

tse:nit | Schnittstelle TXT-Import

gültig ab Version 2.73





Wolters Kluwer | Software+Services

Maximizing Value for Customers



Inhalt

1	Einleitung	3
1.1	Allgemeine Programmbeschreibung	۲۲
1.2	Installation	+4 ۸
1.3	Start über die tse nit Programmoberfläche (interner Programmaufruf)	۲۲ 4
1 3 2	Programmaufruf über das Windows Startmenü (externer Programmaufruf)	4
1.4	Unterstützte tse:nit Datenbereiche. Schnittstellenbeschreibungen	
2		
2	Programmbeschreibung	6
Z.1	Grundlegende Bildschirmgestaltung	0
2.1.1 2 1 2	Moniiloisto	0 6
2.1.2	Symbolschaltflächen	0 ع
2.1.3	Datengrid	۵ 8
2.1.4	Objektauswahlfeld	99
2.1.6	Objektexplorer	
2.1.7	Feldanzeige	12
2.1.8	Formeleingabefeld	13
2.1.9	Aktionseingabefeld	13
2	Europhiansweise und Festures	1.4
3 3 1	Allgemeine Vorgebensweise bei externem Programmaufruf	14 1 <i>1</i>
3.1	Allgemeine Vorgehensweise bei internem Programmaufruf	14
3.2	Arbeiten mit Varjahlen	18
3.4	Arbeiten mit Aktionen	
3.5	Modifizierung der Obiektstruktur	22
3.6	Das Datei-Verbindungsobjekt	26
3.7	Das Zusammenführen mehrerer Quelldateien	30
3.8	Eingaben während der Scriptausführung, die Funktion UserInput()	38
3.8.1	Verwendung der Funktion USERINPUT() zur Eingabe von Property-Werten	38
3.8.2	Verwendung der Funktion USERINPUT() zur Eingabe von Variablen - Werten	41
3.9	Erstellen einer Protokolldatei, die Funktion WRITETOLOG()	42
4	Anhang	43
4.1	Der Textimportassistent	43
4.1.1	Einlesen von Daten im Format MIT TRENNZEICHEN	45
4.1.2	Einlesen von Daten im Format Feste Breite	46
4.1.3	Einlesen von Daten im Excel - Format	47
4.2	Der Formel- / Aktionseditor	48
4.3	Der Funktionseditor	50
4.4	Debug-Modus	52



1 Einleitung

1.1 Allgemeine Programmbeschreibung

Bei dem Programm, TXTImport, handelt es sich um eine in tse:nit implementierte, frei programmierbare Schnittstelle, die es ermöglicht, Daten, die aus Fremdsystemen im ASCII/ANSI- bzw. Excel-Format1 bereitgestellt werden, in das tse:nit Austauschformat (Metafile - Format) zu konvertieren. Die konvertierte Datei erhält die Dateierweiterung "mta". Die Schnittstellendefinition für eine bestimmte Datenstruktur wird in einem Script einmalig durch den Anwender festgelegt und gespeichert und steht damit für weitere Importe zur Verfügung. Die Scriptdateien erhalten die Dateierweiterung "opt".

Es können Textdateien mit fester Breite der Spalten (sogenannten Feldern), Textdateien mit Trennzeichen, sogenannte CSV - Dateien oder Excel-Dateien in das tse:nit Austauschformat konvertiert werden.

Es kann mit Variablen, Funktionen und Prozeduren gearbeitet werden, wobei bei Funktionen ein eingeschränkter Sprachumfang von VBScript verwendet werden kann. Eine objektunterstützende Programmierung von Funktionen und Prozeduren ist nicht möglich.

Der TXTImport ermöglicht den Import von Quelldaten, die, entweder in einer einzelnen Datei vorliegen, oder aus mehr als einer Quelldatei zusammengeführt werden müssen. D.h. mit Hilfe des TXTImportes ist es möglich, mehrere Quelldateien zu einem Datenbestand zusammenzuführen. Die Zusammenführung der Quelldaten kann bedingungsabhängig durchgeführt werden.

Angenommen, die Mandantendaten liegen in zwei Dateien, Mandanten.txt und Mandanten Adressen.txt mit folgender Datenstruktur vor:

Mandanten.txt¶ ¶	Mandanten·Adressen.txt¶
Mandanten nummer¶ Kurzbezeichn ung¶ Name¶ Vorname¶ Titel¶ Geburtstag¶ Beruf¶ Religion¶ Familienstand¶ UStID¶ Registergericht¶ Registergericht¶ Registernummer¶ Betriebsnummer¶ Branche¶ UnternehmenBezeichnung¶ zuHaendenVon¶	Mandanten nummer¶ Kurzbezeichn ung¶ Strasse¶ Postfach¶ PLZ·Strasse¶ PLZ·Postfach¶ Ort¶

Zwischen diesen beiden Dateien wird über das Feld MANDANTENNUMMER eine Beziehung erstellt, so dass die Feldwerte der Datei MANDANTEN.TXT und die der Datei MANDANTEN ADRESSEN.TXT zusammengeführt werden, sofern der Wert des Feldes MANDANTENNUMMER in beiden Dateien übereinstimmt.

¹ ab tse:nit Version 2.73



Während der Konvertierung würden diese beiden Textdateien zu einer Datei Mandant.mta zusammengeführt, die dann in das tse:nit Mandatsverzeichnis importiert werden kann.



Abbildung 2

1.2 Installation

Eine gesonderte Installation des Programmes durch den Anwender ist nicht erforderlich. Die Installation erfolgt im Rahmen der tse:nit Installation bzw. des tse:nit Updates.

Zum Programmumfang gehören die Dateien

- TXTImport.exe
- srvTXTImport.dll

im tse:nit Installationsverzeichnis C:\Programme\Wago-Curadata\10It\BIN².

1.3 Programmaufruf

Der Programmaufruf kann über die tse:nit Programmoberfläche (interner Programmaufruf) oder extern über das Windows Startmenü erfolgen.

1.3.1 Start über die tse:nit Programmoberfläche (interner Programmaufruf)

- 1. tse:nit starten
- 2. Öffnen des Dokumentes (z.B. KUNDENSTAMM)
- 3. Starten des TXTImportes über den Menüpunkt Datei | Importieren...
- 4. Auswahl des Importsystems TEXT-DATEI-IMPORT

10 <mark>.</mark> Kundenstamm		×
Import von Text-Date	-Import	F
<u> </u>	Abbrechen	Optionen

Abbildung 3

1.3.2 Programmaufruf über das Windows Startmenü (externer Programmaufruf)

👼 10it	🕨 🔚 10it Tools 🔹 🔌 Sonstige Adressen C-Line
💼 Autostart	🕨 📻 Produktinfo 🔹 💁 TXTImport
👼 Adobe Acrobat 4.0	🕨 💁 Start 🥒 WACService
🚖 AOL	MACService

² bei Standard Einzelplatzinstallation



1.4 Unterstützte tse:nit Datenbereiche, Schnittstellenbeschreibungen

Für die folgenden tse:nit Datenbereiche kann eine Konvertierung von ASCII/ANSI- bzw. Excel-Dateien in das tse:nit Austauschformat vorgenommen werden:

Datenbereich:

tse:nit Dokumentpfad ³:

Kanzleiakte:

- Mandanten
 DAUERAKTE\ALLGEMEINE STAMMDATEN\MANDATSVERZEICHNIS
- Banken
 DAUERAKTE\ALLGEMEINE STAMMDATEN\BANKEN
- Finanzämter DAUERAKTE\ALLGEMEINE STAMMDATEN\FINANZÄMTER
- Gemeinden DAUERAKTE\ALLGEMEINE STAMMDATEN\GEMEINDEN
- Mitarbeiter
 DAUERAKTE\ALLGEMEINE STAMMDATEN\MITARBEITERLISTE

Mandanten- und Kanzleiakte:

- Kunden DAUERAKTE\GRUNDLAGEN RECHNUNGSWESEN (EIGENES REWE)\KUNDENSTAMM
 - Lieferanten DAUERAKTE\GRUNDLAGEN RECHNUNGSWESEN (EIGENES REWE)\LIEFERANTENSTAMM
- Stammkonten JAHRESAKTE\RECHNUNGSWESEN\STAMMDATEN\KONTENSTAMM
- Einzelbuchungen Jahresakte\Rechnungswesen\Finanzbuchhaltung\Buchungsliste
 - Verkehrszahlen JAHRESAKTE\RECHNUNGSWESEN\FINANZBUCHHALTUNG\SUMMEN- UND SALDENLISTE oder
 - Offene Posten JAHRESAKTE\RECHNUNGSWESEN\OP BUCHHALTUNG\OFFENE POSTEN LISTE oder JAHRESAKTE\RECHNUNGSWESEN\FINANZBUCHHALTUNG\SUMMEN- UND SALDENLISTE oder JAHRESAKTE\RECHNUNGSWESEN\JAHRESABSCHLUSS\ HAUPTABSCHLUSSÜBERSICHT
- Wirtschaftsgüter JAHRESAKTE\RECHNUNGSWESEN\ANLAGENBUCHHALTUNG\ABSCHREIBUNGSVERZEICHNIS

Die Schnittstellenbeschreibungen zu o.a. tse:nit Importformaten (Metafile, tse:nit Austauschformat) sind zum Teil auf der tse:nit Auslieferungs-DVD im Verzeichnis

<LW>\10IT_INFO\PRODUKTIONFORMATIONEN\SCHNITTSTELLEN als PDF-Dokumente abgelegt.

³ tse:nit Dokumentpfad nach Erstinstallation, Angabe in () für Kanzleiakte



2 Programmbeschreibung

2.1 Grundlegende Bildschirmgestaltung

2.1.1 Programmoberfläche



2.1.2 Menüleiste

Datei



- Erstellen eines neuen Scriptes
- Einlesen eines vorhandenen Scriptes
- Speichern des bearbeiteten Scriptes
- Speichern unter einem anderen Namen
- Neue Textdatei einlesen⁴
- Definition einer Textdatei entfernen⁵
- erstelltes Metafile anschauen
- Anzeige der bearbeiteten Scripte
- Programm beenden

⁴ Starten des Textimportassistenten

⁵ Der logische Name der Textdatei wird aus dem Script entfernt.



BEARBEITEN

Tautotaiumhanannan	- Ändern des logischen Namens einer Textdatei
r extuater unbenennen	Andern des togischen Humens einer Textauter
😭 Textdatei Einstellungen	 - Anzeige / Ändern der Textdatei Einstellungen⁶
Andere Textdatei benutzen	- Auswahl der zu verarbeitenden Textdatei
XY ⊻ariable	- Variablen definieren
🏂 Eunktion	- Funktionen / Prozeduren bearbeiten
🖒 [tse:nit] Austauschformatdatei erzeugen 👘	- tse:nit Austauschformat (Metafile) erzeugen

Abbildung 7

Extras

<u>O</u> bjekte 🕨	- Anpassung der Objektstruktur
S Optionen	- Programmoptionen

Abbildung 8

<u>N</u>eu

🗎 <u>K</u>opieren

👌 ⊻orschau

<u>E</u>ditieren X <u>L</u>öschen

EXTRAS | OBJEKTE

- neues Objekt erstellen
 - Objekteigenschaften editieren
- Objekt löschen
- Objekt kopieren
 - Objektvorschau mit den getroffenen Feldzuordnungen

Abbildung 9

EXTRAS | OPTIONEN

Optionen	×	Auswahl des Ordners für die einzulesende Quelldatei
Ordner Textdatei C:VF [tse:nit] Austauschformatdatei C:VF	Programme\Wago-Curadata\10it\	Auswahl des Ordners für die zu generierende tse:nit Austauschformatdatei (Metafile)
Script C:\F Anzeige-Programm note	Programme\Wago-Curadata\10it\E 🔄	Auswahl des Ordners zur Speicherung des erstellten Scriptes
Einrückung im Funktionseditor 4		Auswahl des Anzeigeprogramms für die zu generierende tse:nit Austauschformatdatei (Metafile)
		Anzahl Leerzeichen für Einrückungen im Funktionseditor bei Verwendung der Tab-Taste

Abbildung 10

HILFE

😰 Erste Schritte - Anzeige der Hilfedatei

⁶ abhängig vom Projektstatus (Dateidefinition bzw. Scripterstellung)



2.1.3 Symbolschaltflächen

Mandant	- Auswahl der Textdatei (wenn mit mehr als einer Textdatei gearbeitet wird)
D	- Erstellen einer neuen Script-Datei
2	- Öffnen eines bestehenden Scriptes
	- Speichern des Scriptes
	- Öffnen der zu verarbeitenden Textdatei
×	- Entfernen einer Textdatei
	- Eigenschaften der Textdateidefinition festlegen
×	- Variablen definieren
f.	- Funktionseditor
	- Metafile erzeugen
	- Ausgabepfad und Dateibezeichnung für das Metafile festlegen

2.1.4 Datengrid

Im Datengrid wird der Inhalt der Textdatei gemäß der getroffenen Felddefinitionen während des Textdateiimportes⁷ angezeigt.

