



Studio Dokumentation

Copyright AVI GmbH

Version 2.2 - 2007

Inhaltsverzeichnis

1. Studio	1
1.1. Leistungsumfang von Studio	1
2. Technologische Vorhaben	2
3. Entwurfsprozess	3
3.1. Prozesszerlegung	3
3.2. Ziele und Inhalt der Prozesszerlegung	3
3.3. Kriterien der Prozesszerlegung	4
3.4. Schritte der Prozesszerlegung	4
3.5. Prozesszerlegungsgraph	5
4. Alle Module	7
5. Projektverwaltung	9
5.1. Funktionen	9
5.2. Variablen	9
5.3. Übersetzung	10
5.4. Prozessordner	10
5.4.1. Auftragsdaten	10
5.4.2. Übersetzung	11
5.4.3. Variablennamen	13
5.4.4. Alle Eigenschaften	15
5.4.5. Aufrufblock	16
5.4.6. Resource	17
5.4.7. Datenbank	18
6. Optionen	21
6.1. Drucken	21
6.2. Allgemein	22
7. Drucken	24
8. System Informationen	25
8.1. Technische Voraussetzungen	25
8.2. Installation und Deinstallation	25
8.3. Hardwaremäßiger Softwareschutz	25
8.3.1. Umfang der Lizenzen	26
8.3.2. Treiber	26
8.3.3. Hilfsprogramme	26
8.3.4. Updates	26
8.4. Einschränkungen der Demo-Version	26
8.5. Bezugsquellen	27
9. Referenz	28
9.1. Menü und Schaltflächen	28

Abbildungsverzeichnis

4.1. Teranis Studio	7
4.2. Teranis Studio Komplex	7
5.1. Eigenschaften - Auftragsdaten	10
5.2. Eigenschaften - Übersetzung	12
5.3. Eigenschaften Variablennamen	14
5.4. Eigenschaften - Alle Eigenschaften	15
5.5. Eigenschaften Aufrufblock	16
5.6. Eigenschaften - Resource	18
5.7. Eigenschaften - Datenbank	18
6.1. Optionen - Drucken	21
6.2. Optionen - Allgemein	23

Tabellenverzeichnis

9.1. Menü und Schaltflächen 28

Kapitel 1. Studio

Studio ist die Projektverwaltung von TeRANiS. Da Projekte meist aus vielen Teilprozessen bestehen, ist es wichtig diese übersichtlich zu verwalten und Funktionen für mehrere Teilprozesse aufzurufen.

Es ist möglich alle Schritte auch ohne Studio mit den Teilmodulen auszuführen, dies ist jedoch mit hohem manuellen Aufwand verbunden.

1.1. Leistungsumfang von Studio

- Zusammenfassung der Funktionen in einer Projektverwaltung
- Hierarchische Anordnung von Ressourcen (SPS) und Bibliotheken und Teilprozessen (Exploreransicht)
- Verwalten aktivierter und deaktivierter Teilprozesse
- Aufrufen der anderen Teilmodule
- Aufrufen von Sammelfunktionen über Teilprozessbäume: Variablen (lesen/schreiben/prüfen), Übersetzen, Transferieren
- Verwalten der Eigenschaften von Prozessordnern
- Generieren von Aufruffunktionen für untergeordnete Teilprozesse mit Initialisierung

Kapitel 2. Technologische Vorhaben

Voraussetzung für die erfolgreiche Realisierung von technologischen Vorhaben ist, dass Technologen und Steuerungsprojektanten eng zusammenarbeiten. Während der Technologe sich genauestens mit den Prozessen industrieller Anlagen auskennt, betrachtet der Automatisierungstechniker den Prozess von der Seite der Steuerungstechnik. Mit der Entwicklung des Programmpaketes TeRANiS wird ein Schritt auf dem Weg gegangen, die Kluft der Verständigung zwischen der verfahrensorientierten Prozessbeschreibung und der Prozessautomatisierung zu verringern. TeRANiS ist in seinen Beschreibungsmitteln so angelegt, dass der Verfahrenstechniker die verfahrenstechnisch relevanten Begriffe in einen Prozessablaufplan einbringen kann. Damit wird für den Automatisierungs-Techniker zugleich die Anpassung an die Steuerungsaufgabe erleichtert. Es ist also Ziel, ein gemeinsam verständliches Beschreibungsmittel zu verwenden. Ausserdem wird dem Automatisierer ein hocheffektives Engineeringwerkzeug in die Hand gegeben. Damit kann bei der Erstellung der Projektierung gegenüber herkömmlicher Programmierung zusätzlich Zeit gespart werden. Das Programmpaket TeRANiS kann insbesondere genutzt werden von

- Verfahrenstechnikern
- Technologen
- Anlagen - und Maschinenbauern
- Automatisierungstechnikern
- Elektrotechnikern

Es kann aber auch von Fachleuten anderer Sachgebiete genutzt werden, die sich die Aufgabe stellen, ihre Probleme zu analysieren und aufzubereiten.

Kapitel 3. Entwurfsprozess

3.1. Prozesszerlegung

Ein komplexer verfahrenstechnischer Prozess gliedert sich in eine Vielzahl von Teilprozessen. Die Gliederung erfolgt hierarchisch. Grundlage für die Prozessanalyse ist der reale technologische Prozess. Jeder Teilprozess wird in einer Datei abgelegt. Eine Ausnahme bilden TeList-Tabellen in denen jede enthaltene Instanz einen Teilprozess darstellt.

Ausgangsdokumente sind im Normalfall:

- Technologieschema (R&I)
- grundsätzliche verbale Beschreibung des Prozesses

Das Grundanliegen der Prozesszerlegung ist die Erzeugung von Teilprozessen. Diese Teilprozesse weisen ihrerseits eine in sich geschlossene Funktionalität auf, sind jedoch in der Lage mit anderen Teilprozessen über verbindende Elemente (Variablen) zu kommunizieren.

Es ist dem Geschick des Ingenieurs überlassen, diese Teilprozesse sinnfällig und überschaubar zu organisieren.

Zwischen den Teilprozessen findet ein Signalaustausch statt. Es werden Informationen über den Zustand des jeweiligen Teilprozesses ausgetauscht.

Hat also ein Teilprozess einen bestimmten Zustand erreicht, so wird diese Information an einen anderen Teilprozess weitergegeben, der darauf reagieren soll. Der Teilprozess fragt in einem bestimmten Prozessabschnitt diese Information ab, prüft sie und reagiert nun mit weiterführenden Aktionen im Steuerungsablauf. Er meldet seinerseits aber auch zurück, dass er einen neuen wichtigen Prozessabschnitt erreicht hat.

Die hierarchische Gliederung erfolgt mit TeRANiS in einem Verzeichnisbaum.

Wegen der meist grossen zu automatisierenden Prozesse ist somit eine Prozesszerlegung unumgänglich.

Ziele und Inhalt der Prozesszerlegung

Kriterien der Prozesszerlegung

Schritte der Prozesszerlegung

Prozesszerlegungsgraph

3.2. Ziele und Inhalt der Prozesszerlegung

Ziel der Prozesszerlegung ist es, einen Überblick über das gesamte Steuerungsproblem (den System-

entwurf) zu erhalten und Steuerungs- und Regelungsaufgaben getrennt analysieren zu können, ohne dabei die gegenseitigen steuerungstechnischen Beziehungen ausser acht zu lassen.

Aus diesen Gründen ist es notwendig, dass Sie sich ein sinnvolles hierarchisches System von Teilprozessen schaffen. Dabei können Teilprozesse auch Stabilisierungsprobleme für Prozessgrößen beinhalten.

Es ist zweckmässig die Prozesszerlegung der Anlagenstruktur anzupassen.

Jeder Teilprozess enthält prinzipiell die drei Komponenten des Prozesses, wie sie im Informationsflussmodell dargestellt sind. Reine Bedien-Teilprozesse und Signalisierungs-Teilprozesse beinhalten häufig keinen Anteil des technologischen Prozesses.

Das entstehende Teilprozess-System benötigt Koppelgrößen, um die Teilprozesse miteinander zu verbinden. Dabei werden Ausgangsgrößen eines Teilprozesses als Eingangsgrößen eines oder mehrerer Teilprozesse verwendet. Damit ist es prinzipiell nur möglich eine Variable in genau einem Teilprozess als Ausgang zu verwenden.

Insbesondere bei gegenseitiger Kopplung von Teilprozessen ist auf die Dynamik der Größen zu achten.

Die Abarbeitungsreihenfolge oder die Taskklasse eines Teilprozesses sollte für die Funktionalität unerheblich sein. Eine Änderung der Abarbeitungsreihenfolge sollte nur aus Performancegründen im Zielsystem nötig werden.

Die Beziehungen zwischen den Teilprozessen werden im [Prozesszerlegungsgraphen](#) dargestellt.

Die Prozesszerlegung wird in mehreren [Schritten](#) ausgeführt.

3.3. Kriterien der Prozesszerlegung

Die Zerlegung des Gesamtprozesses in Teilprozesse erfolgt nach drei Kriterien:

Der Prozessablaufplan wird als Steuerungsmodell bei parallel laufenden Steuerungsanforderungen für den Entwickler bald unübersichtlich. Deshalb lohnt es sich, mehr als einen PRAP zu entwickeln und somit das Problem weiter zu zerlegen.

Jeder Teilprozess kann sinnvoll einem Anlagenteil oder Anlagenabschnitt aus dem Technologischen Schema zugeordnet werden. Dabei spielt der Aufbau aus Behältern, Transportbändern, Reaktoren, Förderschnecken usw. eine wesentliche Rolle, da die Struktur der Prozesszerlegung dem Anlagenschema weitestgehend entspricht.

