen. Von dort werden sie an die SPS weitergereicht.
- Laden: Die im Puffer gehaltenen Rezeptdaten werden aus einer Rezeptdatei gelesen. Die Dateisteuerung (Name, Verzeichnis und Beschreibung der Datei) erfolgt mithilfe der in der Rezeptdefinition enthaltenen Parameter.

- Speichern: Die im Puffer gehaltenen Rezeptdaten werden in eine Rezeptdatei geschrieben. Die Dateisteuerung (Name, Verzeichnis und Beschreibung der Datei) erfolgt mithilfe der in der Rezeptdefinition enthaltenen Parameter.

Informationen über die Einstellmöglichkeiten des Rezeptservers befinden sich im Kapitel "Konfiguration des Rezeptservers" (Kap.: 6) in diesem Handbuch.

**Rezeptclient**

Der Rezeptclient ermöglicht als zentrale Sammelstelle in einer Visualisierungsapplikation den Zugriff auf die im Rezeptserver gepufferten Rezeptdaten und die Transferfunktionen im Rezeptserver.

**Steuerelemente**

Die Steuerelemente im Rezeptmodus können zwischen zwei Ansichten umgeschaltet werden. Sie können entweder die aktuellen Werte aus dem Variablenkernel oder die gepufferten Werte aus dem Rezeptserver anzeigen.

**Projektdatenbank**

Die Projektdatenbank enthält die Definitionen des Rezeptsystems. Zur Entwicklungszeit muss festgelegt werden, welche Prozessvariablen zu einem Rezept gehören (Rezeptelemente). Über die Parameter einer Rezeptdefinition wird festgelegt, wie und wo Rezeptdaten gespeichert werden, welche Prozessvariable bei einer SPS-seitigen Steuerung die Transferfunktionen im Rezeptserver steuern kann, welche Normfaktoren bei der relativen Angabe von Rezeptgrößen angewandt werden sollen und wie Unterrezepte eingebunden werden sollen.

**Rezepteditor**

Der Rezepteditor ist das Projektierwerkzeug für die Definitionen des Rezeptsystems. Er wird in der Entwicklungsumgebung im Projektexplorer dargestellt. Hinweise zur Bedienung des Rezepteditors befinden sich im gleichnamigen Kapitel in diesem Handbuch.

**Rezeptdateien**

Enthalten die anhand von Rezeptdefinitionen zusammengestellten Listen der Rezeptwerte. Informationen über das Format der Rezeptdateien befinden sich im Kapitel "Konfiguration des Rezeptservers" (Kap.: 6) in diesem Handbuch.

### 3 Projektierung

VisiWinNET stellt einen Editor zum Projektieren der Definitionen des Rezeptsystems zur Verfügung. Dieser wird bei der Installation von VisiWinNET in die Entwicklungsumgebung integriert. Die Rezeptdefinitionen werden in der Projektdatenbank gespeichert.

Der Rezepteditor wird im VisiWinNET-Projektexplorer durch das Symbol  repräsentiert.

Nach dem erstmaligen Anklicken des Knotens "Rezepte" initialisiert sich der Editor: Im darunter liegenden Knoten "Rezeptklassen" werden alle bereits in der Projektdatenbank definierten Rezeptdefinitionen als Unterknoten eingefügt.

Zusätzlich wird der Knoten "Konfiguration Rezeptsystem" eingefügt. Hier werden systemweite Einstellungen, wie beispielsweise das Format der Rezeptdateien eingestellt.

Der Tabelleneditor wird über das Kontextmenü "Editor anzeigen" oder einen Klick auf einen Rezeptsystemknoten im Projektexplorer geöffnet. Der Tabelleneditor stellt den Inhalt des gewählten Knotens dar.