Manuariterinuminer iku	urzbezeichnung	Name	Vorname	Litel	Geburtstag	Beruf	Religion	Familienstand	1
4711 Be	eyer KG	Beyer	Anton	Dr.	01111966	Dipl. Betriebswirt	lt	verh	
4712 Da	aimler GmbH	Daimler	Christopher		06121957	Kfz-Meister	rk	led	

Abbildung 12

Über das Kontextmenü im Grid können die Werte aus dem markierten Datensatz in die Feldanzeige übernommen werden, was für den Test von Formeln besonders hilfreich ist. Weiterhin besteht über das Kontextmenü die Möglichkeit, ganz gezielt nach bestimmten Werten im Quelldatenbestand (Inhalt der Textdatei) zu suchen.

Kontextmenü im Datengrid:

	<u>T</u> estwerte aus Datei übernehmen
# \$	<u>S</u> uchen (Strg-F)
₩.,	Suchen <u>n</u> ächsten (F3)

```
Abbildung 13
```

- Testwerte in die Feldanzeige übernehmen
- nach Wert suchen
- nächsten Wert suchen

⁷ siehe 4.1 Textimportassistent



2.1.5 Objektauswahlfeld



Über das Objektauswahlfeld erfolgt die Auswahl der zu erzeugenden Metafilestruktur.

Bank 🔺
Finanzamt
Gemeinde
Mitarbeiter
Mandant
Kunde
Lieferant
Mandant_Buchung 📃 💌
Abbildung 15

Nach Auswahl des Datenbereiches (z.B. Mandant) wird die Metastruktur im Objektexplorer angezeigt und die Zuordnung der Datenfelder der Quelldatei zu den Properties des Metafiles kann, nachdem eine logische Verbindung mit der Textdatei vorgenommen wurde, erfolgen.

2.1.6 Objektexplorer

Im Objektexplorer erfolgt die Anzeige der zu erzeugenden Metafilestruktur mit den getroffenen Feldzuordnungen. Die Anzeige der Namen von Objekten und Properties (Eigenschaften der Objekte) kann wahlweise im Normaltext oder im tse:nit Austauschformat erfolgen. Die Umschaltung erfolgt über das Kontextmenü.

📮 Mandant <-> Mandant	📮 Mandant <-> Mandant
Mandantennummer <-> Mandant("Mandantennummer").AsString	plManNr <-> Mandant("Mandantennummer").AsString
Kurzname <-> Mandant("Kurzbezeichnung").AsString	psKurzname <-> Mandant("Kurzbezeichnung").AsString
Mandant seit	psSeit
Mandant bis	Objekt [Vendent]
Name <-> Mandant("Name").AsString	ODJEKT [Mandant] Mandant("Name"),AsString
Vorname <-> Mandant("Vorname").AsString	psvomame <-> Mandant("Vorname").AsString
Titel <-> Mandant("Titel").AsString	Figenschaft psName (Name) des
Geburtstag <-> Mandant("Geburtstag").AsDate	Objektes [Mandant]
Beruf	Objektes [Manualit]
Religionskenner <-> Mandant("Religion").AsString	piReligionid <-> Mandant("Religion").AsString
Familienstandskenner <-> Mandant("Familienstand").AsString	
USt-IdNr <-> Mandant("UStID").AsString	Subobjekt [poHausAnschrift] des tiD"].AsString
Handelsregistergericht <-> Mandant("Registergericht").AsString	Objektes [Mandant] Indant["Hegistergericht"].AsString
Handelsregisternummer <-> Mandant("Registernummer").AsString	ht("Registernummer").AsString
Betriebsnummer <-> Manefant("Betriebsnummer").AsString	b Hard String
Rechtsformkenner <- Mandant("Rechtsform").AsString	Eigenschaft psStrasse des
Branchenschlüsser <-> Mandant("Branche"),AsString	Subobjektes [poHausanschrift]
Unternehmen Bezeichnung <>> Mandant("UnternehmenBezeichnur	UnternenmenBezeichnung" J.Asstring
Unternehmen z.Hd. von Mandant("zuHaendenVon").AsString</td <td>LuHaendenVon'),Asstring</td>	LuHaendenVon'),Asstring
Art des Unternehmen <-> Mandant("UnternehmenArt"),AsString	psunternenmensart <-> Mandant(UnternenmenArt).Asstring
HausAnschrift	
Strasse	psotrasse
Zusatz	ps_usatz
	<u>⊢</u> po∟anu
Nenner Dessisterer	pskurzbezeichnung
	psbezeichnung
	im ∓ ho⊾osrauscum





Kontextmenüs im Objektexplorer

Abhängig von der ausgewählten Zeile in der Objektstruktur werden über die rechte Maustaste unterschiedliche Kontextmenüs angezeigt.

Auswahl steht auf der obersten Objektebene (z.B. MANDANT):

${\bf S}_{\rm s}$	⊻erbinden mit Datei Mandant	 logische Verbindung zur Datei erzeugen
\times	Verbindung zur <u>D</u> atei lösen	- logische Verbindung zur Datei lösen
	Verbinden mit Formel (Bedingung)	- Bedingung aus dem Formeleingabefeld übernehmen
	Formel bearbeiten	- Formel (Bedingung) bearbeiten
	ins Formeleingabefeld übernehmen	- Formel (Bedingung) ins Formeleingabefeld übernehmen
	Verbindung mit Formel lösen	- Formel (Bedingung) löschen
Ø	Alle Verbindungen zum Objekt aufheben	- alle getroffenen Zuordnungen werden gelöscht
	Editieren	
\times	 Lioschen	
Þ	<u>K</u> opieren	
ß	Einfügen +	
	Property hinzufügen	 Hinzufügen eines Propertys zum Objekt
	Subobjekt hinzufügen	 Hinzufügen eines Unterobjektes zum Objekt
	Verbindungs <u>o</u> bjekt zur Datei erzeugen	
	Aktion hinzufügen	
	Objekt Nr hinzufügen	 Hinzufügen des Objektes ObjectNr
	Öffnen der Unterobjekte	- Öffnen aller Unterobiekte (Expand)
	Schließen der Unterobjekte	- Schließen aller Unterobjekte (Collapse)
	Namen im [tse:nit] Austauschformat anzeigen	- Bezeichnungen werden im tse:nit Austauschformat gezeigt
		J



Auswahl steht auf Subobjektebene (z.B. POHAUSANSCHRIFT):

 Verbinden mit Datei Verbindung zur <u>D</u>atei lösen Verbinden mit <u>F</u>ormel (Bedingung) Formel bearbeiten ins Formeleingabefeld übernehmen Verbindung <u>m</u>it Formel lösen Alle Verbindungen zum Objekt aufheben 	 logische Verbindung zu einer weiteren Datei erzeugen⁸ logische Verbindung zur Datei lösen Bedingung aus dem Formeleingabefeld übernehmen Formel (Bedingung) bearbeiten Formel (Bedingung) ins Formeleingabefeld übernehmen Formel (Bedingung) löschen
Editieren ★ Löschen È Kopieren È Einfügen Property hinzufügen	- Subobjekt bearbeiten ⁹ - Subobjekt mit allen Properties löschen - Subobjekt kopieren - Subobjekt einfügen - Hinzufügen eines Property zum Unterobjekt
<u>S</u> ubobjekt hinzufügen Verbindungs <u>o</u> bjekt zur Datei erzeugen <u>A</u> ktion hinzufügen <u>O</u> bjekt Nr hinzufügen	 Hinzufügen eines Unterobjektes zum Unterobjekt Verbindungsobjekt zu einer Datei erzeugen Aktion aus dem Eingabefeld in die Struktur einfügen Hinzufügen des Objektes ObjectNr
Öffnen der Unterobjekte Schließen der Unterobjekte ✓ Namen im [tse:nit] Austauschformat anzeigen Abbildung 19	- Öffnen aller Unterobjekte (Expand) - Schließen aller Unterobjekte (Collapse) - Bezeichnung werden im tse:nit Austauschformat gezeigt

Auswahl steht auf Property (z.B. PSORT):

Service Verbinden mit Textdateispalte Ort	- Property einem Datenfeld der Quelldatei zuordnen
Verbinden mit <u>K</u> onstante (D)	- Property einer Konstante zuweisen
Verbinden mit <u>F</u> ormel (mid(now(),1,8))	- Property durch Formel aus dem Eingabefeld errechnen
Formel bearbeiten	- Berechnungsformel bearbeiten
ins Formeleingabefeld übernehmen	- aktuell verbundene Formel ins Eingabefeld übernehmen
🔀 🛛 🗠 Verbindung lösen	- Verbindung Formel bzw. Datenfeld löschen
<u>E</u> ditieren	- Property bearbeiten
🗙 Löschen	- Property löschen
Aktion hinzufügen	- Aktion unterhalb des Property einfügen
Öffnen der Unterobjekte	- Öffnen aller Unterobjekte (Expand)
Schließen der Unterobjekte	- Schließen aller Unterobjekte (Collapse)
✓ Namen im [tse:nit] Austauschformat anzeigen	- Bezeichnung werden im tse:nit Austauschformat gezeigt
Abbildung 20	

 ⁸ bei Arbeit mit mehr als einer Textdatei
 ⁹ Bearbeiten von angezeigtem Namen, Namen im Metafile, Kommentar, Mehrfachauftreten zulassen



2.1.6.1 Das Objekt ObjectNr (Objekt Nr)

Das Objekt **ObjectNr** (Objekt Nr) wird zur Unterscheidung von mehreren Objekten / Objekteigenschaften mit gleichem Namen verwendet (Objekte oder Eigenschaften treten mehrfach auf).

Das Hinzufügen des Objektes ObjectNr hat folgende Auswirkungen:

Jedes Objekt / Objekteigenschaft erhält bei der Konvertierung in das Metafile-Format eine fortlaufende Nummer, die dieses Objekt von Objekten mit gleichem Namen unterscheidet. Die Verknüpfung (Wertzuweisung) zu dem Objekt ObjectNr erzielt man in Verbindung mit einer Zählvariablen unter Verwendung von einer Aktion, die in die Objektstruktur eingefügt wird.¹⁰

2.1.7 Feldanzeige

In der Feldanzeige werden alle Datenfelder der Quelldatei mit ihrem festgelegten Datentyp und ihren aktuellen Werten des im Datengrid ausgewählten Datensatzes angezeigt.

Spaltenbezeichnung	Datentyp	Wert	
Mandantennummer	AsString	4711	
Kurzbezeichnung	AsString	Beyer KG	
Name	AsString	Beyer	
Vorname	AsString	Anton	
Titel	AsString	Dr.	— Auswahlfeld zur Änderung
Geburtstag	AsDate	01111966	des Datentyps
Beruf	AsString	Dipl. Betriebswirt	
Religion	AsString	lt	
Familienstand	AsString	verh	
UStID	AsString		
Registergericht	AsString	Mannheim	
Registernummer	AsString	HRA 12345	
Betriebsnummer	AsString	123456789	
Rechtsform	AsString	KG	
Branche	AsString		
UnternehmenBezeichnung	AsString	Beyer KG	
zuHaendenVon	AsString	Herrn Beyer	
UnternehmenArt	AsString		

Abbildung 21

Durch Auswahl eines Feldes und Herüberziehen mit gedrückter linker Maustaste (Drag & Drop) in den Objektexplorer auf ein Property in der Objektstruktur erfolgt die Festlegung der Feldzuordnung zwischen der zuvor mit Hilfe des Textimportassistenten eingelesenen Quelldatei (Textdatei bzw. Excel-Datei). Weiterhin ist es in der Feldanzeige möglich, auch noch im Nachhinein den Datentyp eines Datenfeldes zu ändern.

Folgende Datentypen werden unterstützt:

Datentyp	Beispiel
Boolean	True, False
Currency	123.456.789,12; -123.456.789,12
Date	21.05.2002 (Werte zw. 01.01.100 u. 31.12.9999)
Float	123456789,12; -123456789,12
Integer	123456789; -123456789
String	"© 2011 ADDISON Software und Service GmbH"

¹⁰ siehe Kapitel 3.3 **Beispiel 1**



2.1.8 Formeleingabefeld

Formel	Variables("DSNr").AsInteger = Variables("DSNr").AsInteger + 1	

Abbildung 22

Das Formeleingabefeld dient zur Vorbereitung von Berechnungsformeln, die dann einem Property als Wert oder einem Objekt als Bedingung zugeordnet werden können. Über die Schaltfläche – wird der Formeleditor gestartet, mit dem komfortabel Berechnungsformeln erstellt werden können.

2.1.9 Aktionseingabefeld

Aktion Variables("Mandantennummer").AsString = UserInput("Angabe der er	
---	--

Abbildung 23

Das Aktionseingabefeld dient zur Vorbereitung von so genannten Aktionen (Formeln, Berechnungen, Eingaben des Anwenders) die dann in die Objektstruktur eingefügt werden und zur Laufzeit des Scriptes ausgeführt werden. So ist es zum Beispiel möglich, während der Scriptausführung Variablen zu berechnen und diese berechneten Variablen Properties zuzuweisen. Über die Schaltfläche 🗐 wird der Aktionseditor gestartet.