Alle 'gleichartigen' Teilprozesse (Antriebssteuerungen, Signalisierungen u.a.) sollten aus dem Anlagenschema bzw. der Aufgabenformulierung besonders entnommen und für die Prozesszerlegung vorgemerkt werden. Häufig kann man für solche Steuerungsanforderungen Typen-Teilprozesse entwerfen und verwenden bzw. Prozessablaufpläne nutzen, die schon existieren.

Es ist sinnvoll für gleichartige Teilprozesse eine Bibliothek anzulegen, die auch projektübergreifend benutzt werden kann.

3.4. Schritte der Prozesszerlegung

Der erste Schritt sollte die Einteilung des Gesamtprozesses an Hand eines technologischen Anlagenschemas betreffen. Die so gewonnenen Prozess-Abschnitte, die aus einem oder mehreren Anlagenteilen (Behälter, Maschinen usw.) bestehen, beinhalten sinnvollerweise je einen Prozessschritt.

Falls Sie für einen Batchprozess das Steuerungskonzept entwickeln, sollten Sie zuerst die parallel laufenden Teile voneinander trennen. Normalerweise zerfällt das Problem in einen übergeordneten Steuerprozess und mehrere Hilfsprozesse, die mit dem Steuerprozess über Koppelvariablen verbunden sind.

Falls Sie für einen Fließprozess das Steuerungskonzept entwickeln, bedenken Sie, dass nach dem Anlauf alle Prozessabschnitte simultan bzw. mehr oder weniger parallel arbeiten. Betrachten Sie jeden Prozessabschnitt und den(die) darin technologisch enthaltenen Prozessschritt(e) als Teilprozess des ersten Schrittes.

Der zweite Schritt soll die beiden folgenden Überlegungen enthalten:

a)

Gibt es steuerungstechnische Beziehungen zwischen den Teilprozessen, dann sollte für jede Beziehung eine Variable mit passendem Namen angelegt werden.

b)

Stellt sich bei näherem Betrachten eines Teilprozesses des ersten Schrittes heraus, dass eine weitere Zerlegung sinnvoll ist, dann sollte dies geschehen.

Als dritten Schritt einer sinnvollen Prozesszerlegung sollten Sie eine Einteilung nach den einzelnen Ebenen des PZG vornehmen. Jedem dabei entstehenden Teilprozess ist eine technologisch sinnvolle Bezeichnung zuzuordnen. Aus Gründen der Übersichtlichkeit sollten Sie für Prozesse, die zu einer technologischen Einheit gehören, Prozessordner anlegen.

Im vierten Schritt sollten Sie zur Dokumentation Ihrer bisherigen Arbeit den **Prozesszerlegungsgraphen** zeichnen.

3.5. Prozesszerlegungsgraph

Der Prozesszerlegungsgraph beschreibt die steuerungstechnischen Beziehungen zwischen allen Teilprozessen eines Prozesses durch deren paarweiser Steuerrelationen.

Jeder Teilprozess wird im PZG als Rechteck und jede Steuerrelation als gerichtete Verbindung (Pfeil) zwischen den Teilprozessen dargestellt. Bei der Anordnung der Teilprozesse sollten Sie nach dem folgenden Konzept verfahren:

- Teilprozesse der Mensch-Maschine-Kommunikation werden oben gezeichnet
- Teilprozesse für automatische Abläufe bzw. zur Koordinierung werden in der Mitte gezeichnet
- Teilprozesse der Stellglieder und Antriebe werden unten gezeichnet

In einer späteren TeRANiS Version wird der Prozesszerlegungsgraph als Modul enthalten sein.

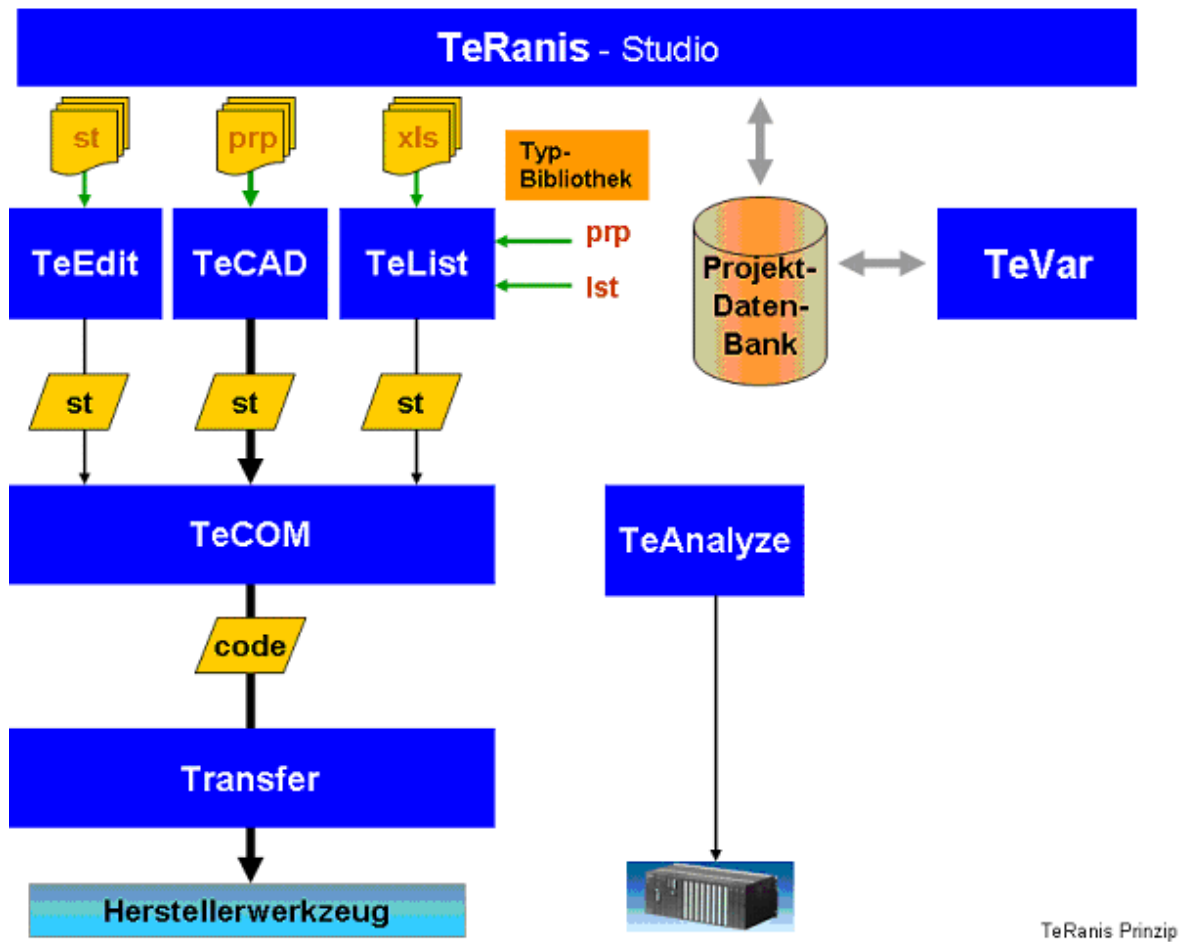
siehe auch: [Schritte der Prozesszerlegung](#)

Kapitel 4. Alle Module

Eine prinzipielle Übersicht zeigt folgendes Bild.

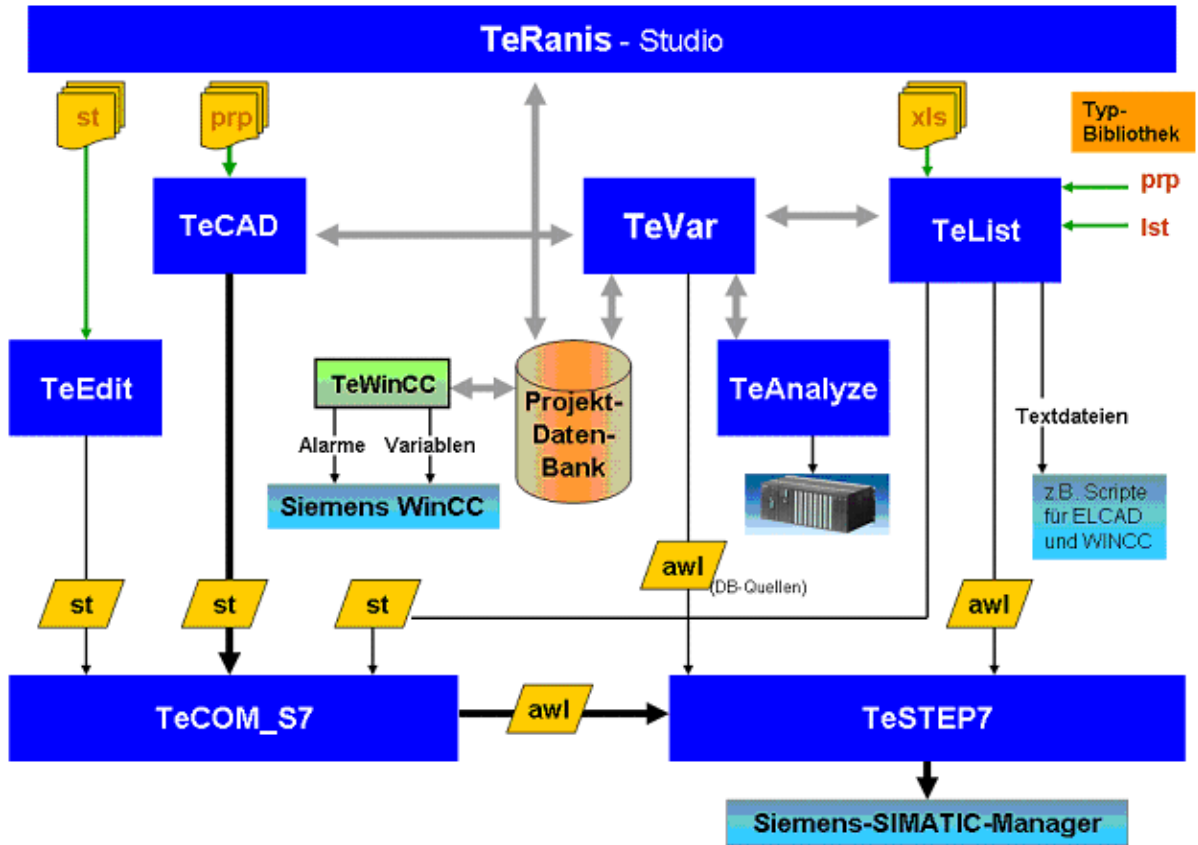
Die drei Codegeneratoren TeEdit, TeCAD und TeList erzeugen alle strukturierten Text, der über Compiler und Transfer in ein Zielsystem gebracht wird.