**Knoten "Rezeptklassen"** Im Tabelleneditor werden alle Rezeptdefinitionen des Projektes angezeigt.

**Knoten <Rezeptdefinition>** Im Tabelleneditor werden die im Rezept projizierten Elemente angezeigt.

## 4 Definitionen des Rezeptsystems

Im Folgenden eine Auflistung der Definitionen des VisiWinNET-Rezeptsystems. Zusätzlich wird die Information angegeben, in welcher Produktvariante die Definition vorhanden ist.

Definition	VisiWinNET Standard	VisiWinNET Compact	VisiWinNET Embedded
<b>Rezept</b>	✓	✓	✓
<b>Rezeptelement</b>	✓	✓	✓

### 4.1 Rezept

Rezeptdefinitionen werden im Projektextplorer unterhalb des entsprechenden Zugriffsknotens dargestellt. Sie beinhalten einen Satz Parameter, der in der Projektdatenbank gespeichert wird und dazu dient, dem Rezeptserver mitzuteilen, wie die Rezeptdefinitionen zur Laufzeit zu interpretieren sind.

Rezeptdefinitionen stellen unter anderem folgende Funktionalität zur Verfügung:

- Angabe von digitalen Variablen, die die vier Grundfunktionen "Lesen", "Schreiben", "Laden" und "Speichern" auslösen.
- Relative oder absolute Interpretation der Rezeptwerte über Soll- und Normfaktor.
- Schachtelung von Rezepten über Unterrezepte.



**Für VisiWinNET Smart Anwender:** In Abhängigkeit vom gewählten Zielgerät stehen einige Einstellungen nicht oder nur eingeschränkt zur Verfügung. Weitere Informationen befinden sich im Gerätecatalog.

#### Parameter der Rezeptdefinition

Name	Beschreibung
<b>Änderungshistorie aktivieren</b>	geänderte Werte werden in der Rezeptdatei gesichert.
<b>Beschreibungsvariable</b>	Variable, die einen beschreibenden Text zum Rezept liefert.
<b>Dateiendung</b>	Endung aller Rezeptdateien, die durch diese Rezeptdefinition erzeugt werden.
<b>Definition aktiv</b>	Soll die Definition vom Laufzeitsystem als gültige Definition interpretiert werden.
<b>geänderte Variablenkernwerte direkt speichern</b>	Direktes Speichern in die Rezeptdatei ohne Triggerfunktion.
<b>geänderte Werte direkt schreiben</b>	Direktes Schreiben zur SPS ohne Triggerfunktion.
<b>geänderte Werte direkt speichern</b>	Direktes Speichern in die Rezeptdatei ohne Triggerfunktion.

<b>Kommentar</b>	Wahlfreier Kommentar zur Definition.
<b>Ladetrigger</b>	Variablenbit, dessen Wertänderung zum Laden der Werte aus der Rezeptdatei führt.
<b>Lesetrigger</b>	Variablenbit, dessen Wertänderung zum Lesen der Rezeptwerte aus dem VWOPC-Kernel führt.
<b>Letzte Rezeptdatei beim Start laden</b>	Legt fest, ob die in der letzten Applikationssession zuletzt geladene Rezeptdatei beim Applikationsstart sofort in die entsprechenden Prozessvariablen geladen werden soll
<b>Max. Anzahl Dateien</b>	Legt fest, wieviele Rezeptdateien für diese Definition erzeugt werden dürfen.
<b>Name</b>	Eindeutiger Bezeichner der Definition im Rezeptsystem.
<b>Normfaktor</b>	Multiplikationsfaktor der Rezeptdateidaten bei einer relativen Angabe der Rezeptwerte.
<b>Rezeptdateipfad</b>	Bestimmt den Pfad, unter dem die Rezeptdateien abgelegt werden.
<b>Rezeptwerte absolut/relativ</b>	Bestimmt, ob die Rezeptwerte über Norm- und Sollfaktor umgerechnet werden.
<b>Schreibtrigger</b>	Variablenbit, dessen Wertänderung zum Schreiben der Werte in den VWOPC-Kernel führt.
<b>Sollfaktorvariable</b>	Variable, die den Multiplikationsfaktor der Rezeptserverdaten bei einer relativen Angabe der Rezeptwerte enthält.
<b>Speichertrigger</b>	Variablenbit, dessen Wertänderung zum Speichern der Werte in die Rezeptdatei führt.
<b>Unterrezept-klassenwerte hier speichern</b>	Bestimmt, ob Rezeptwerte aus Unterrezepten mit in der Datei gespeichert werden.
<b>Variable für Dateiname</b>	Bestimmt die Variable, die den Rezeptdateinamen enthält.