3 Funktionsweise und Features

3.1 Allgemeine Vorgehensweise bei externem Programmaufruf

Den Konvertierungsprozess von Quelldaten (Textdateien), die in einfacher Datenstruktur in lediglich einer Quelldatei vorliegen, kann man, bei externem Programmaufruf, folgendermaßen beschreiben:

- 1. Starten des TXTImportes (siehe 1.3.2)
- 2. Auswahl des Datenbereiches über das Objektauswahlfeld (siehe 2.1.5)
- 3. Auswahl der Quelldatei (ASCII, ANSI, Excel) über den Textimportassistenten (Menü DATEI | NEUE TEXTDATEI)¹¹



Abbildung 24

4. Festlegen der Datei- und Verarbeitungseinstellungen



Abbildung 25

5. Festlegen der logischen Datei- und Feldnamen und der Felddatentypen.



ADDITUUTING 20

¹¹ siehe 4.1 Der Textimportassistent



- 6. Einlesen der Quelldatei über <u>Fertigstellen</u>.
- 7. Über das Kontextmenü im Objektexplorer verbinden des Objektes (z.B. MANDANT) mit dem logischen Namen der eingelesenen Datei



- 8. Zuordnung der Datenfelder zu den Properties im Objektexplorer per Drag & Drop über die Feldanzeige und den Objektexplorer vornehmen
- 9. Erzeugung der tse:nit Austauschformatdatei (z.B. Mandant.mta) mittels der Schaltfläche 🕨
- 10. In tse:nit das entsprechende Dokument öffnen (z.B. MANDATSVERZEICHNIS in der Kanzleiakte) und die erzeugte Metadatei über den Menüpunkt DATEI | IMPORTIEREN... als TSE:NIT AUSTAUSCHFORMAT einlesen



Abbildung 28



3.2 Allgemeine Vorgehensweise bei internem Programmaufruf

Den Konvertierungsprozess von Quelldaten (Textdateien), die in einfacher Datenstruktur in lediglich einer Quelldatei vorliegen, kann man, bei internem Programmaufruf, folgendermaßen beschreiben:

- 1. tse:nit starten und Öffnen des entsprechenden Dokumentes, in das Daten importiert werden sollen (z.B. MANDATSVERZEICHNIS)
- 2. Starten des TXTImportes über den Menüpunkt DATEI | IMPORTIEREN... | TEXT-DATEI-IMPORT (siehe 1.3.1)



Der Datenbereich (Objekt) ist bei internem Programmaufruf voreingestellt. Eine Änderung über das Objektauswahlfeld ist nicht möglich.

3. Auswahl der Quelldatei (ASCII, ANSI, Excel) über den Textimportassistenten (Menü DATEI | NEUE TEXTDATEI)



Abbildung 29

4. Festlegen der Datei- und Verarbeitungseinstellungen



Abbildung 30

5. Festlegen der logischen Datei- und Feldnamen und der Felddatentypen.



Abbildung 31



- 6. Einlesen der Quelldatei über <u>Fertigstellen</u>.
- 7. Über das Kontextmenü im Objektexplorer verbinden des Objektes (z.B. MANDANT) mit dem logischen Namen der eingelesenen Datei



- 8. Zuordnung der Datenfelder zu den Properties im Objektexplorer per Drag & Drop über die Feldanzeige und den Objektexplorer vornehmen
- 9. Erzeugung der tse:nit Austauschformatdatei (z.B. Mandant.mta) mittels der Schaltfläche 🕨
- 10. Import der Daten über die Schaltfläche Übergehmen in der Metafileanzeige

💁 C:\Programme\Wago-Curadata\10it\MetaFile\ 📃 🗖 🗙					
Datei Bearbeiten Übernehmen					
ExportedSystem="Text-Import" LangVer="300" Vorkstaion01b="0" CodeFage="ANSI" Created="24.05.2002.12.31:41" Comment="Importient von Text-Date: C:VProgramme\Wago-Curr (Block,24.05.202.12.31:41> [Mandan1.] pManN=0 psKurzname=Beyer KG psSei=01.01.55	adata\				

Abbildung 33



3.3 Arbeiten mit Variablen

In der Applikation TXT-Import ist es möglich, mit benutzerdefinierten Variablen zu arbeiten. Durch die Verwendung von Variablen kann eine Vereinfachung komplizierter Ausdrücke, die für die Auswertung von Bedingungen bzw. für die Verbindung von Feldern der Quelldatei mit Properties von Objekten verwendet werden, erreicht werden. Die definierten Benutzervariablen sind Bestandteil des Scriptes und werden mit diesem gespeichert.

Die Definition von Variablen erfolgt entweder über den Menüpunkt BEARBEITEN | VARIABLE oder über die Symbolschaltfläche X. Es öffnet sich das folgende Formular.

X <mark>y</mark> Variable	X		
Variable Name DSNr	Datentyp Asinteger	Initial-Wert	<u></u> K
MNr	AsInteger	0	Abbrechen
Useit	AsDate	mid(now(),1,8)	<u>N</u> eu
			<u>L</u> öschen
			<u>B</u> earbeiten

Abbildung 34

In diesem Beispiel sind drei Variablen definiert:

DSNr	- Datensatznummer	mit dem Initialwert 0	
MNr	- Mandantennummer	mit dem Initialwert 0	
Dseit	- Datum seit (Mandant seit)	Ermittlung des Initialwertes über die Kombination d	er
		VBScript Funktionen mid() und now()	
		Bsp: now() = 22.05.02 14:35	
		mid(now(),1,8) = 22.05.02	
		(Teilstring beginnend bei 1 mit der Länge 8)	ļ

Wie aus dem o.a. Beispiel ersichtlich wird, kann der Initialwert (Anfangswert der Variablen) entweder als eine Konstante festgelegt oder über eine Funktion ermittelt werden, wobei es **nicht** möglich ist, Verweise auf andere Variablen oder Felder von logischen Dateien zu verwenden.

Syntaxnotation für den Einsatz einer Benutzervariablen in Verknüpfungen, benutzerdefinierten Funktionen oder Prozeduren:

VARIABLES("VARIABLENNAME").ASDATENTYP

ASDATENTYP bestimmt, welcher Datentyp der Variablen zugeordnet wird. Folgende Datentypen sind zugelassen: AsBoolean, AsCurrency, AsDate, AsFloat, AsInteger, AsString ¹²

¹² siehe auch 2.1.7, Datentypen



Beispiel 1

Variables("DSNr").AsInteger = Variables("DSNr").AsInteger + 1

Die Variable "DSNr" wird bei jedem Scriptdurchlauf um 1 erhöht. Mit Hilfe einer derartigen Variablen ist es zum Beispiel möglich, das Objekt **ObjectNr**¹³ fortlaufend hochzuzählen.

Auszug aus der Objektstruktur:

Mandant <-> Mandant (len(Mandant("Mandantennummer").AsString) <> 0) Variables("DSNr").AsInteger = Variables("DSNr").AsInteger + 1 Diskt Nr.co. Asinteger("DSNr").AsInteger)	—	Eingefügte Aktion, in der die Berechnung der Variablen erfolgt
Mandantennummer <-> Mandant("Mandantennummer").AsString	— [Die berechnete Variable wird dem Objekt OBJECTNR zugewiesen.

Abbildung 35

Beispiel 2

Bedingungsabhängige Verknüpfung eines Propertys unter Verwendung einer Variablen und eines Feldes einer logischen Datei.

Dem Property psSeit (Mandant seit) wird, abhängig vom Feldinhalt des Feldes "MandantSeit" der logischen Datei "Mandant", entweder der Feldinhalt (wenn Feldinhalt <> "") oder das aktuelle Systemdatum (wenn Feldinhalt = "") unter Verwendung der VBScript Funktion IIF() zugewiesen.

Auszug aus der Objektstruktur:



¹³ siehe auch 2.1.6, Kontextmenüs im Objektexplorer



3.4 Arbeiten mit Aktionen

Aktionen sind Anweisungen, die während des Konvertierungsprozesses in der Objektstruktur mit benutzerdefinierten Variablen ausgeführt werden. Die Erstellung von Aktionen kann entweder manuell im Aktionseingabefeld oder über den Aktionseditor erfolgen. Der Inhalt des Aktionseingabefeldes wird über das Kontextmenü im Objektexplorer in die Objektstruktur eingefügt, d. h. es wird so die Position im Konvertierungsprozess definiert (siehe Abbildung 37).

atei <u>B</u> earbeite	en <u>E</u> xtras <u>H</u> ilfe						
andant	💽 🗅 🚅 🖬 🗖 🗙 🖆	🚰 🦄 🏂 🕒 🖓 🏹	go-Curadata\10it\Met	aFile\Mandan	t.mta		
idantennummei 1	r Kurzbezeichnung Name Vorname Beyer KG Beyer Anton	Titel Geburtstag Beruf Dr. 01111966 Dipl. Betriebswirt	Religion Familiens It verh	tand UStID	Registergericht Mannheim	Registernummer HRA 12345	Betriebsnumm 123456789
dant	T						
Mandaat Objekt N	Verbinden mit Datei Mandant (Mandante Verbinden mit Datei Mandant	nonumner").AsString) <> 0)	Spaltenbezeich Mandantennun	nung mer	Dater AsStr	ntyp Wing 47	ert 11
Mandar Kurznan	Verbinden mit Eormel (Bedingung)		Name Vorname	ig	AsStr AsStr AsStr	ing Be ing An	yer ku yer iton
Mandar Mandar Name≺	ins Formeleingabefeld übernehmen Verbindung mit Formel lösen	how(J,1,8),Mandant("MandantSe	Titel Geburtstag Beruf		AsStr AsDa AsStr	ing Dr. ite 01 ing Dij	111966 pl. Betriebswirt
Vornami Titel <->	Alle Verbindungen zum Objekt aufheben		Religion Familienstand		AsStr AsStr	ing It ing ve	rh
Beruf X Religion	Löschen Kopieren		Registergericht Registernumme	ſ	AsStr AsStr	ing Ma ing HF	annheim RA 12345
Familien USt-IdN Handels	Einfügen Property binzufügen		Betriebsnumme Rechtsform Branche	ſ	AsStr AsStr AsStr	ing 12 ing KG ing	3456789 ì
Handels Betriebs	Subobjekt hinzufügen Verbindungsobiekt zur Dateilerzeugen	p	UnternehmenB zuHaendenVor	ezeichnung I	AsStr AsStr	ing Be ing He	yer KG arm Beyer
Rechtsf Branche	Aktion hinzufügen	nuna'') AsString}	UnternehmenA MandantSeit	t	AsStr AsStr	ing 01	.01.95
Unterne Art des I	Öffnen der Unterobjekte		Konstante Formel				
🕀 Hau	Namen im Iteanit1 Austauschformat anzeig	ressen("Strasse").AsString	Aktion Mar	ablee("DSNr") Aslinteger – Varial	blee("DSNr") Aelpt	ener + 1

Abbildung 37

Der Menüpunkt AKTION HINZUFÜGEN im Kontextmenü des Objektexplorers ist nur aktiv, wenn eine Aktion im Aktionseingabefeld hinterlegt ist.

Durch den Menüpunkt AKTION HINZUFÜGEN wird eine neue Zeile mit der im Aktionseingabefeld hinterlegten Berechnungsformel in die Objektstruktur eingefügt. Die Position dieser Zeile bestimmt die Stelle im Script, an der die entsprechende Anweisung ausgeführt wird.



Abbildung 38

Im o.a. Beispiel (siehe Abbildung 38) wird die benutzerdefinierte Variable **DSNr** bei jedem Scriptdurchlauf um eins erhöht. Die über die Aktion

Variables("DSNr").AsInteger = Variables("DSNr").AsInteger + 1



berechnete Variable wird in der nächsten Objekt-Zeile dem Objekt ObjectNr zugewiesen¹⁴.

Das Ergebnis der eingefügten Aktion sieht dann in der erzeugten Metadatei folgendermaßen aus:



Abbildung 39

Als Aktionen können außer Berechnungsformeln auch benutzerdefinierte Funktionen¹⁵ in die Objektstruktur eingefügt werden, deren Berechnung dann während der Konvertierungsprozesses an der so definierten Position in der Objektstruktur ausgeführt werden. So wurde die benutzerdefinierte Funktion MNR_KONV() vor dem Property PLMANNR (Mandantennummer) in die Objektstruktur eingefügt (siehe Abbildung 38). Diese Funktion berechnet die benutzerdefinierte Variable MNR. Der Wert dieser berechneten Variablen wird dann in der nächsten Zeile dem Property plManNr (Mandantennummer) zugewiesen. Auf diese Art und Weise wurde eine Konvertierung ehemals vierstelliger Mandantennummern auf fünfstellige erreicht.

Das Ergebnis der als Aktion eingefügten Funktion sieht dann folgendermaßen aus:



¹⁴ siehe auch Abschnitt 3.3 Arbeiten mit Variablen, **Beispiel 1**

¹⁵ Zur Erstellung von Funktionen siehe Abschnitt 4.3 Der Funktionseditor.



3.5 Modifizierung der Objektstruktur

Die über das Objektauswahlfeld festgelegte Objektstruktur kann modifiziert werden. So ist es möglich, einzelne Properties aber auch ganze Unterobjekte aus der Objektstruktur zu entfernen, wenn es dafür in den Quelldaten keine Entsprechung gibt und diese nicht benötigt werden. Die Objektstruktur wird dadurch vereinfacht und übersichtlicher. Es können aber auch Properties und Subobjekte hinzugefügt oder kopiert werden. Das Kopieren ist jedoch nur möglich, wenn in den Objekteigenschaften (Kontextmenü EDITIEREN) die Eigenschaft OBJEKT TRITT MEHRFACH AUF gesetzt ist.