Abbildung 4.1. Teranis Studio



In der Realität sind die Verknüpfungen etwas komplexer. Es gibt Sonder- und Zusatzfunktionalitäten, z.B. kann TeList auch andere Zielcodes generieren.

Abbildung 4.2. Teranis Studio Komplex



Kapitel 5. Projektverwaltung

Das TeRANiS-Projekt wird in einer Baumstruktur, die der Verzeichnisstruktur entspricht angezeigt. Die Wurzel dieses Baums bildet die Projektdatei bzw. die Projektdatenbank.

In der ersten Gliederungsebene sind nur Bibliotheken und Ressourcen erlaubt. Sie können manuell auch weitere Ordner zu anderen Verwaltungszwecken anlegen. Diese werden jedoch nicht in Studio angezeigt.

Alle untergeordneten Ordner werden **Prozessordner** genannt. Sie dienen zur Gliederung des technologischen Prozesses oder unterteilen verschiedene Aufgaben, die die SPS zu erfüllen hat. Die Variablenverwaltung TeVar benutzt die Ordnerstruktur gleichzeitig um den Gültigkeitsbereich von Variablen zu bestimmen.

In den Prozessordnern können Teilprozesse verschiedenen Typs abgelegt werden. Es besteht die Einschränkung, dass ein untergeordneter Ordner oder Teilprozess nicht genauso heissen darf wie der Ordner, da dieser Name für die Ordnerereigenschaften reserviert ist.

TeRANiS legt verschiedene Dateien und Ordner zu Verwaltungszwecken im Verzeichnisbaum ab, die in Studio nicht angezeigt werden.

Zur Erleichterung der Integration einer Versionsverwaltung werden die Ordner CVS oder .svn ebenfalls nicht angezeigt. Damit können Sie einfach das TeRANiS-Projekt mit einer externen Versionsverwaltung wie SubVersion verwalten.

5.1. Funktionen

In der Baumansicht können über das Popup-Menü über die rechte Maustaste verschiedene Funktionen aufgerufen werden. Sofern die Funktionen Teilprozesse betreffen ruft dabei die Funktionen der Einzelmodule über die OLE-Schnittstelle auf. Die entsprechenden Programme werden dazu geöffnet und im rechten Bereich von Studio angeordnet. Die Funktionen der Module verhalten sich dabei so wie sie auch als Einzelfunktionen in den Modulen aufgerufen werden können.

Fehlermeldungen werden jedoch nicht als Meldung, sondern im unteren Bereich von Studio in einer Liste angezeigt.

Durch Doppelklick auf einen Listeneintrag wird der fehlerhafte Teilprozess geöffnet. Eventuell muss dort die Funktion nochmals ausgeführt werden um den Fehler genauer zu lokalisieren.

5.2. Variablen

Die Funktionen

- aus DB lesen
- in DB schreiben
- mit DB abgleichen

können aufgerufen werden.

Zusätzlich werden die Prozessordner bearbeitet.

Die Funktionen verhalten sich in allen Modulen gleich und sind dort genau beschrieben.

5.3. Übersetzung

Die Übersetzung läuft in mehreren Schritten ab. So wird z.B. zuerst der ST-Code aus dem Teilprozess generiert und danach aus dem ST-Code der Zielcode für das entsprechende Zielsystem.

Wenn nur geänderte Teilprozesse übersetzt werden, wird ein Teilprozess immer dann übersetzt, wenn die Quelldatei jünger ist, als die zu generierende Datei oder diese Datei noch nicht vorhanden ist.

Dabei werden immer zuerst der Inhalt eines Prozessordners und dann erst der Ordner selbst übersetzt.

Treten Fehler bei der Übersetzung auf, werden diese in der Meldungsliste angezeigt. In manchen Fällen wird ein Fehler nicht schon bei der Generierung des ST-Codes erkannt, sondern erst bei der Generierung des Zielcodes. Falls die Fehlermeldung nicht selbsterkändig ist, muss dieser Fehler mit TeEdit im ST-Code gesucht werden. TeEdit können Sie direkt mit Shift-Doppelklick auf den Listeneintrag aufrufen.

5.4. Prozessordner

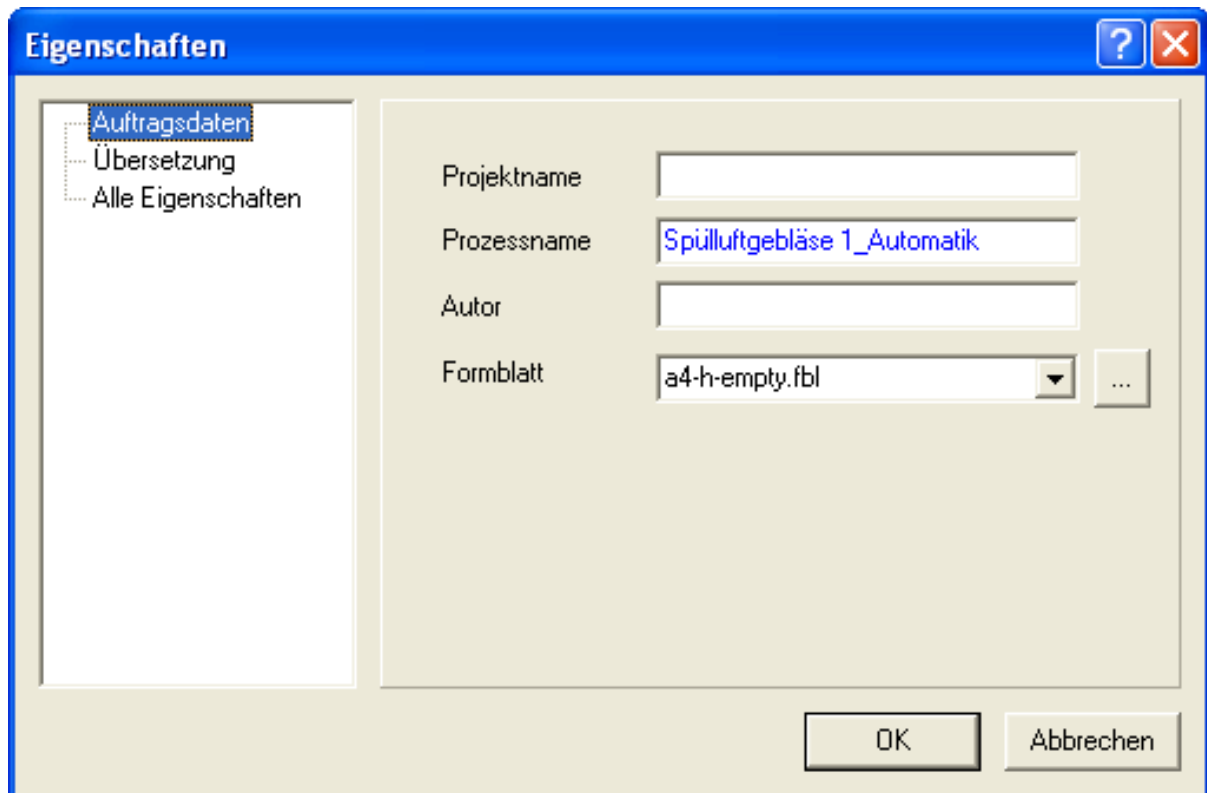
Zum Einstellen der Eigenschaften müssen Sie auf dem Kontextmenüpunkt **Eigenschaften** klicken. Im Eigenschaftsdialog befinden sich je nach Prozessordner bestimmte Registereinträge ([Auftragsdaten](#) , [Übersetzung](#) , [Variablennamen](#) , [Alle Eigenschaften](#), [Aufrufblock](#), [Resource](#) und die [Datenbank](#)) die durch Klicken auf den jeweiligen Eintrag angezeigt werden. Änderungen werden erst beim Verlassen des Dialogs durch die **OK** Schaltfläche übernommen.

5.4.1. Auftragsdaten

Der Eigenschaftsdialog eines Prozessordners kann über das Kontextmenü **Eigenschaften** des jeweiligen Eintrages im Projektverzeichnis aufgerufen werden.

Die Auftragsdaten dienen vor allem der Verwaltung und Beschriftung der Formblattfelder beim Drucken:

Abbildung 5.1. Eigenschaften - Auftragsdaten



Im Register **Auftragsdaten** können folgende Einstellungen vorgenommen werden:

- **Projektname**

Der Projektname ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden.

- **Prozessname**

Der Prozessname kann nur lokal eingestellt bzw. verändert werden.

- **Autor**

Der Autor ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden.