### 4.1.1 Rezepte bearbeiten

Rezepte werden im Projektexplorer unter dem Knoten "Rezepte→Rezeptklassen" dargestellt. Jede Rezeptdefinition beinhaltet:

- einen editierbaren Parametersatz: Hier kann der Name oder beispielsweise die Vergabe von Dateinamen und -beschreibung eingestellt werden.
- ggf. Rezeptelemente: Rezeptelemente werden im Tabelleneditor angezeigt. Sie beinhalten die Informationen, welche Prozesswerte in den Rezeptdateien enthalten sind.

Der Editor bietet folgende Funktionen zum Projektieren von Rezepten:

<b>Neu erzeugen</b>	Über Eintrag "Neu" im Kontextmenü des Knotens "Rezeptklassen".
<b>Parameter bearbeiten</b>	Ein Klick auf einen Rezeptknoten lädt die entsprechenden Parameter in die VisiWinNET-Eigenschaftenseite. Hier können die Parameterwerte der Definition bearbeitet werden.
<b>Löschen</b>	Über den Eintrag "Löschen" im Kontextmenü eines Rezeptknotens wird die Definition gelöscht. Dabei werden alle untergeordneten Definitionen (Rezeptelemente) ebenfalls aus dem Projekt entfernt.

## 4.2 Rezeptelement

Rezeptelemente legen fest, welche Variablen zu einem Rezept gehören. Die Verknüpfung kann direkt über die Benennung der Variablen erfolgen oder indirekt durch die Verknüpfung zu einem Unterrezept. Elemente von Unterrezepten können in Abhängigkeit von den Einstellungen der Rezeptdefinition separat in einer Rezeptdatei oder gemeinsam mit den Daten der übergeordneten Rezeptdefinition gespeichert werden.

### Parameter der Rezeptelemente

Name	Beschreibung
<b>Unterrezept</b>	Fügt ein Rezept als Rezeptelement ein.
<b>Variable</b>	Legt eine Variable fest, deren Wert im Rezept enthalten sein soll.
<b>Variable für Dateiname</b>	Bestimmt die Variable, die den Namen der Rezeptdatei enthält, in der die Daten des Unterrezeptes gespeichert werden sollen.

### 4.2.1 Rezeptelemente bearbeiten

Rezeptelemente werden in Rezepten projiziert und im Tabelleneditor angezeigt. Jedes Rezeptelemente zeichnet sich durch einen Prozessvariablenbezeichner aus. Als Rezeptelemente festgelegte Prozessvariablen werden so mit den Funktionen des Rezeptservers verknüpft.

Der Tabelleneditor des Rezeptsystems wird über einen Klick auf einen entsprechenden Knoten im Projektexplorer geöffnet.

Der Editor bietet folgende Funktionen zum Projektieren von Rezeptelementen:

**Neu erzeugen**

Über Eintrag "Neu" im Kontextmenü des Tabelleneditors wird ein Dialog angezeigt, in dem alle Prozessvariablen des Projektes zur Auswahl angeboten werden.

**Löschen**

Eine oder mehrere Rezeptelemente können gelöscht werden, indem:

- zuerst die zu löschenden Definitionen markiert werden (Klick auf die Selektorspalte am linken Tabellenrand ggf. bei gehaltener Strg- oder Shift-Taste für eine Mehrfachselektion)
- Anschließend im Kontextmenü des Tabelleneditors der Eintrag "Löschen" gewählt wird.

