***3.Programmierung***

Der Abschnitt Programmierung unterteilt sich in

   3.1  Prozeduren

   3.2. Module

   3.3 Auswertungen

Durch diese Einteilung kann ein Programmierer sich leichter Einarbeiten und bei einer Fehlerdiagnose gezielter Vorgehen.

***3.Programmierung / Prozeduren***

**3.1 Prozeduren**

Die Prozeduren enthalten in Klartext die Programmierung und werden mit dem Conzept16-Compiler in "Maschinensprache" übersetzt.

Die Prozeduren sind in Textform für die Lauffähigkeit des Programmes nicht notwendig.

Sind sie aber in der Datenbank vorhanden, so können Programmänderungen direkt im Echtsystem geändert werden, ohne erst ein Update zu erstellen.

Da bei der Programmierung oftmals auf Bibliotheken zurückgegriffen wird, kann der Compiler Änderungen nur dann kompilieren, wenn alle inkludierten (verknüpften) Prozeduren in Textform vorliegen.

**WICHTIG!**

*Werden Prozeduren im laufenden Betrieb geändert und kompiliert, so müssen  sich die Anwender, die in dem geänderten Bereich arbeiten,am System neu anmelden. Der Programmcode wird in den lokalen Speicher geladen und bei Bedarf nachgeladen.*

*Durch das Kompilieren ändern sich der Code und die Speicheradressen und beim Nachladen wird dann auf eine falsche Adresse zugegriffen. Dadurch erfolgt eine Laufzeit-Fehlermeldung.*

*Bei Laufzeitfehlern wird die entsprechende Prozedur bis zu einem Neustart immer wieder abgebrochen ohne erneut einen Fehler zu melden.*

*Im Laufzeitfehler werden folgende Angaben gemacht:*

 *Aktuelles Ereignis*

*Prozedur*

*Zeile*

*Meldungstext*

*Der Programmierer kann so direkt in der Prozedur  die Zeile heraussuchen und den Fehler lokalisieren.*

***3.Programmierung / Prozeduren / Bibliotheken***

Die Prozeduren sind wie folgt eingeteilt:

**3.1.1 Bibliotheken (Der Name beginnt immer mit: sysB.)**

    Bibliotheken enthalten

1. allgemeine Funktionen, die nicht für ein bestimmtes Modul ausgerichtet sind. Sie können auch für andere Module genutzt werden.
2. Globale Variablen die entweder System- oder Modulweit gelten
3. Funktionsaufrufe

**Folgende Bibliotheken sind derzeit in DB-System vorhanden:**

Jede Bibliothek enthält am Anfang eine Beschreibung der Funktionen.

sysB.Bin                    - Funktionen für binäre Operationen (z. B. Grafiken importieren)

sysB.Check               - Prüfroutinen

sysB.Fibu                  -  Funktionen zur Fakturierung und Preisfindung

sysB.Funktion            - Hauptbibliothek , die in allen Modulen benötigt wird. Dort befinden sich

die Definitionen für Funktionsaufrufe. Hier werden auch die Bibliotheken: sysB.Msg, sysB.VarGlobal, sysB.F\_Dateien, sysB.F\_StatusWerte  inkludiert

sysB.F\_Dateien         - Enthält die Definitionen für Dateinummern, so dass diese mit einem

 "sprechenden" Namen angesprochen werden können

sysB.F\_Statuswerte   - Enthält die Definitionen für Numerische Werte, so dass diese mit

 "sprechenden" Namen angesprochen werden können

sysB.Hilfe                  - Funktionen zu Update und Onlinehilfe

sysB.Info                   - Funktionen für Menüpunkt "?"  und weitere Informationen

sysB.InfoDelRec        - Funktion, um gelöschte Datensätze zu finden

sysB.logBuch             - Funktionen für das Logbuch- Datensatz Restore etc.

sysB.MSG                - Funktionen zu Ausgabe von Meldungen und Definition der

 Meldungsnummer

SysB.Printer              - Funktionen zur Druckersteuerung und Ausgabe der Druckelemente

sysB.RecLst              - Definitionen und Funktionen für alle RecList (Übersichtslisten)

sysB.RecToObj        - Funktionen zum Füllen von Objekten und Anzeigefeldern mit

 Datensatzinformationen

sysB.Statistik            - Funktionen für alle Datensatzoperationen für die tmporäre Datei (Auswertungen