- **Formblatt**

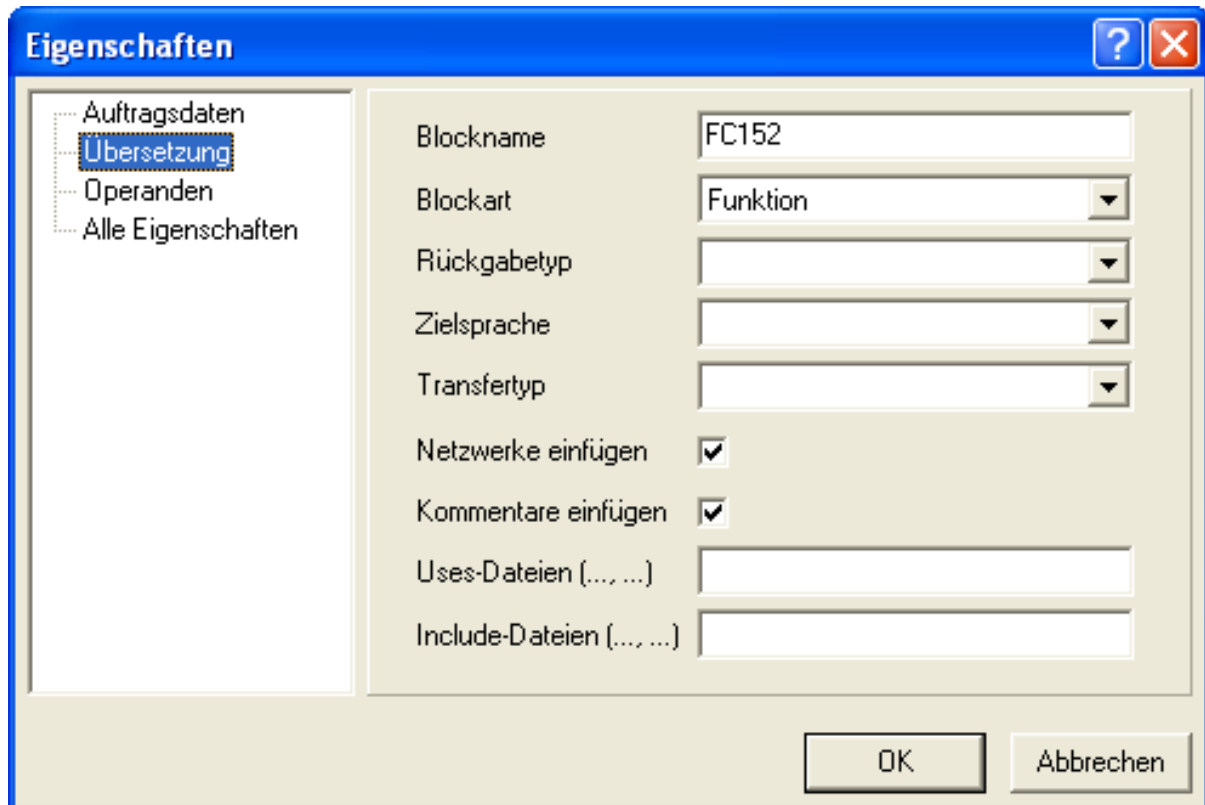
Das Formblatt ist eine hierarchische Eigenschaft und muss nicht in jedem Unterordner neu eingetragen werden. Es stehen nur die Formblätter zur Auswahl, die in Ihrem Projektverzeichnis vorhanden sind. Wenn Sie andere Formblätter verwenden wollen, dann können Sie diese über den rechten Button '...' in Ihr Projektverzeichnis importieren.

5.4.2. Übersetzung

Der Eigenschaftsdialog eines Prozessordners kann über das Kontextmenü **Eigenschaften** des jeweiligen Eintrages im Projektverzeichnis aufgerufen werden.

Die Eingabefelder der Übersetzung legen fest, wie und in welche Sprache der Quellcode erzeugt werden soll.

Abbildung 5.2. Eigenschaften - Übersetzung



Im Dialog können folgende Einstellungen vorgenommen werden:

- Blockname

Es muss mindestens ein Block angegeben werden und muss vom Zielsystem bekannt sein. Bei STEP-7 kann z.B. direkt ein FC oder FB angegeben oder ein symbolischer Name verwendet werden. Im Blocknamen dürfen keine Leerzeichen o. ä. enthalten sein.

- Blockart

Dient zur Angabe, ob eine Funktion, ein Funktionsblock oder ein Programm generiert werden soll. Der Funktionsblock hat den Vorteil, dass interne statische Variablen deklariert werden können. Auch die Hilfsvariablen für die Automatenzustände u.ä. müssen dann nicht adressiert werden. Ein Nachteil ist die schwierigere Verwaltung auf dem Zielsystem. Von manchen Zielsystemen wird die Generierung von Programmen erwartet.

- Rückgabetyt

Kann nur bei Angabe einer Funktion in Blockart verwendet werden und legt den Rückgabetyt

(z.B. VOID, BOOL) der Funktion fest.

- Zielsprache

Es kann zwischen verschiedenen Zielsprachen (ST, Step7-AWL, PCWorX-ST, Concept-ST, Unity-ST oder C) gewählt werden.

- Transfertyp

Mit diesem Transfertyp können Sie festlegen, mit welchem Hilfsprogramm der Quell-Code übermittelt werden soll.

- Netzwerke einfügen

Die Umsetzung der Optionen zu Netzwerken hängt davon ab, ob das Zielsystem Netzwerke kennt und TeCOM diese umsetzen kann.

- Kommentare einfügen

Wird eine Art von Know-How-Schutz benötigt oder will man einfach die Dateien verkleinern, kann man die Kommentare im generierten Code abschalten.

- Uses-Dateien

Dient zum Einbinden von Datentypen und Funktionen, die in einer anderen ST-Datei deklariert wurden. Die Dateien können entweder vom Projektverzeichnis (Ordner mit der Projekt-Datenbank) oder im aktuellen Ordner der Prap- Datei gesucht werden. Dabei muss der Dateiname mit < > oder respektive mit " " eingeschlossen werden. Wenn sie nichts angegeben haben, dann wird im aktuellen Prap- Verzeichnis gesucht. Bei der Verwendung mehrerer Dateinamen, müssen diese durch Komma voneinander getrennt werden.

- Include - Dateien

Include-Dateien sind nötig, wenn eigene Funktionen oder Datenstrukturen verwendet werden sollen, die in einer anderen ST-Datei deklariert wurden. Dabei wird im Gegensatz zu Uses-Dateien der ST-Code in den bestehenden Quellcode kopiert. Die Angabe der Dateien erfolgt nach dem gleichen Prinzip wie bei Punkt /op0"Uses-Dateien".

5.4.3. Variablennamen

Je nach Zielsystem und Benutzerwünschen können die automatisch generierten Variablen verschiedene Namen tragen. Die Generierung der Variablennamen soll jetzt von Einstellungen abhängig gemacht werden. Dafür kann für jede Variablenart getrennt ein Formatstring gesetzt werden. Als Parameter im Formatstring muss (außer bei _INIT) mindestens die Nummer X für den Platzhalter @num@ vorhanden sein. Weiterhin sind weitere Parameter möglich, die mit Werten von Eigenschaften gefüllt werden. Der Formatstring muss außerdem weitere Zeichen enthalten, die das eindeutige Auffinden der Variablen in der Variablen-tabelle auch bei veränderten Parametern ermöglichen.

Die Formatstrings sind hierarchische Eigenschaften mit einem Standardwert der dem bisherigem Format entspricht. Normalerweise sind diese Strings in den Resouceneigenschaften gespeichert und werden dort beim Anlegen einer neuen Resource in Abhängigkeit von deren Typ angelegt. Da die Formatstrings bei jeder Änderung des Praps benötigt werden, ist ein Cache nötig. Es genügt, wenn die

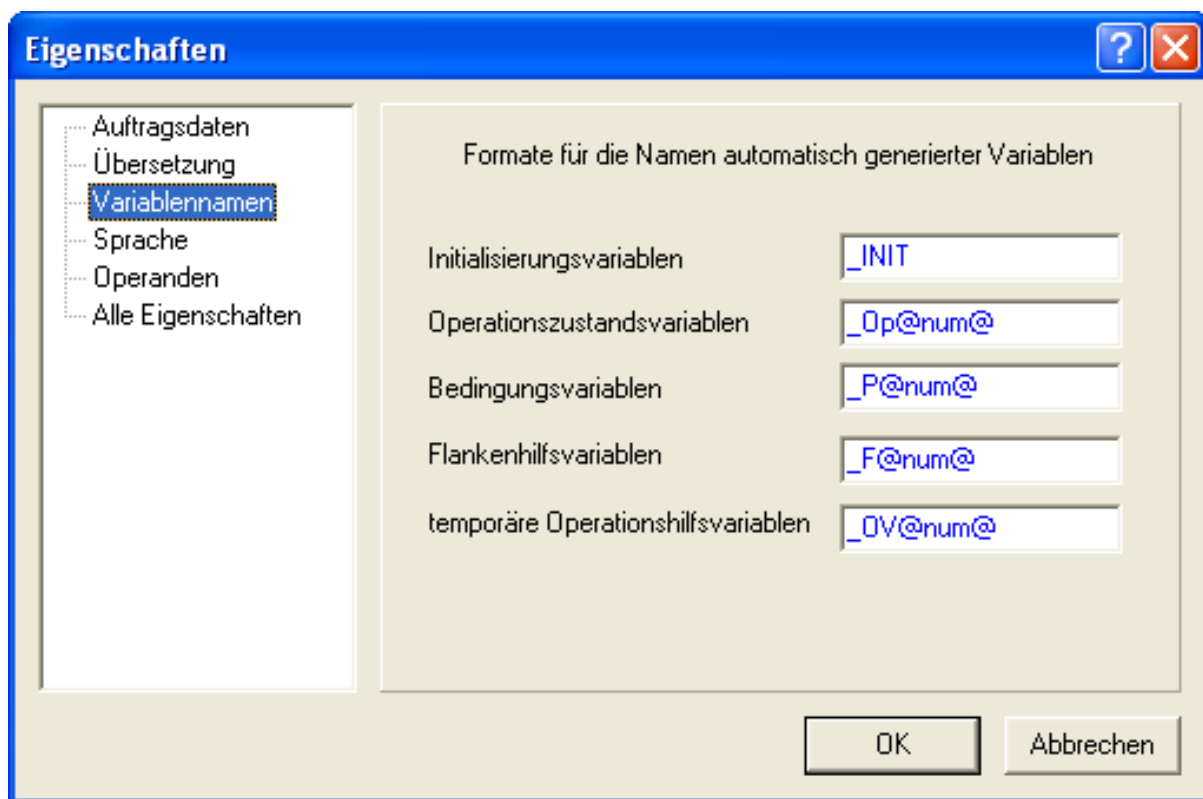
Strings beim Laden eines Prap aus den Eigenschaften gelesen werden und dann unverändert bleiben. Alle nicht lokalen Eigenschaften werden dabei bereits ersetzt, da sonst ständig ini-Dateien gelesen werden müssten.