## 5 Parameter der Rezeptdefinitionen

Die Beschreibung der Parameter enthält folgende Informationen:

Block	Beschreibung
<b>Parameter für</b>	Listet die Definitionen auf, die diesen Parameter beinhalten.
<b>Beschreibung</b>	Gibt eine Beschreibung der Parameterfunktionalität.
<b>Datenbankfeld</b>	Der Name der Tabellenspalte in der VisiWinNET-Projektdatei.
<b>Datentyp</b>	Der Datentyp des Parameters.
<b>Standardwert</b>	Der Wert, der für den Parameter nach Einfügen einer neuen Definition standardmäßig vergeben wird.
<b>max. Länge</b>	Die maximale Länge möglicher Eingaben.

### 5.1 Parameter in alphabetischer Reihenfolge

Im Folgenden eine alphabetische Auflistung der Parameter. Zusätzlich noch die Information, von welcher VisiWinNET-Produktvariante der Parameter unterstützt wird.

Parametername	VisiWinNET Standard	VisiWinNET Compact	VisiWinNET Embedded
<b>Änderungshistorie aktivieren</b>	✓	✓	✓
<b>Beschreibungsvariable</b>	✓	✓	✓
<b>Dateiendung</b>	✓	✓	✓
<b>Definition aktiv</b>	✓	✓	✓
<b>geänderte Variablenkernwerte direkt speichern</b>	✓	✓	✓
<b>geänderte Werte direkt schreiben</b>	✓	✓	✓
<b>geänderte Werte direkt speichern</b>	✓	✓	✓
<b>Kommentar</b>	✓	✓	✓
<b>Ladetrigger</b>	✓	✓	✓
<b>Lesetrigger</b>	✓	✓	✓
<b>Letzte Rezeptdatei beim Start laden</b>	✓		
<b>Name</b>	✓	✓	✓
<b>Max. Anzahl Dateien</b>		✓	✓
<b>Normfaktor</b>	✓	✓	✓
<b>Rezeptdateipfad</b>	✓	✓	✓
<b>Rezeptwerte absolut/relativ</b>	✓	✓	✓
<b>Schreibtrigger</b>	✓	✓	✓
<b>Sollfaktorvariable</b>	✓	✓	✓

<b>Speichertrigger</b>	✓	✓	✓
<b>Unterrezept</b>	✓	✓	✓
<b>Unterrezept- klassenwerte hier speichern</b>	✓	✓	✓
<b>Variable</b>	✓	✓	✓
<b>Variable für Dateiname</b>	✓	✓	✓

### 5.1.1 Änderungshistorie aktivieren

Parameter für	Rezept
Beschreibung	Das Setzen des Parameters "Änderungshistorie aktivieren" bewirkt, dass nicht nur die aktuellen Rezeptwerte in der Datei gespeichert werden. Auch die älteren überschriebenen Werte bleiben erhalten und können bei Bedarf über die Schnittstellen des Rezeptclients wieder gelesen werden.

### 5.1.2 Beschreibungsvariable

Parameter für	Rezept
Beschreibung	Gibt die Variable (vom Datentyp VT_BSTR) an, deren Wert einen beschreibenden Text zum Rezept enthält. Beim Speichern der Rezeptdatei wird dieser Wert als Beschreibung in der Datei gespeichert. Beim Laden wird die Beschreibung aus der Rezeptdatei in diese Variable übertragen.

### 5.1.3 Dateiendung

Parameter für	Rezept
Beschreibung	Endung aller Rezeptdateien, die durch die Rezeptdefinition erzeugt werden. Beim Speichern neuer Rezeptdateien wird automatisch vor die angegebene Endung ein "R" gestellt. Beim Laden ist über die Endung eine Filterung in der Dateiauswahlansicht möglich.