Editieren der	Objektstruktur		×
Anzeige-Name	Kommunikation	•	<u>0</u> K
[tse:nit] Name	poKommunikation	•	<u>A</u> bbrechen
Objekt tritt meh	nfach auf		
Kommentar			



Beispiel 3

In einer Quelldatei MANDANT KOMMUNIKATION.CSV mit folgendem Dateiinhalt

```
Mandantennummer;Kurzbezeichnung;TelefonFirma;TelefaxFirma;TelefonPrivat;TelefaxPrivat;Mobiltelefon
4711;Beyer KG;0123-12345;0123-12346;0123-7890;0123-7891;0170-123456789
4712;Daimler GmbH;0456-98765;0456-98766;0456-976410;0456-976422;0171-987654321
```

Abbildung 42

sind die Kommunikationsdaten der Mandanten hinterlegt. Um diese verschiedenen Kommunikationsdaten in tse:nit einzulesen, muss das Objekt POKOMMUNIKATION (Kommunikation) mehrfach kopiert werden. Im Objektexplorer wird die Objektzeile POKOMMUNIKATION (Kommunikation) markiert und über das Kontextmenü das Objekt kopiert und wieder eingefügt. Auf diese Art werden vier Kopien des Objektes POKOMMUNIKATION (Kommunikation) erstellt.

Kopieren und Einfügen eines Objektes:







Abbildung 44



Nachdem das Objekt POKOMMUNIKATION (Kommunikation) mehrfach kopiert, die Datei MANDANT KOMMUNIKATION.CSV über den Textimportassistenten eingelesen, die Datei mit dem Objekt Mandant verbunden und die Felder der Textdatei den Properties der Objekte zugeordnet wurden, ergibt sich die unten dargestellte Objektstruktur. Zur besseren Übersicht wurden die Anzeigenamen der Kommunikationsobjekte editiert und nicht benötigte Objekte und Properties aus der Objektstruktur entfernt, so dass die Feldzuordnung einfacher vorgenommen werden kann. Der tse:nit Name der Objekte bleibt jedoch unverändert, da eine Änderung beim Import zu einem Fehler führt.

Normalansicht:

Anzeige im tse:nit Austauschformat:

Mandant <-> Mandant Kommunikation	Handant Z Mandant Kommunikation
Mandantennummer <-> {Mandant_Kommunikation("Mandantennummer") AsString}	Manuant <-> Manuant_Kommunikation
- Telefon Firma	pindana v (nandan_kation
Nummer <-> Mandant Kommunikation("TelefonEirma") AsString	psNummer (-> Mandant, Kommunikation("TelefonFirma") AsString
Bemerkung <-> {"Telefon Firma"}	psRemerkung <-> {"Telefon Firma"}
Hauptflag <-> {1}	pbBauntflag <-> {1}
- KommunikationTvp	
Tup <-> {1}	plKomMitTup <-> {1}
Bezeichnung <-> {"Telefon"}	nsBezeichnung <-> {"Telefon"}
🖂 Telefax Firma	
Nummer <-> Mandant_Kommunikation("TelefaxFirma").AsString	psNummer <-> Mandant Kommunikation("TelefaxFirma"),AsString
Bemerkung <-> {"Fax Firma"}	psBemerkung <-> {"Fax Firma"}
Hauptflag <-> {1}	pbHauptflag <-> {1}
🖃 KommunikationTyp	poKommunikationTyp
Typ <-> {2}	plKomMittTyp <-> {2}
Bezeichnung <-> {"Fax"}	psBezeichnung <-> {"Fax"}
- 🖂 Telefon privat	
Nummer <-> Mandant_Kommunikation("TelefonPrivat").AsString	psNummer <-> Mandant_Kommunikation("TelefonPrivat").AsString
Bemerkung <-> {"Telefon privat, nach 18:00"}	psBemerkung <-> {"Telefon privat, nach 18:00"}
Hauptflag <-> {0}	pbHauptflag <-> {0}
- KommunikationTyp	poKommunikationTyp
Тур <-> {1}	plKomMittTyp <-> {1}
Bezeichnung <-> {"Telefon"}	psBezeichnung <-> {"Telefon"}
🖃 Telefax privat	
Nummer <-> Mandant_Kommunikation("TelefaxPrivat").AsString	psNummer <-> Mandant_Kommunikation("TelefaxPrivat").AsString
Bemerkung <-> {"Fax privat"}	psBemerkung <-> {"Fax privat"}
Hauptflag <-> {0}	pbHauptflag <-> {0}
KommunikationTyp	poKommunikationTyp
Typ <-> {2}	plKomMittTyp <-> {2}
Bezeichnung <-> {"Fax"}	psBezeichnung <-> {"Fax"}
Mobilteleton	- poKommunikation
Nummer <-> Mandant_Kommunikation("Mobiltelefon").AsString	psNummer <-> Mandant_Kommunikation("Mobilteleton").AsString
Bemerkung <-> { "Handy"}	psBemerkung <-> {"Handy"}
	ppHaupmag <-> (U)
Tur () (1)	·····(=) por ommunikation i yp
Typ <-> (1) Demoister and (2) (Marking (2))	pikommitti yp <-> (1)
Dezeichnung <-> { Mobileleron }	psbezeichnung <-> (Mobilteleron)

Abbildung 45

Abbildung 46

Die Ausführung des so erstellten Scriptes mittels der Symbolschaltfläche 🖻 liefert das auf der folgenden Seite dargestellte Metafile.



Modifizierte Objekte müssen unbedingt der tse:nit Schnittstellenbeschreibung¹⁶ entsprechen, da Formatverletzungen beim tse:nit Import zu Fehlern führen.

¹⁶ siehe tse:nit Auslieferungs-CD im Verzeichnis <LW>\10it-Info\Produktinformation\Schnittstellen



Erstellte tse:nit Austauschformatdatei aus Beispiel 3:

```
ExportedSystem="Text-Import"
LangVer="3.00"
WorkstationID="0"
CodePage="ANSI"
Created="28.05.2002 13:27:33"
Comment=""
<Block, 28.05.02 13:27:33>
[Mandant]
  plManNr=4711
  [poKommunikation]
    psNummer=0123-12345
    psBemerkung=Telefon Firma
    pbHauptflag=1
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Telefon
    [END]
  [END]
  [poKommunikation]
    psNummer=0123-12346
    psBemerkung=Fax Firma
    pbHauptflag=1
    [poKommunikationTyp]
      plKomMittTyp=2
      psBezeichnung=Fax
    [END]
  [END]
  [poKommunikation]
    psNummer=0123-7890
    psBemerkung=Telefon privat, nach 18:00
    pbHauptflag=0
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Telefon
    [END]
  [END]
  [poKommunikation]
    psNummer=0123-7891
    psBemerkung=Fax privat
    pbHauptflag=0
    [poKommunikationTyp]
      plKomMittTyp=2
      psBezeichnung=Fax
    [END]
  [END]
  [poKommunikation]
    psNummer=0170-123456789
    psBemerkung=Handy
    pbHauptflag=0
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Mobiltelefon
    [END]
  [END]
[END]
[Mandant]
  plManNr=4712
  [poKommunikation]
    psNummer=0456-98765
    psBemerkung=Telefon Firma
    pbHauptflag=1
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Telefon
    [END]
  [END]
  [poKommunikation]
    psNummer=0456-98766
    psBemerkung=Fax Firma
    pbHauptflag=1
    [poKommunikationTyp]
```



```
plKomMittTyp=2
    psBezeichnung=Fax
[END]
  [END]
  [poKommunikation]
    psNummer=0456-976410
    psBemerkung=Telefon privat, nach 18:00
    pbHauptflag=0
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Telefon
    [END]
  [END]
  [poKommunikation]
    psNummer=0456-976422
    psBemerkung=Fax privat
    pbHauptflag=0
    [poKommunikationTyp]
     plKomMittTyp=2
      psBezeichnung=Fax
    [END]
  [END]
  [poKommunikation]
    psNummer=0171-987654321
    psBemerkung=Handy
    pbHauptflag=0
    [poKommunikationTyp]
      plKomMittTyp=1
      psBezeichnung=Mobiltelefon
    [END]
  [END]
[END]
<END>
END_OF_FILE
```



3.6 Das DATEI-VERBINDUNGSOBJEKT

Das DATEI-VERBINDUNGSOBJEKT ist ein virtuelles Objekt, das zur Verbindung von Unterobjekten mit Textdateien verwendet wird. Es ermöglicht, die Erstellung der Bedingungen für die Verbindung von Objekten zu Quelldateien zu vereinfachen. Bei diesem Objekt handelt es sich um ein spezielles Objekt, das nicht in das tse:nit-Austauschformat exportiert wird, sondern dass lediglich der Scriptsteuerung dient. Zur Veranschaulichung der Wirkungsweise soll das folgende Beispiel dienen.

Beispiel 4

Die Kommunikationsdaten (Telefon, Fax) für die Mandanten liegen einer Datei mit folgendem Inhalt vor:

```
Mandantennummer;Kurzbezeichnung;Kommunikationsnummer;Kommunikationstyp;Bemerkung
4711;Beyer KG;0123-12345;TelefonFirma;
4711;Beyer KG;0123-7890;TelefaxFirma;
4711;Beyer KG;0123-7891;TelefonPrivat;nach 18:00
4711;Beyer KG;0123-7891;Telefax;
4711;Beyer KG;0170-123456789;Mobiltelefon;
4712;Daimler GmbH;0456-98765;TelefonFirma;
4712;Daimler GmbH;0456-98766;TelefaxFirma;
4712;Daimler GmbH;0456-976410;TelefonPrivat;nur im Notfall
4712;Daimler GmbH;0456-976422;TelefaxPrivat;
4712;Daimler GmbH;0456-976422;TelefaxPrivat;
```

Abbildung 47

Zum Import dieser Daten empfiehlt sich der Einsatz des Datei Verbindungs-Objektes, da in diesem Fall die zu einem Mandanten gehörenden Kommunikationsdaten nicht in Spalten sondern zeilenweise hinterlegt sind.



Es wird ein Script mit folgendem Aufbau erstellt, wobei auch hier nicht benötigte Unterobjekte und Properties zur besseren Übersicht aus der Objektstruktur entfernt wurden. Außerdem wurde in diesem Beispiel mit einer weiteren Textdatei (MANDANT) gearbeitet, die lediglich als Mandantenverzeichnis dient.¹⁷ Die Kommunikationsdaten wurden über den Textimportassistenten in die logische Datei MANDANT_KOMMDATEN und das Mandatsverzeichnis in die logische Datei MANDANT eingelesen.



¹⁷ siehe auch Abschnitt 3.7 Das Zusammenführen mehrerer Quelldateien



Als Ergebnis von Beispiel 4 wird bei der Ausführung des Scriptes folgendes Metafile erstellt:

ExportedSystem="Text-Import" LangVer="3.00" WorkstationID="0" CodePage="ANSI" Created="30.05.2002 13:22:34" Comment="" <Block, 30.05.02 13:22:34> [Mandant] plManNr=4711 [poKommunikation] psNummer=0123-12345 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0123-12346 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation] psNummer=0123-7890 psBemerkung=nach 18:00 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0170-123456789 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [END] [Mandant] plManNr=4712 [poKommunikation] psNummer=0456-98765 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0456-98766 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation] psNummer=0456-976410 psBemerkung=nur im Notfall [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0456-976422 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation]



psNummer=0171-987654321
[poKommunikationTyp]
 plKomMittTyp=1
 psBezeichnung=Telefon
 [END]
 [END]
 [END]
 <END>
 END_OF_FILE



3.7 Das Zusammenführen mehrerer Quelldateien

In der Applikation TXTImport ist es möglich, Informationen aus mehreren Textdateien in eine tse:nit-Austauschformatdatei zusammenzuführen. Die Vorgehensweise dazu lässt sich folgendermaßen beschreiben:

- Einlesen der ersten Datei über den Menüpunkt DATEI | NEUE TEXTDATEI (Textimportassistent)
- Verbinden des Hauptobjektes mit dieser Textdatei (Kontextmenü der Objektstruktur, Menüpunkt <u>Verbinden mit Datei</u> xxx)
- Einlesen weiterer Textdateien über den Menüpunkt DATEI | NEUE TEXTDATEI (Textimportassistent)
- Verbinden der weiteren Textdateien mit den Unterobjekten des Hauptobjektes (Kontextmenü der Objektstruktur, Menüpunkt 🖲 Verbinden mit Datei...)

Folgendes Beispiel soll die Arbeit mit mehreren Dateien veranschaulichen.