Die Parameter haben folgende Platzhalter:

@num@ : Nummer der Variablen

@property@ : Wird ersetzt durch den Wert der Eigenschaft 'property'

Abbildung 5.3. Eigenschaften Variablennamen



Im Register **Variablennamen** können folgende Einstellungen vorgenommen werden:

- Initialisierungsvariablen

Die Variable ist normalerweise für mehrere Teilprozesse gültig. Es kann aber pro Resource mehrere Variablen geben um Teilbereiche getrennt initialisieren zu können. Die Variable ist nicht lokal. Der Name dieser Eigenschaft ist 'VARFORMAT_INIT' und kann auch unter dem Register 'Alle Eigenschaften' geändert werden.

- Operationszustandsvariablen

Lokale Variable, die den Zustand des PRAP speichert. Für den Platzhalter '@num@' steht die

Nummer der Operation. Der Name dieser Eigenschaft ist 'VARFORMAT_OP' und kann auch unter dem Register 'Alle Eigenschaften' geändert werden.

- Bedingungsvariablen

Lokale Variable, die das Ergebnis der Berechnung einer Bedingung speichert. Die Speicherung ist nur bei Zeitglied abgelaufen bzw. für die Online-Ansicht notwendig. Der Platzhalter '@num@' wird automatisch durchnummeriert. Der Name dieser Eigenschaft ist 'VARFORMAT_P'.

- Flankenhilfsvariablen

Lokale Variable zur Speicherung des Zustandes der Eingangsvariable zur Flankenbildung. Der Platzhalter '@num@' wird automatisch durchnummeriert. Der Name dieser Eigenschaft ist 'VARFORMAT_F'

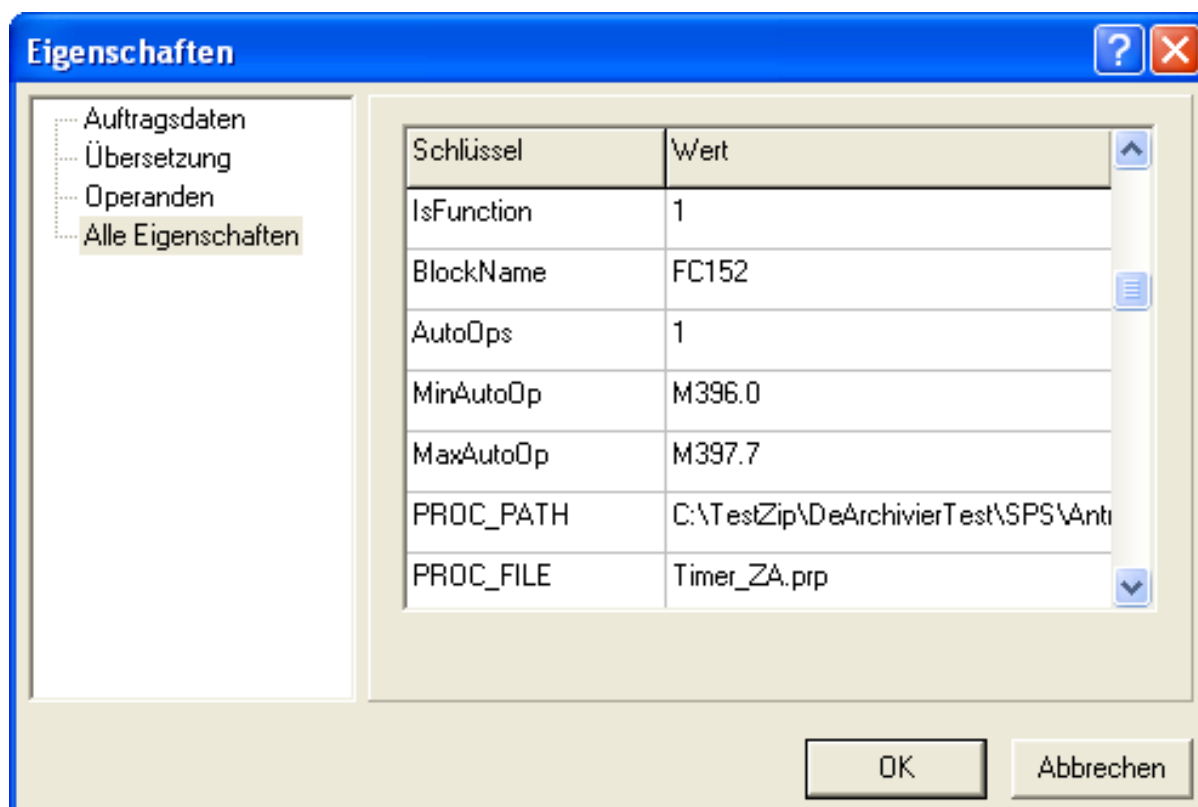
- temporäre Operationshilfsvariablen

Temporäre Variable um den Zustand einer Operationsvariable an die nachfolgende Aktion zu übergeben. Eine Speicherung ist nicht erforderlich. Die Variable ist nicht in der Variablen-tabelle enthalten und wird nur bei der Codegenerierung eingesetzt. Der Platzhalter '@num@' ist die Nummer der Operationsvariable. Der Name dieser Eigenschaft ist 'VARFORMAT_OV'

5.4.4. Alle Eigenschaften

Bei der Auswahl von Alle Eigenschaften werden alle Eigenschaften aufgelistet und können bei Bedarf geändert, hinzugefügt oder gelöscht werden. Hinzufügen oder Löschen von Eigenschaften kann über das Kontextmenü der Tabelle ausgewählt werden.

Abbildung 5.4. Eigenschaften - Alle Eigenschaften

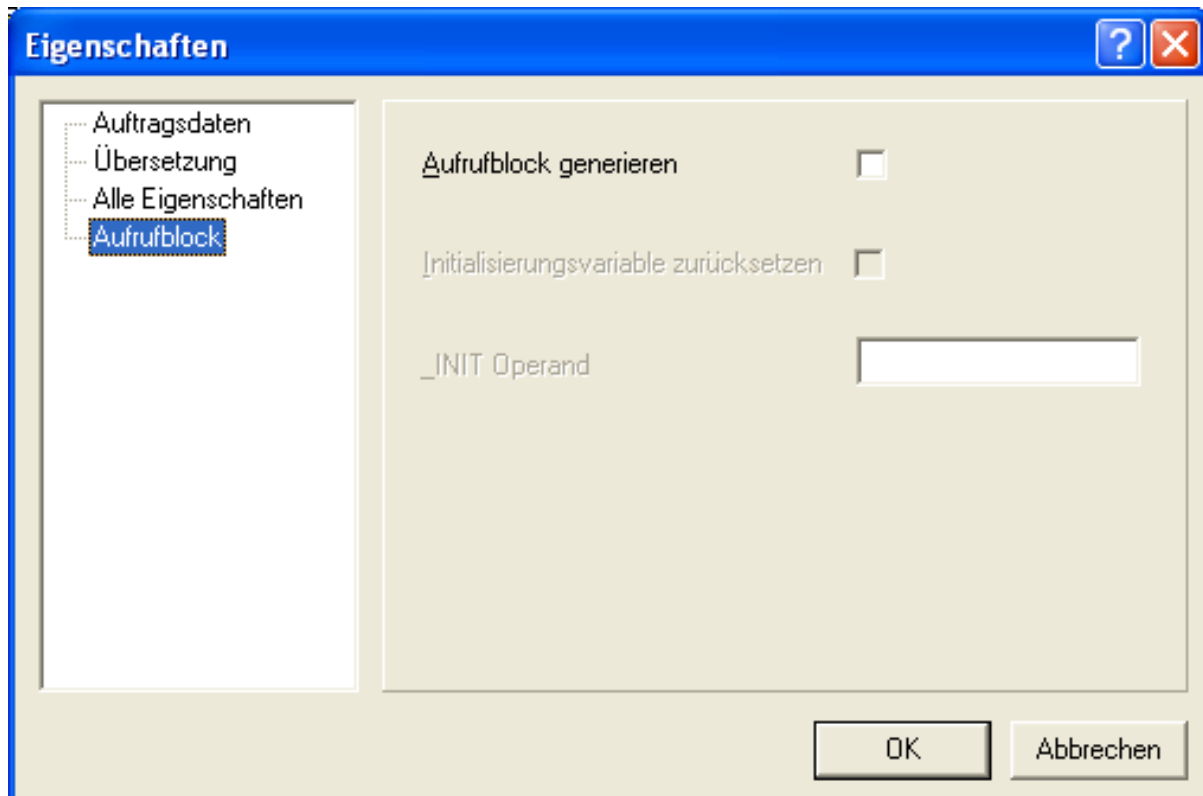


5.4.5. Aufrufblock

Der Eigenschaftsdialog eines Prozessordners wird über das Kontextmenü **Eigenschaften** des jeweiligen Eintrages im Projektverzeichnis aufgerufen werden.