### 5.1.4 Definition aktiv

Parameter für	Rezept
Beschreibung	Bestimmt, ob die Definition vom Laufzeitsystem als gültig anerkannt wird. Die Deaktivierung des Parameters ist gleichbedeutend mit einem Auskommentieren.

### 5.1.5 geänderte Variablenkernwerte direkt speichern

Parameter für	Rezept
Beschreibung	Der Parameter "geänderte Variablenkernwerte direkt speichern" bewirkt, dass im Variablenkern geänderte Werte direkt (also ohne Auslösen der Triggerfunktion "Speichern") über den Rezeptserver in die Rezeptdateien geschrieben werden.

### 5.1.6 geänderte Werte direkt schreiben

Parameter für	Rezept
Beschreibung	Der Parameter "geänderte Werte direkt schreiben" bewirkt, dass über die Applikation (z.B. in den VisiWinNET-Steuerelementen) geänderte Werte direkt (also ohne Auslösen der Triggerfunktion "Schreiben") über den Variablenkernel an die SPS weitergesendet werden.

### 5.1.7 geänderte Werte direkt speichern

Parameter für	Rezept
Beschreibung	Der Parameter "geänderte Werte direkt speichern" bewirkt, dass über die Applikation (z.B. in den VisiWinNET-Steuerelementen) geänderte Werte direkt (also ohne Auslösen der Triggerfunktion "Speichern") über den Rezeptserver in die Rezeptdateien geschrieben werden.



Die geänderten Prozesswerte werden dabei in die zuletzt verwendete Rezeptdatei geschrieben. Damit die Werte gespeichert werden können, muss also vorher mindestens einmal die "Laden" oder die "Speichern"-Funktion ausgeführt worden sein.

### 5.1.8 Kommentar

Parameter für	Rezept
Beschreibung	Zu jeder Rezeptdefinition steht ein wahlfreies Kommentarfeld zur Verfügung. Kommentare dienen zur Entwicklungszeit als Hilfestellung für den Entwickler. Sie haben zur Laufzeit keine Funktion.

### 5.1.9 Ladetrigger

Parameter für

Rezept

Beschreibung

Bestimmt eine Variable, deren Wertänderung auf "1" das Laden von Werten aus einer Rezeptdatei in den Rezeptserver auslöst. Nach dem Laden wird der Wert der Variablen automatisch auf "0" zurückgesetzt. Die Variable kann entweder einen booleschen oder einen ganzzahligen Datentypen haben. Über die Festlegung der Bitnummer auf dem Variablenauswahldialog wird dann das als Trigger dienende Bit ausgewählt.

Der Name der zu ladenden Datei wird in einer Prozessvariablen erwartet, die im Parameter "Variable für Dateinamen" anzugeben ist.

### 5.1.10 Lesetrigger

Parameter für

Rezept

Beschreibung

Bestimmt eine Variable, deren Wertänderung auf "1" das Lesen von Werten aus dem Variablenkernel in den Rezeptserver auslöst. Nach dem Laden wird der Wert der Variablen automatisch auf "0" zurückgesetzt. Die Variable kann entweder einen booleschen oder einen ganzzahligen Datentypen haben. Über die Festlegung der Bitnummer auf dem Variablenauswahldialog wird dann das als Trigger dienende Bit ausgewählt.

### 5.1.11 Letzte Rezeptdatei beim Start laden

Parameter für

Rezept

Beschreibung

Legt fest, ob die in der letzten Applikationssession zuletzt geladene Rezeptdatei beim Applikationsstart sofort in die entsprechenden Prozessvariablen geladen werden soll.

Nach dem Start der Applikation ist der Rezeptpuffer einer Rezeptklasse üblicherweise leer. Über die hier beschriebene Option kann der Rezeptpuffer mit der zuletzt geladene Rezeptdatei initialisiert werden. Direkt anschließend werden die Werte automatisch in die entsprechenden Prozessvariablen und damit zur SPS übertragen.