Beispiel 5

Die Mandantendaten liegen in folgenden Dateien mit folgendem Inhalt vor:

• Mandant Adressen.CSV (allgemeine Mandantenadressdaten)

```
Mandantennummer;Kurzbezeichnung;Strasse;Postfach;PLZ_Strasse;PLZ_Postfach;Ort
4711;Beyer KG;An der Mühlenbreite;3520;12345;12346;Imlande
4712;Daimler GmbH;Zur Stadtmauer;9911;78945;78946;Nirgendwo
```

Abbildung 49

• Mandant Kommunikation.CSV (Kommunikationsdaten der Mandanten)

```
Mandantennummer;Kurzbezeichnung;Kommunikationsnummer;Kommunikationstyp;Bemerkung
4711;Beyer KG;0123-12345;TelefonFirma;
4711;Beyer KG;0123-7890;TelefonPrivat;nach 18:00
4711;Beyer KG;0123-7891;TelefonPrivat;nach 18:00
4711;Beyer KG;0123-7891;Telefax;
4711;Beyer KG;0123-7891;Telefax;
4712;Daimler GmbH;0456-98765;TelefonFirma;
4712;Daimler GmbH;0456-98766;TelefaxFirma;
4712;Daimler GmbH;0456-976410;TelefonPrivat;nur im Notfall
4712;Daimler GmbH;0456-976422;TelefaxPrivat;
4712;Daimler GmbH;0456-976422;TelefaxPrivat;
```

Abbildung 50

• Mandant Bank.CSV (Bankverbindungen der Mandanten)

```
Mandantennummer;Kurzbezeichnung;BLZ;BankKurzname;Bankname;Kontonummer
4711;Beyer KG;58550130;Sparkasse;Sparkasse Trier (Züsch);801411900
4711;Beyer KG;72050101;Kreissparkasse;Kreissparkasse Augsburg (Zusmarshausen);8999777
4711;Beyer KG;72069274;Raiffeisenbank;Raiffeisenbank Zusmarshausen-Altenmünster;9111333
4712;Daimler GmbH;57051870;Kreissparkasse;Kreissparkasse (Alf);1000200
4712;Daimler GmbH;58761343;Raiffeisenbank;Raiffeisenbank Zeller Land (Alf);5000000
```

Abbildung 51

Über ein geeignetes Script sollen diese drei Textdateien zu einer tse:nit Austauschformatdatei zusammengefasst werden.



Die Quelldaten sollten immer in sortierter Form vorliegen.



Im ersten Schritt wird die Datei MANDANT ADRESSEN.CSV über den Textimportassistent in das Datengrid eingelesen und über das Kontextmenü des Objektexplorers mit dem Hauptobjekt MANDANT verbunden Setzenden mit Datei Mandant_Adressen Objektstruktur entsteht. Da die Datei MANDANT_ADRESSEN in diesem Beispiel mit dem Hauptobjekt MANDANT verbunden wurde, dient sie gleichzeitig als Verzeichnisdatei für die Verbindung mit den weiteren Quelldateien.

🖻 Mandant <-> Mandant_Adressen	
Mandantennummer	
🕀 🕀 HausAnschrift	
E PostAnschrift	
• • • • • • • • • • • • • • • • • • •	

Abbildung 52

Das Property MANDANTENNUMMER (plManNr) wird mit dem Feld MANDANTENNUMMER der Adressdatei verbunden.

💁 Textdatei in [ts	e:nit] Austausch	format wandeln				
<u>]</u>	i <u>E</u> xtras <u>H</u> ilfe					
Mandant_Adresse	n 🔽 🗋	🧀 🖬 🔚 🗶 🖆] ¹ 😽 f*	🕒 🕞 🖸 C:\Pro	gramme\Wago-C	Jurad
Mandantennummer	Kurzbezeichnung	Strasse	Postfach	PLZ_Strasse	PLZ_Postfach	Ort
4711	Beyer KG	An der Mühlenbreite	3520	12345	12346	Im
4712	Daimler GmbH	Zur Stadtmauer	9911	78945	78946	Nig
Mandant □ Mandant <->	 Mandant_Adres	sen		Spaltenbeze	ichnung ummer	
pimannr		1	1	Kurzbezeich	nummer	
	ien mit Textgateispa	alte		Strasse	nang	
Verbind	ien mit <u>K</u> onstante ()			Postfach		
Verbind	len mit <u>F</u> ormel ()			PLZ_Strasse	•	
Formel	bearbeiten			PLZ_Postfac	ch	
ins For	meleingabefeld über	nehmen		Ort		
🔀 Verbind	lung lösen					



Die Objektstruktur stellt sich jetzt so dar.





Im nächsten Schritt werden die weiteren Textdateien über den Menüpunkt DATEI | NEUE TEXTDATEI eingelesen und die logischen Namen, MANDANT_KOMMDATEN und MANDANT_BANK vergeben. Über das Textdatei-Auswahlfeld in der Symbolleiste können jetzt die eingelesenen Dateien ausgewählt und deren Inhalt im Datengrid dargestellt werden.





Um das gewünschte Ergebnis zu erzielen, sind noch einige Änderungen an der Objektstruktur notwendig. Das Objekt HAUSANSCHRIFT wird markiert und über den Kontextmenüpunkt Verbindungsobjekt zur Datei erzeugen ein Verbindungsobjekt über diesem eingefügt.



Abbildung 56

Das Objekt HAUSANSCHRIFT ist damit um eine Hierarchiestufe tiefer gesetzt worden. Um jetzt das Objekt POSTANSCHRIFT ebenfalls als Unterobjekt des Datei Verbindungs-Objektes zu erhalten, wird es an der jetzigen Position gelöscht



Abbildung 57

und als neues Subobjekt des Datei Verbindungs-Objektes eingefügt (Kontextmenüpunkt Subobjekt hinzufügen).

Editieren der (×	
Anzeige-Name	PostAnschrift	-	<u>0</u> K
[tse:nit] Name	poPostAnschrift	•	Abbrechen
Objekt tritt mehrfach auf			
Kommentar			

Abbildung 58

Die Objektstruktur sieht dann so aus:



Abbildung 59

In der weiteren Bearbeitung werden die Properties mit den Feldern verbunden und ggf. Formeln hinterlegt (siehe Objekt POSTANSCHRIFT, Property STRASSE in der folgenden Abbildung).



E Mandant / > Mandant Adroson
E wannaur (-> wannaur Wilessen
Mandantennummer <-> Mandant_Adressen("Mandantennummer").AsString
Datei-Verbindungsobjekt
PLZ <-> Mandant_Adressen("PLZ_Strasse").AsString
Strasse <-> Mandant_Adressen("Strasse").AsString
Zusatz
Ort <-> Mandant_Adressen("Ort").AsString
€ Land
PLZ <-> Mandant_Adressen("PLZ_Postfach").AsString
Strasse <-> {"Postfach " & Mandant_Adressen("Postfach").AsString}
Zusatz
Ort <-> Mandant_Adressen("Ort").AsString
• Kommunikation
Bankkonto

Abbildung 60

Im weiteren Verlauf der Scripterstellung wird über dem Objekt Kommunikation ein Datei Verbindungs-Objekt erzeugt und dieses Datei Verbindungs-Objekt über das Kontextmenü Stebinden mit Datei... mit der Datei MANDANT_KOMMDATEN verbunden. Als Verbindungsformel wird

MANDANT_ADRESSEN("MANDANTENNUMMER").ASSTRING = MANDANT_KOMMDATEN("MANDANTENNUMMER").ASSTRING

hinterlegt.

		Auswahl der Datei
Datei-Zuordnung (Datei Verbindungsformel Suchbeschränkung	Link) auswählen	Bei Aktivierung wird die verbundene Datei bei jedem Scriptdurchlauf immer erneut vom Anfang durchlaufen, bis die Verbindungsformel erfüllt ist. Sollten die Daten sortiert abgelegt sein, kann diese Option deaktiviert werden.
Abbildung 6	1	Angabe der Verbindungsformel (Bedingung), die erfüllt sein muss, damit das Datei Verbindungs-Objekt ausgeführt wird.
		Angabe einer Suchbeschränkung, d.h. sobald die hier hinterlegte Bedingung erfüllt ist, wird die weitere Abarbeitung des Datei Verbindungs-Objektes beendet.

Um alle Kommunikationstypen einzulesen, wird für jeden Typ eine Kopie erstellt und für deren Ausführung eine Bedingung hinterlegt.

(z.B. **MANDANT_KOMMDATEN("KOMMUNIKATIONSTYP").ASSTRING = "TELEFONFIRMA"** wenn das Feld KOMMUNIKATIONSTYP der Datei MANDANT_KOMMDATEN den Wert "TelefonFirma" hat)



📮 Mandant <-> Mandant_Adressen
Mandantennummer <-> Mandant_Adressen("Mandantennummer").AsString
🖳 Datei-Verbindungsobjekt
- 🖓 HausAnschrift
PLZ <-> Mandant_Adressen("PLZ_Strasse").AsString
Strasse <-> Mandant_Adressen("Strasse").AsString
Zusatz
Ort <-> Mandant_Adressen("Ort").AsString
Land
PostAnschrift
PLZ <-> Mandant_Adressen("PLZ_Postfach").AsString
Strasse <-> {"Postfach " & Mandant_Adressen("Postfach").AsString}
Zusatz
Ort <-> Mandant_Adressen("Ort").AsString
E Land
🖂 Datei-Verbindungsobjekt <-> Mandant_KommDaten
Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefonFirma")
Entropy Bankkonto

Abbildung 62

Die Properties der Objekte KOMMUNIKATION werden mit den Feldern verbunden. Die Objektstruktur sieht dann folgendermaßen aus:

🔤 Datei-Verbindungsobjekt <-> Mandant_KommDaten	
🕞 Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefonFirma")	
Nummer <-> Mandant_KommDaten("Kommunikationsnummer").AsString	Kommunikationsmittel in der
Bemerkung <-> Mandant_KommDaten("Bemerk <u>ung"),As</u> String	Kanzlejakte
Hauptflag <-> {1}	Ranzielakie
	Typ = 1 Tolofon
Typ <-> {1}	Typ = 1 - Telefold
Bezeichnung <-> {"Telefon"}	Typ = 2 $-Tax$
🗁 Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefaxFirma")	Typ = 3 - email
Nummer <-> Mandant_KommDaten("Kommunikationsnummer").AsString	Typ = 4 - Telex
Bemerkung <-> Mandant_KommDaten("Bemerkung").AsString	Typ = 5 - Daten
Hauptflag	
🖳 🖂 KommunikationTyp	
Typ <-> {2}	
Bezeichnung <-> {"Fax"}	
🕀 Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefonPrivat")	
Nummer <-> Mandant_KommDaten("Kommunikationsnummer").AsString	
Bemerkung <-> Mandant_KommDaten("Bemerkung").AsString	
Hauptflag	
KommunikationTyp	
Typ <-> {1}	
Bezeichnung <-> {''Telefon''}	
- 🕀 Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefaxPrivat")	
Nummer <-> Mandant_KommDaten("Kommunikationsnummer").AsString	
Bemerkung <-> Mandant_KommDaten("Bemerkung"),AsString	
Hauptflag	
KommunikationTyp	
Typ <-> {2}	
Bezeichnung <-> {"Fax"}	
- ⊡ Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "Mobiltelefon")	
Nummer <-> Mandant_KommDaten("Kommunikationsnummer").AsString	
Bemerkung <-> Mandant_KommDaten("Bemerkung").AsString	
Hauptflag	
KommunikationTyp	
Typ <-> {1}	
Bezeichnung <-> {"Telefon"}	



Im Weiteren wird das Objekt BANKKONTO mit der Datei MANDANT_BANK verbunden. Als Verbindungsformel wird hier ebenfalls

MANDANT_ADRESSEN("MANDANTENNUMMER").ASSTRING = MANDANT_KOMMDATEN("MANDANTENNUMMER").ASSTRING

hinterlegt und die Feldzuordnung der Properties des Objektes BANKKONTO zu den Feldern der Datei MANDANT_BANK vorgenommen.