Der Aufrufblock kann nur über einen Ordner des Projektverzeichnisses eingestellt werden.

Abbildung 5.5. Eigenschaften Aufrufblock



Im Register **Aufrufblock** können folgende Einstellungen vorgenommen werden:

- Aufrufblock generieren

Beim Übersetzen wird eine Funktion generiert, die alle untergeordneten Teilprozesse aufruft.

- Initialisierungsvariable zurücksetzen

Dient zum Rücksetzen der Initialisierungsvariable. Nach Abarbeitung der Teilprozesse bzw. nach Abschluss der Initialisierung besitzt die Initialisierungsvariable den Wert TRUE. Das NULL-Setzen der Variablen beim Start der SPS muss je nach Zielsystem im Anlauf der Steuerung manuell programmiert werden.

- _INIT Operand

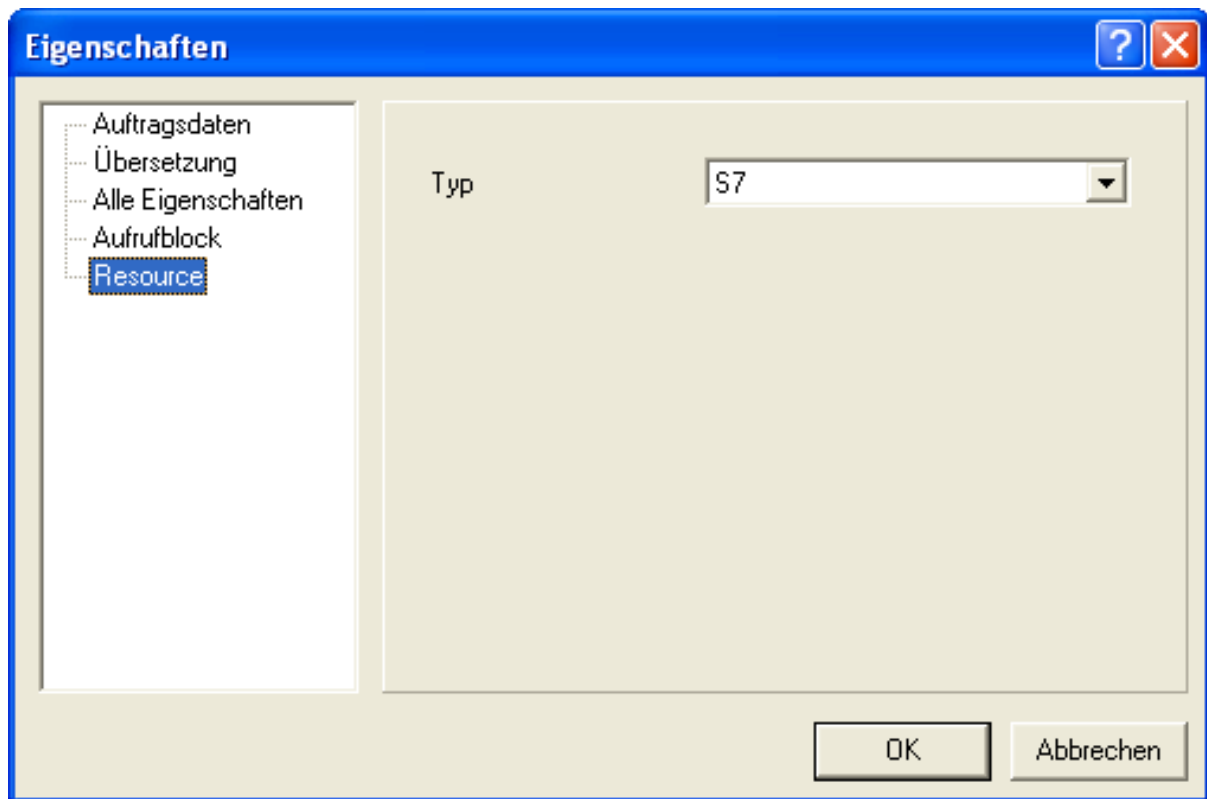
Der Operand der _INIT-Variablen wird mit TeVar beim lesen, schreiben oder prüfen entsprechend behandelt.

Der generierte ST-Code für den Prozessordner liegt in einer Datei im Prozessordner, die den gleichen Namen wie der Prozessordner hat.

5.4.6. Resource

Der Eigenschaftsdialog eines Prozessordners kann über das Kontextmenü **Eigenschaften** des jeweiligen Eintrages im Projektverzeichnis aufgerufen werden.

Abbildung 5.6. Eigenschaften - Resource



Das Register Resource ist nur über den Resourceeintrag des Projektverzeichnisses auswählbar. In diesem Register kann folgende Einstellung vorgenommen werden:

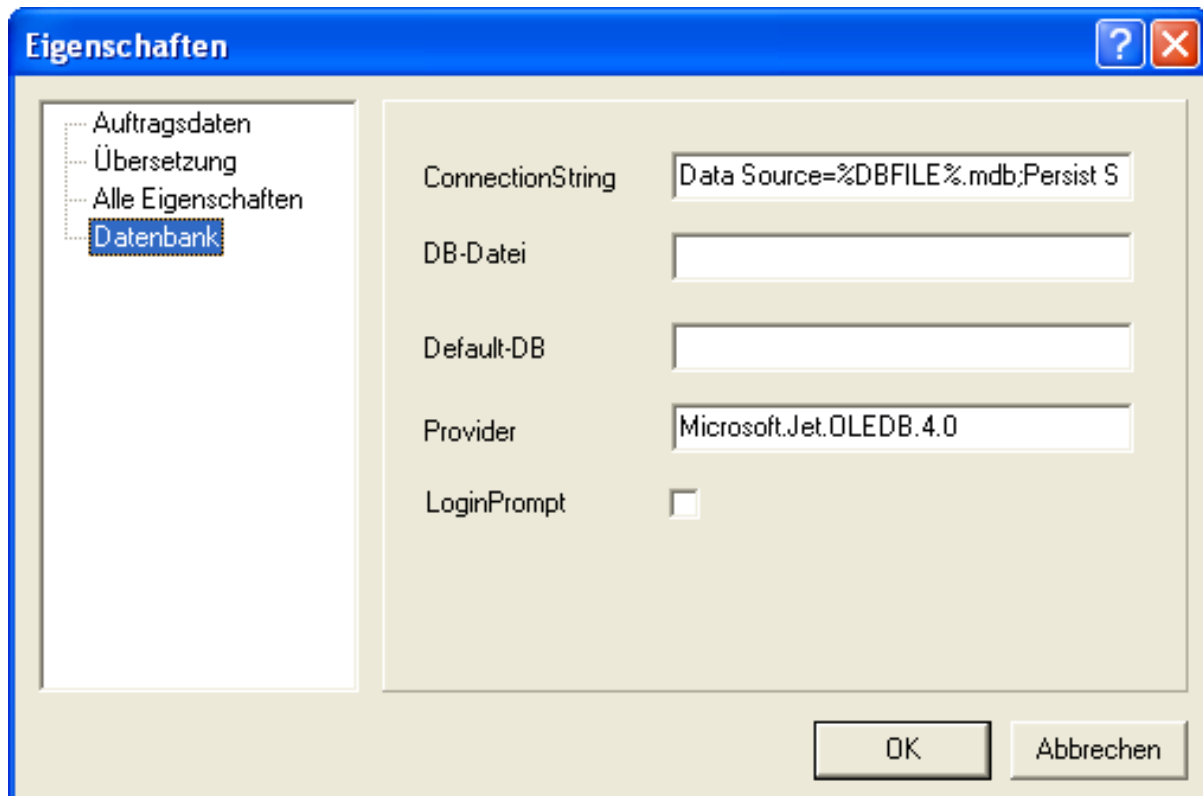
- Typ

Dient zur Festlegung des Ressourcetypen. Dabei können S7 oder IEC verwendet werden.

5.4.7. Datenbank

Der Eigenschaftsdialog eines Prozessordners kann über das Kontextmenü **Eigenschaften** des jeweiligen Eintrages im Projektverzeichnis aufgerufen werden.

Abbildung 5.7. Eigenschaften - Datenbank



Das Register Datenbank ist nur über den Datenbankeintrag des Projektverzeichnisses auswählbar. In diesem Register können folgende Einstellungen vorgenommen werden:

- ConnectionString

Die Eigenschaft ConnectionString setzt sich zusammen aus der Datenquelle (Data Source =%DBFILE%.mdb;) und der Angabe des Parameters Persist Security Info. Die Parameterangabe %DBFILE% wird durch Angabe einer DB-Datei ersetzt und darf somit nicht verändert werden. Damit wird ausgeschlossen das keine anderen Datenbanken verwendet werden dürfen. Stellen Sie sicher, dass für den Parameter "Persist Security Info" nicht true oder yes angegeben wurde, da andernfalls nach Öffnen der Verbindung durch diese vertrauliche Informationen (einschließlich Benutzername und Kennwort) abgerufen werden können.

- DB-Datei

Mit diesem Parameter können Sie eine andere Datenbank angeben. Wenn keine Datenbank eingestellt wurde, wird immer die Projektdatenbank verwendet.

- Default-DB

Name der Standarddatenbank.

- Provider

Ein Datenbank Provider stellt die Verbindung zwischen der OLE DB Ebene, die unter ADO liegt, und der Datenbank her. Zur Verwendung von Access 2000 müssen Sie den "Microsoft.Jet.OLEDB.4.0" Provider eingeben.