Das System merkt sich zwischen den Applikationsläufen den Namen der zuletzt geöffneten Rezeptdatei in der Projektdatenbank.

### 5.1.12 Max. Anzahl Dateien

Parameter für	Rezept
Beschreibung	<p>Legt fest, wieviele Dateien für diese Rezeptdefinition angelegt werden dürfen. Diese Einstellung ist ein Schutzmechanismus für Systeme mit wenig Speicherkapazitäten.</p> <p>Das System prüft beim Speichern einer Rezeptdatei die Anzahl der vorhandenen Dateien anhand der Dateiendung und dem Dateipfad. Bei Überschreitung der maximalen Anzahl Dateien meldet das System den Fehler "EceededMaxFiles" im Ereignis "SaveDone".</p>

### 5.1.13 Name


Parameter für	Rezept
Beschreibung	Der eindeutige Bezeichner der Definition in der Projektdatenbank.

### 5.1.14 Normfaktor

Parameter für	Rezept
Beschreibung	<p>Rezeptwerte können absolut oder relativ interpretiert werden. Bei der absoluten Interpretation werden die Werte direkt mit dem VWOPC-Kernel ausgetauscht. Die relative Interpretation ermöglicht über den Normfaktor und die Sollfaktorvariable die Umrechnung bezogen auf eine "Normmenge". Die Werte werden nach folgender Formel umgerechnet:</p>

$$\text{Wert im VWOPC-Kernel} = \frac{\text{Normfaktor} \times \text{Wert im Rezeptserver}}{\text{Wert der Sollfaktorvariable}}$$

### 5.1.15 Rezeptdateipfad

Parameter für	Rezept
Beschreibung	<p>Bestimmt den Pfad, unter dem die Rezeptdateien gespeichert/geladen werden. Die Pfadangabe kann sowohl absolut (die Schaltfläche  öffnet eine Dialog zur Auswahl eines Verzeichnisses) als auch relativ zum Projektverzeichnis (das Schlüsselwort "&lt;ProjectDir&gt;" leitet die relative Pfadangabe ein z.B.: "&lt;ProjectDir&gt;\Rezepte") angegeben werden.</p>

### 5.1.16 Rezeptwerte absolut/relativ

Parameter für	Rezept
Beschreibung	<p>Rezeptwerte können absolut oder relativ interpretiert werden. Bei der absoluten Interpretation werden die Werte direkt mit dem Variablenkern ausgetauscht. Die relative Interpretation ermöglicht über den Normfaktor und die Sollfaktorvariable die Umrechnung bezogen auf eine "Normmenge".</p>

### 5.1.17 Schreibtrigger

Parameter für

Rezept

Beschreibung

Bestimmt eine Variable, deren Wertänderung auf "1" das Schreiben von Werten aus dem Rezeptserver in den Variablenkernel auslöst. Nach dem Laden wird der Wert der Variable automatisch auf "0" zurückgesetzt. Die Variable kann entweder einen booleschen oder einen ganzzahligen Datentypen haben. Über die Festlegung der Bitnummer auf dem Variablenauswahldialog wird dann das als Trigger dienende Bit ausgewählt.

### 5.1.18 Sollfaktorvariable

Parameter für

Rezept

Beschreibung

Rezeptwerte können absolut oder relativ interpretiert werden. Bei der absoluten Interpretation werden die Werte direkt mit dem Variablenkernel ausgetauscht. Die relative Interpretation ermöglicht über den Normfaktor und die Sollfaktorvariable die Umrechnung bezogen auf eine "Normmenge". Die Werte werden nach folgender Formel umgerechnet:

$$\text{Wert im WWOPC-Kernel} = \frac{\text{Normfaktor} \times \text{Wert im Rezeptserver}}{\text{Wert der Sollfaktorvariable}}$$

### 5.1.19 Speichertrigger

Parameter für

Rezept

Beschreibung

Bestimmt eine Variable, deren Wertänderung auf "1" das Speichern von Werten aus dem Rezeptserver in die Rezeptdatei auslöst. Nach dem Laden wird der Wert der Variablen automatisch auf "0" zurückgesetzt. Die Variable kann entweder einen booleschen oder einen ganzzahligen Datentypen haben. Über die Festlegung der Bitnummer auf dem Variablenauswahldialog wird dann das als Trigger dienende Bit ausgewählt.