📮 Mandant <-> Mandant_Adressen
Mandantennummer <-> Mandant_Adressen("Mandantennummer").AsString
Datei-Verbindungsobjekt
🕀 HausAnschrift
PostAnschrift
🖳 Datei-Verbindungsobjekt <-> Mandant_KommDaten
🖅 Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefaxFirma")
⊕ Kommunikation (Mandant_KommDaten("Kommunikationstyp").AsString = "TelefonPrivat")
🖶 📮 Bankkonto <-> Mandant_Bank
Kontonummer <-> Mandant_Bank("Kontonummer"),AsString
🖃 Bank
BLZ <-> Mandant_Bank("BLZ").AsString
Name <-> Mandant_Bank("Bankname").AsString
Bezeichnung <-> Mandant_Bank("BankKurzname").AsString

Abbildung 64

Damit ist die Scripterstellung abgeschlossen und die Scriptausführung über die Symbolschaltfläche 🌬 liefert als Ergebnis die tse:nit Austauschformatdatei auf der folgenden Seite.



tse:nit Austauschformatdatei nach Ausführung des Scriptes aus Beispiel 5:

ExportedSystem="Text-Import" LangVer="3.00" WorkstationID="0" CodePage="ANSI" Created="30.05.2002 18:12:03" Comment="" <Block, 30.05.02 18:12:03> [Mandant] plManNr=4711 [poHausAnschrift] psPLZ=12345 psStrasse=An der Mühlenbreite psOrt=Imlande [END] [poPostAnschrift] psPLZ=12346 psStrasse=Postfach 3520 psOrt=Imlande [END] [poKommunikation] psNummer=0123-12345 pbHauptflag=1 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0123-12346 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation] psNummer=0123-7890 psBemerkung=nach 18:00 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0170-123456789 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poBankkonto] psKontonummer=801411900 [poBank] psBLZ=58550130 psName=Sparkasse Trier (Züsch) psKurzname=Sparkasse [END] [END] [poBankkonto] psKontonummer=8999777 [poBank] psBLZ=72050101 psName=Kreissparkasse Augsburg (Zusmarshausen) psKurzname=Kreissparkasse [END] [END] [poBankkonto] psKontonummer=9111333 [poBank] psBLZ=72069274 psName=Raiffeisenbank Zusmarshausen-Altenmünster psKurzname=Raiffeisenbank [END] [END] [END] [Mandant] plManNr=4712



[poHausAnschrift] psPLZ=78945 psStrasse=Zur Stadtmauer psOrt=Nirgendwo [END] [poPostAnschrift] psPLZ=78946 psStrasse=Postfach 9911 psOrt=Nirgendwo [END] [poKommunikation] psNummer=0456-98765 pbHauptflag=1 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0456-98766 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation] psNummer=0456-976410 psBemerkung=nur im Notfall [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poKommunikation] psNummer=0456-976422 [poKommunikationTyp] plKomMittTyp=2 psBezeichnung=Fax [END] [END] [poKommunikation] psNummer=0171-987654321 [poKommunikationTyp] plKomMittTyp=1 psBezeichnung=Telefon [END] [END] [poBankkonto] psKontonummer=1000200 [poBank] psBLZ=57051870 psName=Kreissparkasse (Alf) psKurzname=Kreissparkasse (AI [END] [END] [poBankkonto] psKontonummer=5000000 [poBank] psBLZ=58761343 psName=Raiffeisenbank Zeller Land (Alf) psKurzname=Raiffeisenbank [END] [END] [END] <END> END_OF_FILE



3.8 Eingaben während der Scriptausführung, die Funktion UserInput()

Die Funktion UserInput() ermöglicht es dem Anwender, Eingaben während der Scriptausführung vorzunehmen. Die Funktion USERINPUT() besitzt eine variable Parameteranzahl.

Syntaxnotation:

USERINPUT(PROMPT[, PROMPTFELDNAME1, PROMPTFELDNAME2, ...])

PROMPT	Textabfrage (Angabe in Hochkomma), die dem Anwender angezeigt wird
PROMPTFELDNAME	Feldnamen aus der aktuellen Datei, deren Werte angezeigt werden sollen

Der Aufruf der Funktion USERINPUT() erfolgt über den Formeleditor.

3.8.1 Verwendung der Funktion USERINPUT() zur Eingabe von Property-Werten

Beispiel 6

Es liegen Bankendaten in einer Datei mit folgendem Inhalt vor:

```
BankKurzname;Bankname
Sparkasse;Sparkasse Trier (Züsch)
Kreissparkasse;Kreissparkasse Augsburg (Zusmarshausen)
Raiffeisenbank;Raiffeisenbank Zusmarshausen-Altenmünster
Kreissparkasse;Kreissparkasse (Alf)
Raiffeisenbank;Raiffeisenbank Zeller Land (Alf)
```

Abbildung 65

In dieser Datei sind lediglich die Bankbezeichnungen und Bankkurzbezeichnungen hinterlegt. Während der Konvertierung in das tse:nit Austauschformat sollen die Bankleitzahlen durch Eingaben des Anwenders ergänzt werden. Dazu wird die Funktion USERINPUT() verwendet.

Im Objektauswahlfeld wird das Objekt BANK als Hauptobjekt ausgewählt. Die Textdatei wird über den Menüpunkt DATEI | NEUE TEXTDATEI in das Datengrid eingelesen und der logische Dateiname BANKEN vergeben. Das Objekt BANK wird mit der Datei BANKEN verbunden und die Zuordnungen für die Felder BANKNAME und BANKKURZNAME (Name) zu den Properties PSNAME und PSKURZNAME (Bezeichnung) vorgenommen. Die Objektstruktur sieht jetzt folgendermaßen aus:

🖃 Bank <-> Banken	
psBLZ	
psName <-> Banken("Bankname").AsString	
psKurzname <-> Banken("BankKurzname").AsString	

Abbildung 66

Durch Doppelklick auf das Property PSBLZ (BLZ) wird der Formeleditor geöffnet und aus der Liste der Funktionen kann die Funktion USERINPUT() ausgewählt und in die Eingabezeile übernommen werden. In der folgenden Abbildung wurde im Formeleditor die Funktion USERINPUT() mit den Parametern "BANKLEITZAHL" als Prompt und als PromptFeldName wird das Datei-Feld BANKEN("BANKNAME").ASSTRING verwendet.



🙀 Formel editiere	n					×
Feld	Variable	Funktion	Opera	tor		
🖃 Banken	blz	Sqr(Zahl)	0	Klammer	•	<u>0</u> K
BankKurzname		InStr(iStart,Text,Text[,Komparati	×	Multiplikation		Al-handhan
Bankname		LCase(Text)	1	Division		Abbrechen
FileName		UCase(Text)	Λ	Integer Division		
	1	Mid(Text,iStart,iLang)	Mod	Modulus		
		StrComp(Text1,Text2[,vbBinary(+	Addition		
		StrReverse(Text)	• •	Subtraktion		
		Len(Text)	- I	Gleich		
		Day(Datum)	\diamond	Ungleich		
		Month(Datum)	>	Größer als		
		Year(Datum)	<	Kleiner als		
		MonthName(month)	>=	Größer gleich		
		IIF	<=	Kleiner gleich		<u>H</u> uckgangig
Datastus	Datastus	IsNotNull	Not	Negation		<u>L</u> öschen
Datentyp	Datentyp	UserInput	And	Logisches UND		Übernschume
AsString 💌	AsString 🗾	Val(Text)	l Or	Loaisches ODER	•	Uperwachung
Formal Hearlanutf ¹⁰ D	ank laita ah!" Dank on ("D	ank name") AsString]	_		_	Test



Durch betätigen der Schaltfläche Lest kann die Funktion überprüft werden.

Werteingabe	×
Bankleitzahl	
12345678	
Aktuelle Feldwerte sind:	
Raiffeisenbank Zeller Land (Alf)=	<u>0</u> K
	Abbrechen



Abbildung 68

Mit der Schaltfläche im Formeleditor wird die Formel aus der Eingabezeile des Formeleditors dem Property zugeordnet.

🖻 Bank <-> Banken	
psBLZ <-> {UserInput("Bankleitzahl",Banken("Bankname"),AsString)}	
psName <-> Banken("Bankname").AsString	
psKurzname <-> Banken("BankKurzname").AsString	

Abbildung 70

Bei Verwendung der Funktion USERINPUT() in einem Script durchläuft der Konvertierungsprozess zwei Phasen. In der 1. Phase werden die Textdateien durchlaufen und eine Liste der einzugebenden Werte wird erstellt und in einer Tabelle angezeigt. Der Anwender kann darin die notwendigen Angaben vornehmen.

Bankleitzahl;Sparkasse Trier (Züsch)=	12345678
Bankleitzahl;Kreissparkasse Augsburg (Zusmarshausen)=	78945612
Bankleitzahl;Raiffeisenbank Zusmarshausen-Altenmünster=	32165498
Bankleitzahl;Kreissparkasse (Alf)=	14725836
Bankleitzahl;Raiffeisenbank Zeller Land (Alf)=	95175314



Nach betätigen der Schaltfläche erfolgt dann in der 2. Phase die eigentliche Konvertierung in das tse:nit Austauschformat. Als Ergebnis der Ausführung des so erstellten Scriptes wird das Metafile auf der folgenden Seite erstellt.



tse:nit Austauschformat Datei aus Beispiel 6:

```
ExportedSystem="Text-Import"
LangVer="3.00"
WorkstationID="0"
CodePage="ANSI"
Created="03.06.2002 14:23:20"
Comment=""
<Block,03.06.02 14:23:20>
[Bank]
 psBLZ=12345678
  psName=Sparkasse Trier (Züsch)
  psKurzname=Sparkasse
[END]
[Bank]
  psBLZ=78945612
  psName=Kreissparkasse Augsburg (Zusmarshausen)
  psKurzname=Kreissparkasse
[END]
[Bank]
 psBLZ=32165498
  psName=Raiffeisenbank Zusmarshausen-Altenmünster
  psKurzname=Raiffeisenbank
[END]
[Bank]
  psBLZ=14725836
  psName=Kreissparkasse (Alf)
  psKurzname=Kreissparkasse
[END]
[Bank]
 psBLZ=95175314
  psName=Raiffeisenbank Zeller Land (Alf)
  psKurzname=Raiffeisenbank
[END]
<END>
END_OF_FILE
```



3.8.2 Verwendung der Funktion USERINPUT() zur Eingabe von Variablen - Werten

Die Funktion USERINPUT() kann außer zur Eingabe von Property-Werten auch zur Wertzuweisung von Variablen zur Laufzeit des Scriptes verwendet werden. Die Anwendung der Funktion unterscheidet sich in diesem Fall geringfügig.

Beispiel 7

Beim Import von Buchungsdaten sollen zur Laufzeit eines Scriptes das Buchungsjahr und der aktuelle Buchungsmonat für die einzulesenden Buchungsdaten angegeben werden.

Es werden zwei Variablen, BUCHUNGSJAHR und BUCHUNGSMONAT definiert, deren Werte zur Laufzeit des Scriptes eingegeben werden sollen.

× y Variable			×
Variable Name	Datentyp	Initial-Wert	
Buchungsjahr	AsString	101	
Buchungsmonat	AsString	100	Abbrechen
			<u>N</u> eu Löschen <u>B</u> earbeiten

Abbildung 72

Im Aktionseditor wird die folgende Aktion erstellt

20 Aktion editieren				×
Feld	Variable	Funktion	Operator	
	Buchungsjahr	Month(Datum)	() Klammer 🔺	<u>0</u> K
	Buchungsmonat	Year(Datum)	* Multiplikation	Abbeecher
		MonthName(month)	7 Division	Abbrechen
		llf	\ Integer Division	
		IsNotNull	Mod Modulus	Bijckaäpaja
		UserInput	+ Addition	
Datentup	Datentup	Val(Text)	- Subtraktion	<u>L</u> öschen
AsChing	Asching	WriteToLog(Filename, Level, InI	= Gleich _1	Überwachung
Assung	Assung			<u></u>
Aktion Variables("Bu	chungsjahr'').AsString =	UserInput("Bitte geben Sie das Buch	ungsjsahr an!")	<u> </u>

Abbildung 73

und diese dann in die Objektstruktur eingefügt. Ebenso wird für die Angabe des Buchungsmonates eine Aktion erstellt und diese dann in die Objektstruktur eingefügt. Dann werden die Variablen den entsprechenden Properties (Monat des Wirtschaftsjahres und Buchungszeitraum) zugewiesen.

Auszug aus der Objektstruktur MANDANT_BUCHUNGEN nach Einfügen der Aktionen:



Abbildung 74

Zur Laufzeit des Scriptes werden jetzt die beiden Variablen - Werte abgefragt.





3.9 Erstellen einer Protokolldatei, die Funktion WRITETOLOG()

Unter Verwendung der Funktion WRITETOLOG() ist es möglich, während der Laufzeit eines Scriptes eine Protokolldatei zu erstellen und in diese Feldwerte, Variablenwerte, Kommentare etc. zu schreiben. Der Aufruf dieser Funktion erfolgt innerhalb von benutzerdefinierten Funktionen oder Prozeduren über den Befehl CALL.

Syntaxnotation		WRITETOLOG(FILENAME, [LEVEL], INFO, [CLEARATINIT])
FILENAME LEVEL: INFO: CLEARATINIT:	:	Name der LOG - Datei in Hochkomma (z.B. "c:\Protokoll.txt") Einrückung, Anzahl Leerzeichen Text, der geschrieben wird, in Hochkomma 1-Inhalt wird bei Initialisierung gelöscht 0-Inhalt bleibt erhalten (anfügen)

Beispiel 8

Das Beispiel 5 wird dahingehend abgewandelt, dass eine Prüfung der Kontonummern und Bankleitzahlen erfolgt. Es erfolgt ein Eintrag in der Protokolldatei "C:\Bankenimport.log", sofern keine Kontonummer übergeben wird oder die Länge der Bankleitzahl <> 8 ist. Im Funktionseditor¹⁸ wird die folgende Funktion erstellt:

function log()

```
if len(Mandant_Bank("Kontonummer").AsString) = 0 then
call WriteToLog("C:\Bankenimport.log",0,Mandant_Bank("Mandantennummer").AsString & ", Die
Bankverbindung ist unvollständig",1)
end if
if len(Mandant_Bank("BLZ").AsString) <> 8 then
call WriteToLog("C:\Bankenimport.log",0,Mandant_Bank("Mandantennummer").AsString & ", " &
Mandant_Bank("Bankname").AsString & ", Die BLZ ist fehlerhaft",0)
end if
```

end function

Der Aufruf dieser Funktion erfolgt in der Objektstruktur unter dem Objekt BANKKONTO als eingefügte Aktion.