- LoginPrompt

Durch Setzen der Eigenschaft LoginPromt kann die Datenbank durch Eingabe eines Benutzernamen mit Kennwort geschützt werden.

Kapitel 6. Optionen

Zum Einstellen der Optionen müssen Sie auf Menüpunkt **Extras > Optionen** klicken. In diesem Dialog befinden sich 2 Registereinträge (**Drucken** und **Allgemein**), die durch Klicken auf den jeweiligen Eintrag angezeigt werden. Änderungen werden erst beim Verlassen des Dialogs durch die **OK** Schaltfläche übernommen. Im Optionsdialog sind für jedes Register Standardwerte definiert, die bei Betätigung der **Standard** Schaltfläche gesetzt werden. Dabei werden immer nur die Werte von dem aktuell angezeigten Register ersetzt.

6.1. Drucken

Der Dialog "Optionen" über Menü **Extras > Optionen...** verfügt bei Druckauswahl von **Drucken** folgende Einstellmöglichkeiten:

- Prozesse

Hier können Sie festlegen, ob Sie die aktivierten bzw. die deaktivierten Prozesse drucken wollen.

- TeCAD

In diesem Feld können Sie einstellen, ob Sie alles, nichts, oder nur ein Teil von der prp-Datei ausdrucken wollen. Wenn Sie nichts Drucken wollen, müssen Sie nur den Haken aus dem Kontrollkästchen "prp - Dateien" entfernen.

- TeList

Wenn Sie xls-Dateien drucken wollen, müssen das Kontrollkästchen "xls - Dateien" durch Klicken setzen.

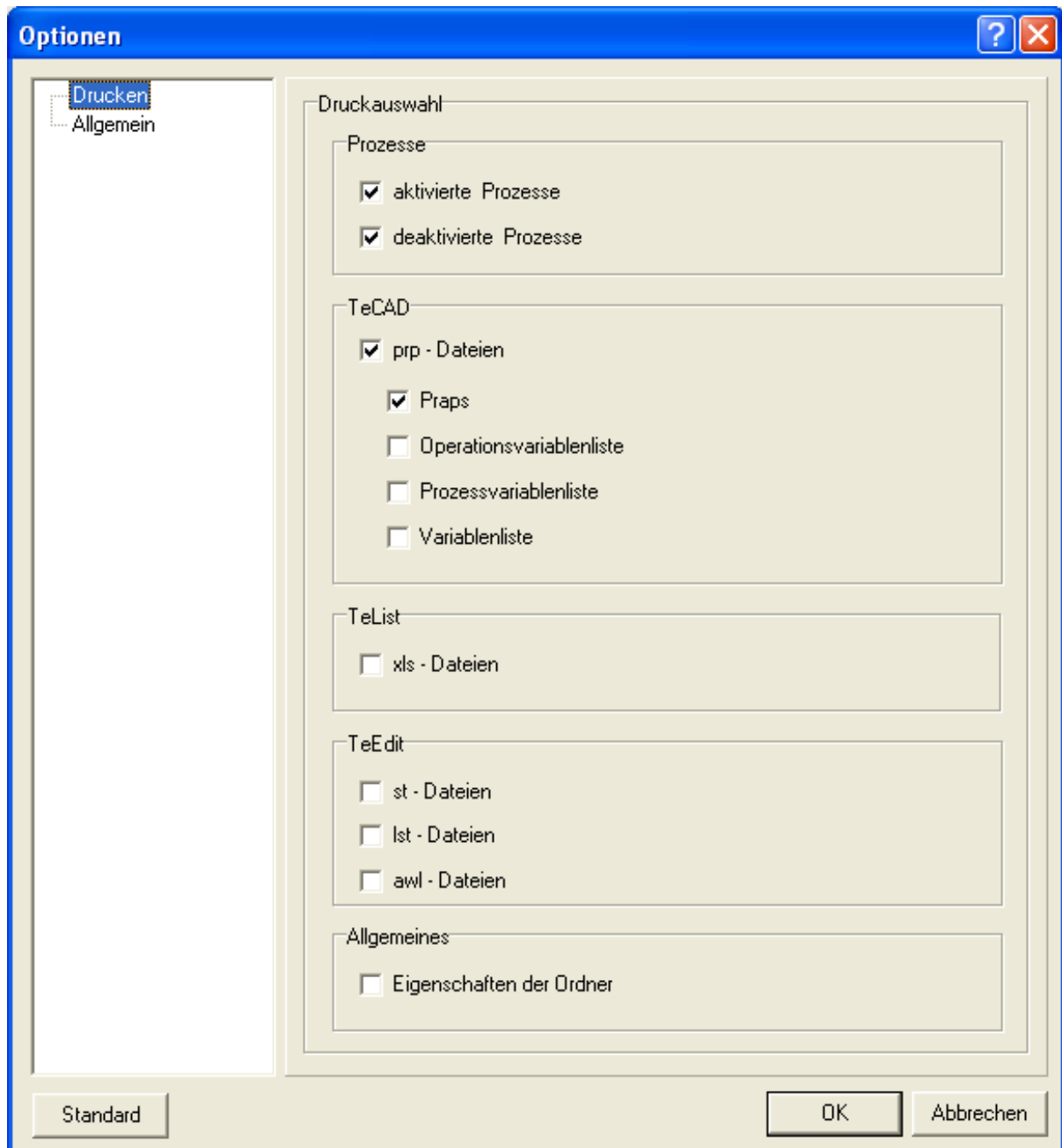
- TeEdit

Beim TeEdit können Sie festlegen, ob st-, lst- oder awl-Dateien gedruckt werden sollen.

- Allgemeines

Zum Drucken der Ordneigenschaften müssen Sie den Haken im Kontrollkästchen setzen.

Abbildung 6.1. Optionen - Drucken



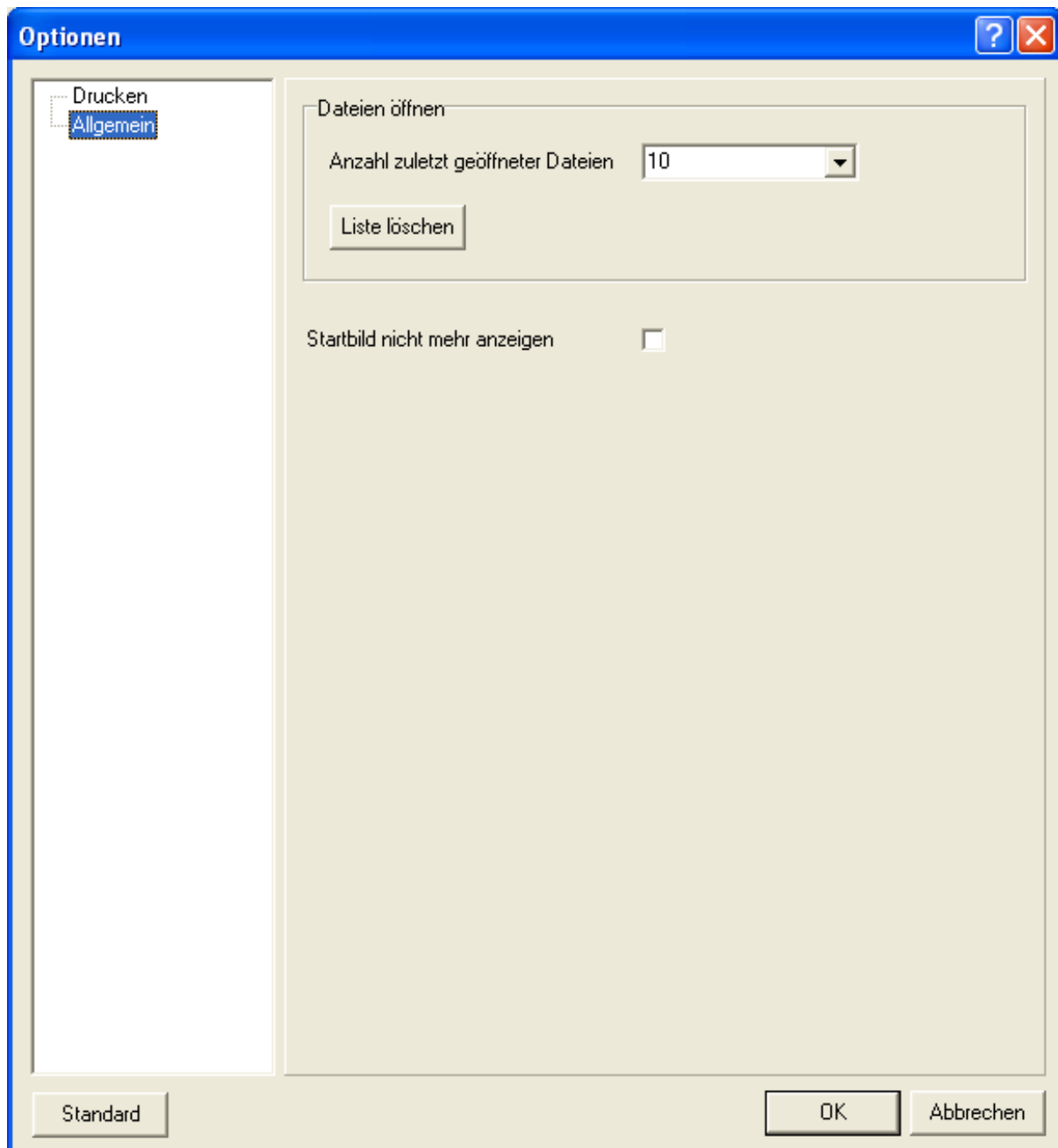
6.2. Allgemein

Der Dialog "Optionen" über Menü **Extras > Optionen** hat unter **Allgemein** folgende Einstellungsmöglichkeiten:

Im Eingabefeld "Anzahl zuletzt geöffneter Dateien" kann die Anzahl der zuletzt geöffneten Dateien eingestellt werden. Mit der darunter liegenden Schaltfläche "Liste löschen" können Sie vorhandene Dateinamen aus der Liste wieder löschen.