Der Name der zu speichernden Datei wird in einer Prozessvariablen erwartet, die im Parameter "Variable für Dateinamen" anzugeben ist.

### 5.1.20 Unterrezept

Parameter für

Beschreibung

Rezeptelement

Gibt den Namen einer Rezeptdefinition an, die als Unterrezept eingebunden werden soll.

#### **Funktion von Unterrezepten**

Unterrezepte sind immer dann sinnvoll, wenn ein Rezept modular zerlegt werden kann. Die Kombination der einzelnen Module über ein "Hauptrezept" (Rezept, das die einzelnen Module als Unterrezepte enthält) ergibt dadurch immer neue Produktionsvarianten. Ein Beispiel (quasi aus dem Kochbuch) wäre die Herstellung von unterschiedlichen Kuchenvarianten, die in die Module "Boden" und "Belag" zerlegt werden können. Über die Kombination der beiden Module sind alle möglichen Gebäckformen von der Erdbeertorte über den Blaubeerkuchen bis hin zum Schokozwieback zu kombinieren. Die Module (oder Unterrezepte) beinhalten hier

für den Belag :

- Erdbeere
- Blaubeere
- Schoko

Für den Boden:

- Torte
- Mürbeteig
- Zwieback

Aus der Zusammenstellung der einzelnen Unterrezepte lässt sich auch der (zugegeben exotische) Erdbeerzwieback herstellen. Das hierfür zu ladende Hauptrezept umfasst die Namen der entsprechenden Unterrezeptdateien. Die Unterrezeptdateien enthalten die eigentlichen Maschinenparameter der einzelnen Module.

Änderungen in einem Unterrezept fließen damit in alle Hauptrezepte ein, die dieses Unterrezept verwenden. Einerseits erleichtert dies wesentlich die Datenhaltung. Andererseits muss beachtet werden, dass bei einer etwaigen Nachweispflicht (FDA) besondere Vorkehrungen getroffen werden müssen. Hier dürfen Änderungen in einem Unterrezept nicht ungemeldet dadurch in die weitere Produktion einfließen, dass ein Hauptrezept verwendet wurde, das das geänderte Unterrezept einbezieht.

## Transferverhalten von Unterrezepten

Der Transfer der Unterrezeptwerte zwischen Prozess, Rezeptpuffer und Rezeptdateien ist so ausgeführt, dass Unterrezeptwerte nicht versehentlich durch die Transferfunktionen des Hauptrezeptes überschrieben werden können.

- **Laden:** Das Laden eines Hauptrezeptes lädt automatisch alle Unterrezeptelemente in den Rezeptpuffer
- **Speichern:** Das Speichern schreibt nur die Hauptrezeptwerte in die Rezeptdatei. Um Unterrezepte zu speichern, muss die entsprechende "Speichern"-Methode des Unterrezeptes aufgerufen werden.
- **Lesen:** Beim Lesen werden nur die Hauptrezeptelemente vom Variablenkern in den Rezeptpuffer übertragen.
- **Schreiben:** Das Schreiben umfasst sowohl die Haupt- als auch die Unterrezeptelemente
- **Editiermodus:** Die Umschaltung der Steuerelemente zwischen Variablenkern und Rezeptpuffer über das Hauptrezept betrifft nur die Elemente des Hauptrezeptes.