Abbildung 76

Als Ergebnis liefert diese Funktion in der Datei "C:\Bankenimport.log" Protokolleinträge, sofern fehlerhafte Bankverbindungen in der Textdatei übergeben werden.

Bankenimport.log - Editor	
<u>Datei Bearbeiten Suchen ?</u>	
4711, Die Bankverbindung ist unvollständig 4712, Die Bankverbindung ist unvollständig 4712, Raiffeisenbank Zeller Land (Alf), Die BLZ	ist fehlerhaft
<u> </u>	

¹⁸ siehe Abschnitt 4.3 Der Funktionseditor



4 Anhang

4.1 Der Textimportassistent

Über den Textimportassistenten erfolgt das Einlesen von Quelldateien (Textdateien, Excel-Dateien), die in das tse:nit Austauschformat konvertiert werden sollen. Nachdem über den Menüpunkt DATEI | NEUE TEXTDATEI eine Textdatei im Format *.asc, *.chr, *.csv, *.dat, *.txt ausgewählt wurde, wird das nebenstehende Formular geöffnet.

😰 Textimport-Assistent 🔀	
Der Assistent hat erkannt, daß Ihre Daten in Format 'Mit Trennzeichen' vorliegen. Sollte das nicht der Fall sein, wählen Sie bitte ein Format aus, das Ihre Daten genauer beschriebt. - Wählen Sie das Format aus, das Ihre Daten am besten beschreibt:	
Mit Trennzeichen - Felder sind durch Zeichen, wie z.B. Komma oder Tabulator, voneinander getrennt.	
Feste Breite - Felder sind in Spalten angeordnet, die durch Leerzeichen unterteilt sind.	
Beispieldaten aus Datei: C:\Programme\Wago-Curadata\10it\MetaFile\Mandant	_ Auswahl des Datenformates
Mandantennummer;Kurzbezeichnung;Name;Vorname;Titel;Geburtstag 4711;Beyer KG;Beyer;Anton;Dr.;Oll11966;Dipl. Betriebswirt;lt; Unternehmensberatung";Herrn Beyer;;Ol.Ol. 95;31.12.01; 4712;Daimler GmbH;Daimler;Christopher;;O6121957;Kfz-Meister;rl Freie Kfz Werkstatt";Herrn H. Bühne;;;	Festlegen der Datei- und Verarbeitungsoptionen
	_ zum nächsten Fenster
Einstellungen Abbrechen Zurück Weiter > Fertigstellen	

Abbildung 78

In diesem Formular wird das Eingabeformat der Quelldatei festgelegt.

MIT TRENNZEICHEN Verwendung bei CSV-Dateien¹⁹, d.h. die einzelnen Feldwerte in den Datensätzen sind durch Trennzeichen getrennt

FESTE BREITE Anordnung der Feldwerte in den Datensätzen mit fester Spaltenbreite

Über die Schaltfläche Enstellungen können weitere Angaben zum Format der Quelldaten und zu den Verarbeitungsoptionen vorgenommen werden.

¹⁹ CSV - Character Separated Value



Auswahl des Datenformates	Datei-Einstellungen - Mandant_Doku	Angabe der Feldtrennzeichen (,;{tab}{space} oder freie Eingabe)
Übernahme der Feldbezeichnungen aus der ersten Zeile der	Grundangaben zur Textdatei Dateitormat Mit Trennzeichen Feldtrennzeichen : C Feste Breite Textbegrenzungszeichen "	Angabe für Textbegrenzung
Textdatei, eventuell fehlerhafte Bezeichnungen werden angepasst.	Microsoft Excel Erste Reihe enthält Feldnamen Anzahl der zu überlesender Kopfzeilen Anzahl der zu überlesender Kopfzeilen	Angabe der Kopfzeilen, die beim Import überlesen werden sollen
Angabe der Spaltenanzahl	Dateiberkunft Windows (ANSI) Datei in Speicher vorberaiten Verarbeitungsoptionen für die Datei	Verknüpfung von n aufeinanderfolgenden Zeilen der Quelldatei (Textdatei) zu einer Zeile der logischen
Angabe zur Datenherkunft, Windows (ANSI) oder DOS (ASCII)	Tausendertrennung . Datumstrennung . Wert für Falsch N	Datei
Angabe der Anzahl Zeilen, die im Datengrid angezeigt werden	Genutzte Dateizeilen für die Scripterstellung C Alle C von Zeilen 1 bis Zeile 20 C Scripterstellung C Scripterstellung	Projektstatus festlegen, nur aktiv, wenn aktueller Status = Scripterstellung und Aufruf der Einstellungen über die Symbolschaltfläche
	Abbildung 79	erfolgt

Im Bereich VERARBEITUNGSOPTIONEN FÜR DIE DATEI können die folgenden Optionen eingestellt werden:

DEZIMALTRENNUNG:	Dezir	naltrennung erfolgt dur	ch:	
	"."	(Punkt)	z.B.	1.23
	""	(Komma)	z.B.	1,23
TAUSENDERTRENNUNG:	Treni	nzeichen zwischen Taus	sendern	ist:
		(Leerzeichen)	z.B.	1 234 567,89
	"."	(Punkt)	z.B.	1.234.567,89
	",	(Komma)	z.B.	1,234,567.89
DATUMSREIHENFOLGE:	Reihe	enfolge der Datumsanga	abe (in V	/erbindung mit Datumstrennzeichen):
	TMJ	(Tag Monat Jahr)	z.B.	30052001
	TJM	(Tag Jahr Monat)	z.B.	30-2001-05
	MTJ	(Monat Tag Jahr)	z.B.	05.30.2001
	MJT	(Monat Jahr Tag)	z.B.	05/2001/30
	JTM	(Jahr Tag Monat)	z.B.	2001:30:05
	JMT	(Jahr Monat Tag)	z.B.	20010530
DATUMSTRENNUNG:	Treni	nzeichen in Datumsanga	aben:	
	"."	(Punkt)	z.B.	29.10.2001
	"-"	(Bindestrich)	z.B.	15-9-01
	"/"	(Schrägstrich)	z.B.	4/6/99
	":"	(Doppelpunkt)	z.B.	9:12:87
	{no}	(kein Trennzeichen)	z.B.	150698
WERT FÜR WAHR:	logiso "True "Ja" "Yes" "On" "T" "Y" "J" "J"	cher Wert für "Wahr" is	st repräs	sentiert als:



WERT FÜR FALSCH: logischer Wert für "Falsch" ist repräsentiert als: "False" "Nein" "No" "Off" "F" "N" "0" (Ziffer 0)

Erläuterung zur Option PROJEKTSTATUS:

Wenn der Dialog EINSTELLUNGEN aus dem Textimport-Assistenten aufgerufen wird, ist die Option DATEIDEFINITION aktiviert und es können alle vorhandenen Einstellungen geändert werden. Im anderen Fall, dem Aufruf aus dem Menü oder über die Symbolschaltfläche III, ist die Option Scripterstellung eingestellt und man kann nur Einstellungen vornehmen, deren Änderung die Anzahl der Felder (Spalten) und deren Namen nicht beeinflusst. Alle Einstellungen können vom Anwender explizit geändert werden, wenn man in die Option DATEIDEFINITION umschaltet. In diesem Fall wird der Anwender vor der Umschaltung gewarnt, da jetzt die Feldnamen modifiziert werden können und es so zu einer "Entwertung" der früher definierten Verbindungen kommen kann und damit das bereits erstellte Script unbrauchbar wird.

4.1.1 Einlesen von Daten im Format MIT TRENNZEICHEN

Zab C genakolon C komma C gerezenen C generes Textbegrenzungszeichen: " Erste Zeile enthält die Feldnamen Textbegrenzungszeichen: " Beispieldaten Mandantennummer Kurzbezeichnung Name Vorname Titel Geburtstag 4711 Beyer KG Beyer Anton Dr. 01111966 4712 Daimler GmbH Daimler Christopher 06121957	_ g Beru	eichen: " Geburtstag	tungsze Titel	Textbegrenz	Mart	i die F <u>e</u> ldnamen	⊂ ⊥ab
Ziste Zeile enthält die Feldhamen Textgegrenzungszeichen: " eispieldaten Mandantennummer Mandantennummer Kurzbezeichnung Name Vorname Titel Geburtstag 4711 Beyer KG Barre Anton Dr. 01111966 4712 Daimler GmbH Daimler Christopher 06121957	g Ben	eichen: " Geburtstag	Titel	Text <u>b</u> egrenz	Marra	: die F <u>e</u> ldnamen	Erste Zeile enthält eispieldaten
eispieldaten Mandantennummer Kurzbezeichnung Name Vorname Titel Geburtstag 1711 Beyer KG Beyer Anton Dr. 01111966 1712 Daimler GmbH Daimler Christopher 06121957	g Beru	Geburtstag	Titel	Vornamo	Marra		eispieldaten
eispieldaten Mandantennummer Kurzbezeichnung Name Vorname Titel Geburtstag 1711 Beyer KG Beyer Anton Dr. 01111366 1712 Daimler GmbH Daimler Christopher 06121957	g Beru	Geburtstag	Titel	Vornamo	News		eispieldaten
espielden Mandanternummer Kurzbezeichnung Name Vorname Titel Geburtstag 771 Beyer KG Beyer Anton Dr. 01111965 4712 Daimler GmbH Daimler Christopher 06121957	g Beru	Geburtstag	Titel	Vername	hlana		eispieldaten
Mandanlernummet Kurzbezeichnung Name Vorname Ittel (seburtsay 771 Beyet KG Beyer Anton Dr. 01111965 1712 Daimler GmbH Daimler Christopher 06121957	g Beru	Geburtstag	litel	Morphone	the second se	1.4 1 1 1	
771 Beyer Ka Beyer (Anton Dr. 01111566 7712 Daimler GmbH Daimler Christopher 06121957	1.61.1	01111000	N	vomanie	Name	Kurzbezeichnung	4andantennummer
	U IDI	00111300	UI.	Christenhor	Deimler	Deven Na Disimler GrebH	712
	NI21	00121337		Chilstopher	Dainiei	Damierumon	112
	D						

Nachdem über die Schaltfläche Entelungen alle erforderlichen Angaben gemacht wurden, gelangt man über die Schaltfläche Weter in das nächste Fenster (siehe Abbildung 80), in dem noch einmal Einstellungen zu dem gewählten Format vorgenommen werden können.

Abbildung 80



Wiederum mit der Schaltfläche weer gelangt man in das folgende Fenster (siehe Abbildung 81), in dem der logische Name der Textdatei und die logischen Namen der Datenfelder und deren Datentypen festgelegt werden. In der unteren Hälfte dieses Fensters werden zur Kontrolle der Einstellungen Beispielwerte aus der Quelldatei angezeigt.

Abbildung 81

Über die Schaltfläche Letigstellen werden die Datensätze aus der Textdatei in das Datengrid eingelesen.



4.1.2 Einlesen von Daten im Format FESTE BREITE



Nachdem über die Schaltfläche Endeurgen alle erforderlichen Angaben gemacht wurden, gelangt man über die Schaltfläche weer in das nächste Fenster (siehe Abbildung 82), in dem die Festlegung der Spaltenbreite der Datenfelder vorgenommen wird. Zum Erstellen einer Feldtrennlinie wird die gewünschte Position mit der linken Maustaste angeklickt. Zum Löschen einer Trennlinie wird diese erneut mit der linken Maustaste angeklickt. Zum Verschieben einer Trennlinie wird diese mit gedrückter linker Maustaste an ihre neue Position gezogen.

Abbildung 82

12 Textimport-Assistent								
Bearbeiten des Dateinamens und der Spaltenbezeichnungen, der aus der Text-Datei erzeugten Tabelle.								
<u>N</u> ame:	Mandant_Doku	1		Spalter	bezeichnung	Datentyp		Breite
				Mnr		AsString	▣	9
				KBez		AsString		20
				Name		AsString		20
				Vornam	e	AsString		20
				Titel		AsString		5
Detected	lala ta m							
Deispier	luateri		L	·	(·			
Mnr	KBez	Name	Vorr	name	Titel			
4711	Beyer KG	Beyer	Anto	n	Dr.			
4712	Daimler GmbH	Daimler	Chri	stopher				
Einstellungen Abbrechen < Zurück Weiter > Fertigstellen								

Wiederum mit der Schaltfläche weter gelangt man in das folgende Fenster (siehe Abbildung 83), in dem der logische Name der Textdatei und die logischen Namen der Datenfelder und deren Datentypen festgelegt werden. In der unteren Hälfte dieses Fensters werden zur Kontrolle der Einstellungen Beispielwerte aus der Quelldatei angezeigt.

Abbildung 83

Über die Schaltfläche <u>Ferigatelen</u> werden die Datensätze aus der Textdatei in das Datengrid eingelesen.