Mit dem Kontrollkästchen "Startbild nicht mehr anzeigen" können Sie festlegen, ob das Startbild von Studio angezeigt werden soll oder nicht.


Abbildung 6.2. Optionen - Allgemein




Kapitel 7. Drucken

Es sind folgende Möglichkeiten zum Drucken gegeben:

- Indexverzeichnis drucken

Zum Drucken des Inhaltes klicken Sie im Menü auf **Projekt >Index-Drucken...** Die zugehörige Seitenansicht kann über die Standardsymbolleiste  aufgerufen werden.

- Projektdateien drucken

Bevor Sie mit dem Drucken beginnen, müssen Sie zuerst einen Eintrag im Projektverzeichnis auswählen. Das heißt, das nur die Dateien gedruckt werden, die unterhalb des markierten Eintrag sind. Der Ausdruck kann dann über die Standardsymbolleiste  oder über dem Menü **Projekt>Drucken...** gestartet werden.

Druckoptionen:

Im Optionsdialog unter Register **Drucken** (siehe [Optionen](#)) können Sie festlegen, welche Dateien zum Ausdruck bzw. zum Indexverzeichnis gehören sollen.

Kapitel 8. System Informationen

8.1. Technische Voraussetzungen

- Betriebssysteme: Windows 2000, Windows XP, Windows 2003

für TeList ist ein installiertes Microsoft-Excel ab Version 97 Voraussetzung, Microsoft-Word für die Dokumentation

(Prinzipiell ist die TeRANiS-Gruppe auf Systemen ab Windows 98 oder Windows NT lauffähig, ggf. müssen Treiber für die Datenbank (ADO-Jet) nachinstalliert werden. Es wird jedoch keine Garantie für die uneingeschränkte Benutzbarkeit übernommen.)
- PC, auf denen die angegebenen Betriebssysteme arbeiten, Bildschirm/Grafikkarte ab 800x600 Punkte
- Für die Online-Darstellung ist eventuell weitere Hard- und Software notwendig, die abhängig von der Zielplattform ist

Weitere Empfehlungen:

- möglichst hohe Bildschirmauflösung
- Microsoft -Excel und -Access können zur Erweiterung der Funktionalität von TeVar verwendet werden.
- Insbesondere für grössere Projekte sollte genügend Arbeitsspeicher vorhanden sein. (>128MB)

8.2. Installation und Deinstallation

Sofern Ihr System CDs automatisch abspielt, brauchen Sie nur die TeRANiS CD einzulegen.

Die Installation erfolgt durch Starten der Teranis.msi-Datei, die sich auf der TeRANiS-CD befindet, oder die Sie auf anderem Weg erhalten haben.

Um Platz auf der Festplatte zu sparen, können Sie bei der benutzerdefinierten Installation die Module auswählen die Sie benutzen möchten bzw. für die Sie eine Lizenz erworben haben. Funktionen für die keine Lizenz vorhanden ist, laufen im DEMO-Modus.

Die Deinstallation kann über den entsprechenden Eintrag im Startmenü oder in der Systemsteuerung unter der Rubrik Software erfolgen.

8.3. Hardwaremäßiger Softwareschutz

Das Produkt TeRANiS wird zusammen mit einem HASP-HL USB-Dongle ausgeliefert. Auf diesem Dongle sind die Funktionen entsprechend Ihrer erworbenen Lizenz freigeschaltet.

Besitzen Sie keinen Dongle oder ist eine Funktion nicht freigeschaltet, kann die Funktion nur eingeschränkt im Demomodus verwendet werden. Nicht freigeschaltete Funktionen dürfen nur zum Testen des Programmes und nicht für produktive Projekte verwendet werden.

Achtung ! Der Dongle muss die gesamte Zeit, während die Anwendung gestartet ist, eingesteckt bleiben, da er zu verschiedenen Zeitpunkten abgefragt wird.

8.3.1. Umfang der Lizenzen

Ihre erworbene Lizenz bezieht sich auf freigeschaltete Funktionen, die maximale TeRANiS-Version und die maximale Betriebssystem-Version auf der TeRANiS eingesetzt werden kann. Sie können mit dieser Lizenz also auch jede frühere TeRANiS-Version verwenden. Auch die Verwendung auf einer früheren Betriebssystem-Version ist möglich. Beachten Sie jedoch die Einsatzvoraussetzungen für TeRANiS.

8.3.2. Treiber

Die Treiber für den HASP-HL-Dongle werden bei der Installation von TeRANiS mit installiert, sofern Sie die Option nicht abgewählt haben. Die Treiber sind von Microsoft zertifiziert und sind Bestandteil von Windows-Update. Damit können die Treiber auch direkt aus dem Internet installiert werden.

8.3.3. Hilfsprogramme

Auf der Installations-CD finden Sie verschiedene Diagnose-Tools für den Donglebetrieb. Diese können auch von der Website des Dongle-Herstellers heruntergeladen werden: <http://www.alladin.de>

8.3.4. Updates

Um weitere Funktionen freizuschalten, eine höhere TeRANiS Version zu verwenden oder es auf einem neuen Betriebssystem einzusetzen müssen Sie ein Update erwerben. Dabei erhalten Sie normalerweise keinen neuen Dongle, sondern eine E-Mail, mit der Ihr Dongle umprogrammiert wird.

8.4. Einschränkungen der Demo-Version

Wenn keine Lizenz auf einem Dongle für eine bestimmte Funktion gefunden werden kann, schaltet das betreffende Modul in den Demomodus. Dieser Zustand wird beim Starten des Moduls in einer Meldung angezeigt. Um den Demomodus zu beenden muss das Modul neu gestartet werden. Ein nachträgliches Einstecken des Dongles hat keine Wirkung.

Im Demomodus haben die einzelnen Module unterschiedliche Einschränkungen. Alle Funktionen der Module sind zugänglich und können somit getestet werden.

Allgemeines

- Beim Starten bestimmter Funktionen erscheint eine Dialogbox, die auf den Demomodus hin-

weist. Diese muss von Hand quittiert werden.

- Beim Drucken erscheint ein Text der auf die Demoversion hinweist auf dem Ausdruck

TeCAD

- Die Variablenanzahl ist auf 30 beschränkt. Da Variablen teilweise automatisch angelegt werden ist damit auch der Planungsmodus begrenzt.
- In den übersetzten ST-Code ist ein Text eingefügt, der vor der Weiterverarbeitung manuell entfernt werden muss

TeList

- Die Anzahl von übersetzbaren Instanzen ist auf 2 beschränkt

TeVar

- Die Variablenanzahl ist auf 100 beschränkt. (Dies gilt nur für das Einfügen und Ändern von Variablen)
- Es wird nur ein frei deklariertes Datentyp übernommen.

TeCOM

- Die maximale Anzahl generierter Codezeilen ist auf 500 beschränkt

TeRANiS Studio

- Die Anzahl durchführbarer Transaktionen ist auf 10 beschränkt. Danach muss das Programm neu gestartet werden.

8.5. Bezugsquellen

TeRANiS wird von der ViDEC-GmbH vertrieben. Nähere Informationen erhalten Sie dort.









ViDEC Data Engineering GmbH Osterdeich 108 D-28205 Bremen Telefon: 0421 - 33 950 - 0 Telefax: 0421 - 33 795 - 61 E-Mail: Info@videc.de

<http://www.videc.de>

Kapitel 9. Referenz

9.1. Menü und Schaltflächen

Tabelle 9.1. Menü und Schaltflächen

Menü	Tastenkürzel	Schaltfläche	Beschreibung
Projekt > Neu...			Erstellen eines neuen Projektes
Projekt > Öffnen...			Öffnen eines Projektes
Projekt > Letzte Dateien öffnen			Öffnen eines zuvor geöffneten Projektes
Projekt > Schliessen			Schliessen des aktiven Projektes
Projekt > Tages-Archivierung			
Projekt > Archivieren...			
Projekt > Dearchivieren...			
Projekt > Index-Seitenansicht			
Projekt > Index-Drucken...			
Projekt > Drucken ...			Drucken des aktiven Projektes
Projekt > Beenden			Beenden von TeEdit
Ansicht > Symbolleiste			
Extras > Optionen...			
Module > Studio	Umsch+F1		Öffnen der Anwendung Studio
Module > TeVar	Umsch+F2		Öffnen der Anwendung TeVar
Module > TeCAD	Umsch+F3		Öffnen der Anwendung TeCAD
Module > TeList	Umsch+F4		Öffnen der Anwendung TeList
Module > TeEdit	Umsch+F5		Öffnen der Anwendung TeEdit
Hilfe > Inhalt			Aufrufen der Studio-Hilfe
Hilfe > Info...			Aufrufen von Info

Stichwortverzeichnis

B

Bezugsquellen, 27

E

Eigenschaftsdialog, 10 , 12 , 16 , 17 , 18

Entwurfsprozess, 3

F

Funktion, 9

I

Installation und Deinstallation, 25

M

Menü, 28

Modul, 7

P

Projektverwaltung, 9

Prozessordner, 10

Prozesszerlegung, 3 , 3 , 4 , 5

Prozesszerlegungsgraph, 5

R

Referenz, 28

S

Schaltfläche, 28

Softwareschutz, 25

Studio, 1 , 1

System Information, 25

T

TeCAD, 7

Technische Voraussetzungen, 25

Technologische Vorhaben, 2

TeEdit, 7

Teilprozess, 4

TeList, 7

U

Uebersetzung, 10

V

Variable, 9