Aus diesem Transferverhalten ergibt sich, dass in der Applikation für jedes Unterrezept die entsprechende Administration der Rezeptdateien (Anlegen neuer Dateien, Bearbeiten von Variablenwerten im Rezeptpuffer, ...) auszuführen ist. Üblicherweise wird für jedes Unterrezept ein eigenes Formular projektiert, auf der die Werte bearbeitet werden können und auf dem die Transferfunktionen für die Unterrezeptwerte angesteuert werden können.

Die Administration des Hauptrezeptes umfasst alle Elemente, also neben den Rezeptvariablen auch die Dateinamen der einzubeziehenden Unterrezeptdateien. Auch hier sind entsprechende Funktionen für den Transfer vorzusehen.

Ein Beispiel für den Aufbau eines Applikationsformulars, das sowohl die Werte als auch die Transfer-Steuerelemente beinhaltet, befindet sich im Beispielprojekt "VWNRecipeDemo".

### 5.1.21 Unterrezeptklassenwerte hier speichern

Parameter für	Rezept
Beschreibung	Bestimmt, ob die Unterrezeptwerte in der übergeordneten Datei abgespeichert werden.
Einstellung	Beschreibung
<b>False</b>	Die Rezeptwerte werden in der Datei des Unterrezeptes gespeichert. Der Parameter "Dateiname" in der als Unterrezept dienenden Rezeptdefinition gibt dabei den Namen der Datei an.
<b>True</b>	Die Werte werden in der übergeordneten Rezeptdatei gespeichert. Der Parameter "Dateiname" in der als Unterrezept dienenden Rezeptdefinition hat keine Funktion.

### 5.1.22 Variable

Parameter für	Rezeptelement
Beschreibung	Legt die Variable fest, deren Wert im Rezept enthalten sein soll. Über die Festlegung eines gültigen Variablenbezeichners wird der entsprechende Wert in die Funktionen der Rezeptdefinition aufgenommen.

### 5.1.23 Variable für Dateiname

Parameter für	Rezept, Rezeptelement
Beschreibung	Legt fest, welche Variable vom Datentyp VT_BSTR den Rezeptdateinamen enthält. Unter diesem Namen wird das Rezept gespeichert/geladen.



Der Inhalt der "Variable für Dateiname" wird nur verwendet, wenn die Triggerfunktionen (Ladetrigger, Speichertrigger) der Rezeptdefinition ausgelöst werden. Bei der Verwendung eines 'RecipeClassHandler' oder eines 'RecipeClass'-Objektes hat die "Variable für Dateinamen" keine Funktion.

Die Daten eines Unterrezeptes können entweder als eigene Datei oder in dem übergeordneten Rezept gespeichert werden. Wird der Parameter "Unterrezeptklassenwerte hier speichern" aktiviert, so ist der Parameter "Dateiname" ohne Bedeutung.

## 6 Konfiguration des Rezeptservers

Die Konfiguration des Rezeptservers ermöglicht systemweite Einstellungen.

Der Konfigurationsdialog des Rezeptsystems wird über den gleichnamigen Projektexplorerknoten aufgerufen. Die Einstellungen werden in der VisiWinNET-Eigenschaftenseite dargestellt.

Einstellung	Beschreibung
<b>Anzahl gleichzeitiger Aufträge</b>	Legt die Anzahl von Aufträgen fest, die der Rezeptserver puffern kann. Ein Auftrag ist dabei die Anforderung einer Transferfunktion (Laden, Speichern, Lesen oder Schreiben eines Rezeptes). Die Anforderung von Transferfunktionen werden gepuffert und nacheinander abgearbeitet.
<b>XML/MS-Access</b>	Legt das Format der Rezeptdateien fest. Es kann zwischen XML (Extended markup language)-Dateien und Access-Datenbanken gewählt werden.
<b>Ok</b>	Bestätigt die Eingaben zur Übernahme in die Projektdatei und schließt den Dialog.
<b>Abbrechen</b>	Verwirft die Eingaben und schließt den Dialog.