4.1.3 Einlesen von Daten im EXCEL - Format

Γ	a. Textimport-Assi	stent					×		
	Auswählen der Tabelle für den Import								
	Mandantendaten								
	✓ Erste Zeile enthält die Feldnamen								
	Mandantennummer	Kurzbezeichnung	Name	Vorname	Titel	Geburtstag	Beruf		
	4711	Beyer KG	Beyer	Anton	Dr.	1111966	Dipl.		
	4712	Daimler GmbH	Daimler	Christopher		6121957	Kfz-M		
	٩						Þ		
	(Einstellungen)	Abbrechen	< Zurück	eit	er>	Eertigst	ellen		

Bei der Auswahl von Excel (*.xls) Dateien über den Menüpunkt DATEI | NEUE TEXTDATEI öffnet sich das nebenstehende Fenster. Die Auswahl des Excel-Tabellenblattes, aus dem die Daten eingelesen werden sollen, erfolgt über ein Listenfeld.

Abbildung 84

Über die Schaltfläche **Entelungen** sind in der Regel keine Angaben zu machen, da die Datei- und Verarbeitungsoptionen schon durch die verwendeten Formate im Excel-Sheet vorgegeben und durch den Importassistenten erkannt werden. Bis auf die u.a. Optionen sind beim Import von Excel-Daten keine Angaben möglich.







Nachdem über die Schaltfläche Entekungen alle erforderlichen Angaben gemacht wurden, gelangt man über die Schaltfläche weters in das nächste Fenster (siehe Abbildung 86), in dem der logische Name des Excel-Datenblattes und die logischen Namen der Datenfelder und deren Datentypen festgelegt werden. In der unteren Hälfte dieses Fensters werden zur Kontrolle der Einstellungen Beispielwerte aus der Quelldatei angezeigt. Über die Schaltfläche Fengeten wird die unter Einstellungen festgelegte Anzahl Zeilen in das Datengrid eingelesen.



4.2 Der Formel- / Aktionseditor

Der Formel - Editor ermöglicht es, das Erstellen von Ausdrücken (Berechnungsformeln) zu vereinfachen. Der Aufruf erfolgt mittels der Symbolschaltfläche im neben dem Formeleingabefeld bzw. dem Aktionseingabefeld, durch Doppelklick auf eine Zeile im Objektexplorer oder über das Kontextmenü im Funktionseditor (Menüpunkt FORMEL EINFÜGEN oder AKTION EINFÜGEN).



Abbildung 87

Das Editorfenster enthält der Reihe nach: Liste der Felder aus den aktuellen Textdatein, Liste der Benutzervariablen, Liste der wichtigsten Funktionen²⁰ und Liste der Operatoren.

Mittels Doppelklick auf einen Eintrag in den Auswahllisten, durch Betätigen der Leertaste oder via Drag & Drop bei gleichzeitigem Drücken der [Strg] Taste werden die jeweiligen Einträge in das Eingabefeld (je nach Aufruf des Formeleditors mit der Bezeichnung FUNKTION oder AKTION) zur weiteren Ausdrucksbearbeitung übernommen.

Der aktuelle Wert von Feldern und Benutzervariablen wird als Tooltip angezeigt, sobald der Mauszeiger auf den jeweiligen Eintrag positioniert wird.

Der Inhalt des Listenfeld DATENTYP zeigt immer den aktuellen Datentyp des markierten Feldes / der markierten Variable. Wird ein anderes Feld oder eine andere Variable ausgewählt, ändert sich entsprechend auch der Inhalt dieses Listenfeldes.

Die Schaltfläche Ubewechung bietet dem Anwender die Möglichkeit, aktuelle Werte von mehreren Ausdrücken zu überwachen. Sie dient ebenfalls als Schalter für das Ein- und Ausblenden der Tabelle zur Ausdrucksüberwachung.

Mit der Schaltfläche Rückgänge werden die letzten ausgeführten Eingaben gelöscht.

Über die Schaltfläche Abbrechen wird der Formeleditor geschlossen, ohne die vorgenommenen Änderungen zu speichern.

²⁰ Außer den enthaltenen Funktionen können weitere Funktionen des VBScript Sprachumfanges genutzt werden.



Aktion editieren								
Feld	Variable	Funktion		Operat	or			
Mandant Mandant_Au Mandantennu Kurzbezeichnu Strasse Postfach PLZ_Strasse PLZ_Boxtfack	DSNr MNr Dseit	Abs[Zahl] Frac[Zahl] Fix[Zahl] Round[Zahl[, Anz. Sqr[Zahl] InStr([Start,Text,Ti LCase(Text] UCase(Text]	al Dezimalstell	0 * / \ Mod + -	Klammer Multiplikation Division Integer Division Modulus Addition Subtraktion Gleich		<u>OK</u> <u>Abbrechen</u> <u>Rückgängig</u>	
Datentyp AsString	Datentyp AsInteger	Mid(Text,iStart,iLang) StrComp(Text1,Text2[,vbBinary($\stackrel{\diamond}{}$	Ungleich Größer als	•		
Aktion Variables("DS	Aktion Variables("DSNr").AsInteger = Variables("DSNr").AsInteger + 1							
Überwachungsausdruck Wert Variables("DSNr").AsInteger 0 Mandant_Adressen("Mandantennummer").AsString 4711 Mandant_Adressen("Kurzbezeichnung").AsString Beyer KG								

Abbildung 88

Überwachungsausdrücke werden über das Kontextmenü im Grid oder per Drag & Drop bei gleichzeitigem Drücken der [Strg] Taste von Listeneinträgen hinzugefügt.

Kontextmenü:

<u>E</u> ditiere Überwachung Hinzufügen Über <u>w</u> achung (Einf)	 Möglichkeit der Modifizierung des aktuellen Ausdrucks Dialog zur Eingabe eines neuen Ausdrucks
Hinzufügen <u>A</u> usdruck (Variables("DSNr").As…) Hinzufügen ⊻ariable (DSNr) Hinzufügen <u>F</u> eld (Mandantennummer)	 Ausdruck aus der Eingabezeile übernehmen Hinzufügen der aktuell gewählten Variablen Hinzufügen des aktuell gewählten Feldes
Entfernen Überwachung (Entf) Entfernen Alle Überwachungen aktualisieren	- Ausgewählte Überwachung Löschen - Alle Überwachungen aus der Liste entfernen - Aktualisieren der Überwachungen in der Liste

Abbildung 89

Durch Betätigen der Schaltfläche Jett wird versucht, die in der Eingabezeile hinterlegte Formel zu berechnen, was zur Kontrolle des erstellten Berechnungsausdrucks dient. Es erfolgt eine Syntaxprüfung und Berechnung der eingegebenen Formel.

Nach Betätigung der Schaltfläche wird der erstellte Ausdruck in das Formeleingabefeld, das Aktionseingabefeld oder den Funktionseditor übernommen.



4.3 Der Funktionseditor

Zur Vereinfachung von Ausdrücken oder zur Berechnung komplizierter Ausdrücke ist es im TXTImport möglich, benutzerdefinierte Funktionen und Prozeduren zu erstellen, und diese mit dem Script, genau wie die benutzerdefinierten Variablen, zu speichern.

Der Aufruf des Funktionseditors erfolgt entweder über den Menüpunkt BEARBEITEN | FUNKTION oder über die Symbolschaltfläche 🖻 in der Programmoberfläche. Es öffnet sich das folgende Formular.

fæ Benutzerfunktionen		Eingabe von Funktions-Code
function mnr_konv()		
Variables("MNr").AsInteger = 10000 + val(Mandant("Mandantennummer").AsString)		Eingabe eines Berechnungsausdruckes zu Testzwecken
end function		
Eormel val(Mandant("Mandantennummer").AsString)	<u>I</u> est	
	<u>OK</u> <u>Abbrechen</u>	

Abbildung 90

Die Abbildung zeigt die Funktion MNR_KONV(), mit deren Hilfe man zum Beispiel eine Konvertierung der vierstelligen Mandantennummern in fünfstellige Mandantennummern erreichen könnte. Die Berechnung wird in dieser Funktion der Variablen VARIABLES("MNR").ASINTEGER zugewiesen, die dann im Objekt-explorer mit dem Property PLMANNR (Mandantennummer) verknüpft wird. Die benutzerdefinierte Funktion wird an geeigneter Stelle in die Objektstruktur als Aktion eingefügt (siehe Abbildung 91).

Im Eingabefeld FORMEL im unteren Teil des Formulars kann ein Testausdruck eingegeben werden, der dann über die Schaltfläche Lett berechnet wird. Falls die Funktion fehlerhaft ist, wird eine Fehlermeldung angezeigt und der Cursor wird in die Code Zeile gesetzt, in der der Fehler erkannt wurde.

Auszug aus der Objektstruktur für das Objekt MANDANT (Mandant):





Kontextmenü im Funktionseditor:



Abbildung 92

Das Kontextmenü enthält neben den Standard Editor Bearbeitungsbefehlen folgende Menüpunkte:

VORLAGE FÜR FUNKTIONSKÖRPER:	An der aktuellen Cursorposition wird ein Muster für einen Funktionskörper eingefügt:			
	Function Function_Name() 'Name durch Funktionsnamen ersetzen!			
	Function_Name = End Function			
	Danach ist die eingefügte Funktion umzubenennen (Function_Name() durch wirklichen Funktionsnamen ersetzen) und der Funktionsinhalt in Form von Programmanweisungen (Befehlen) einzugeben.			
Formel einfügen:	Der Formeleditor wird aufgerufen. Bilden und testen eines Berechnungsausdrucks im Formeleditor. Der so gebildete Berechnungsausdruck wird an der aktuellen Cursorposition in den Formeleditor eingefügt.			
AKTION EINFÜGEN:	Der Aktionseditor wird aufgerufen. Die gebildete Aktion wird an der aktuellen Cursorposition in den Funktionseditor eingefügt.			



4.4 Debug-Modus

Über die Funktionstaste **[F8]** wird der Debug-Modus eingeschaltet. In diesem speziellen Programm-Modus besteht die Möglichkeit, Scriptdurchläufe zu testen und anhand von Überwachungsausdrücken die Wertzuweisung zu Properties oder Variablen zu überprüfen. Weiterhin besteht im Debug-Modus die Möglichkeit, in der Objektstruktur so genannte Haltepunkte zu setzen, bei denen die Abarbeitung eines Scriptes unterbrochen wird.

Funktionstastenbelegung im Debug-Modus:

- [F8] Ausführen Einzelschritt (Symbolschaltfläche 🕨)
- [F5] Ausführen bis Haltepunkt bzw. Scriptende (Symbolschaltfläche 🕨)
- [ESC] Scriptausführung beenden, Rückkehr in den Definitions-Modus (Symbolschaltfläche 🔳

Bildschirmansicht im Debug-Modus:

😰 Textdatei in [tse:nit] Austauschformat wandeln - C:\Programme\\Mandantenimport Adressen,KommDaten,Banken.opt									
Detei Bearbeiten Extras Hille									
Mandant_Bank 🔽 🗅 😂 🖩 🗮 🗙 🚰 🦄 🎄 🕒 🕨 א 🔳 C:\Programme\Wago-Curadata\10it\MetaFile\Mandant.mta									
Mandantennummer Kurzbezeichnung BLZ BankKurzname Banknar	ime	Kontonummer							
4711 Beyer KG 58550130 Sparkasse Sparkas	sse Trier (Züsch)	801411900							
4711 Beyer KG 72050101 Kreissparkasse Kreisspa	arkasse Augsburg (Zusmarshausen)	8999777							
4/11 Beyer KG 72059274 Raiffeisenbank Raiffeise	enbank Zusmarshausen-Altenmunster	9111333		Farbliche Markierung zur					
4712 Damler GmbH 57051870 Kreissparkasse Kreisspa 4710 Deieler GebU 5070104 Deiffeirenbergh Deiffeire	arkasse (Alt)	T000200		Kennzeichnung der aktuellen					
14/1/ Illaimet 13076134 Bairreisenbank Bairreise	ennank / eller i and i bit	501000		Position der Scriptausführung					
Mandant	_/								
🗘 🖻 Mandant <-> Mandant_Adressen 🥏	Überwachungsausdruck		Wert						
Mandantennummer <-> Mandant_Adressen("Mandantennummer").AsString	g Mandant_Adressen("Mandanten	nummer").AsString	4711						
Dater-Verbindungsobjekt	Mandant_KommDaten["Mandant	tennummer"].AsString	4/11	Anzeige von					
Datei-Verbindungsobjekt <-> Mandant_KommUaten	Mandant_Bank("Mandantennum	mer"J.AsString	4/11 Mandaut Kanana Jania	Überwachungsausdrücken,					
Kontonummer () Mandant_Bank	Mandant_KommDiaten, FileName		Hinzufügen, Entfernen, Editieren						
	Valiables(biz).Assoling			von Überwachungen über die					
BLZ <> {Variables("blz").AsString}		/	rochto Maustasto						
Name <-> Mandant Bank("Bankname").AsString				Techie Mausiasie					
Bezeichnung <-> Mandant_Bank("BankKurzname").AsString			L						
		E ditiere Ü	berwachung						
		Hinzufija	en Überwachung (Einfl						
Setzen von Hal	Itepunkten durch								
Doppelklick im	rau markierten Entfernen Überwachung (Entf)								
Bereich des Ob	ektexplorers Entfermen Alle								
	🗘 Überwachungen aktualisieren								
				·					
	Kanatanta								
	Konstante								
	Formel								
	Aktion								

Abbildung 93