 Statistiken etc.)

sysB.Tools               - Verschiedene Funktionen zu Konvertierungen, Datensicherung etc.

sysB.Txt                   - Funktionen zur Textverarbeitung

***3.Programmierung/ Prozeduren / Bibliotheken***

sysB.VarGlobal        - Enthält die Variablen die systemweit oder  MDI-weit gelten.

 sysB.Windows        - Funktionen für Excel Aufruf und Datenübergabe

***3.Programmierung/ Prozeduren / Systemprozeduren***

**3.1.2 Systemprozeduren (Der Name beginnt immer mit: sysP.)**

Systemprozeduren enthalten die Steuerung für Globale Objekte, wie z.B. Initialisierungen für Module, Aufbau der Übersichtlisten, Druckausgabe etc.

Sie werden von vielen Modulen verwendet.

sysP.AutoTermin    - Erzeugung von Terminen

sysp.DbAkt            - Kontrollprozedur für Datenaktualisierungen

sysP.DbStart          - Funktionen für Datenaktualisierungen

sysP.DBStartJobs   - Protokollierung der Datenaktualisierung

sysP.DBSystem      - Prüfung der Lizenz

sysP.DelHistorie     - Historien-Datensätze löschen (mit Del-Kennzeichen)

sysP.Externe Datei  - Funktionen für Externe Dateiverarbeitung

sysp.InitMdi            - Initialisierungsprozedur bei Start der MDI-Frames

sysP.InitMenu         - Initialisierungsprozedur für alle Menüs, Toolbar und Kontextmenüs

sysP.Konfig            - Schriftwahl

sysP.Labeldrucken  - Funktionen für das Bedrucken von Label (Geschäftspartner)

sysp.LabelImport     - Import von Daten für das Bedrucken von Label

sysP.Login              - Funktion für das Anmelden ans System

sysP.Msg                - Funktionen für die Ausgabe von Meldungen

sysP.NachUpdate  - Funktionen die nach dem Update ausgeführt werden

sysP.NewAPP        - Funktion um einen neuen Mandanten anzulegen

sysP.ObjKonfig      - Funktionen zur MDI-Objektkonfiguration

sysp.Pos                 - Positionsverarbeitung (Auftrag, Faktura, Vertrag, Angebote, Aktionen,

 Begleitpapiere, Wiegescheine)

sysp.PosVk            - Positionsverarbeitung für verknüpfte Daten

sysP.PrintLabel       - Steuerung für die Ausgabe der Label bei Geschäftspartner

sysP.PrintListe        - Steuerung zur Ausgabe aller Standardlisten

sysP.PrintRegister   - Ausgabe Statistik Register für EANV

sysp.PrintStatistik   - Funktionen für die Ausgabe von Statistiken

sysP.RecIo            - Funktionen für Dateioperationen

sysP.Reclst             - Enthält alle Funktionen zur Steuerung der Übersichtslisten im Modul.

sysP.RecLstLuchs  - Enthält alle Funktionen zur Steuerung der Übersichtslisten im Modul. für

 das Modulpaket "Luchs"

sysP.RecLstZL      - Enthält alle Funktionen zur Steuerung von Auswahllisten

sysP.Spezial          - Modulsteuerung um Datensätze physikalisch zu löschen

sysP.System          - Leeren einer Systemdatei

***3.Programmierung/ Prozeduren / Systemprozeduren***

sysP.Transfer        - Funktionen für Import und Export

sysP.Winrar          - Packen von Verzeichnissen

syslp.Pos              - Positionsverarbeitung für Lieferscheine

sysPP.Pos            - Positionsverarbeitung für Aufträge LK

***3.Programmierung/ Prozeduren / Variable***

**3.1.3 Variable**

Variable sind temporäre Speicherorte, für die kein Datenbankfeld erforderlich ist. Sie dienen zum Zwischenspeichern von Werten für den schnellen Zugriff.

Der Geltungsbereich ist wie folgt geregelt:

Globale Variable                              - Können systemweit für den angemeldeten Benutzer verwendet werden.

Globale Variable für das Modul - Der Geltungsbereich ist vom Start bis zum Beenden des

 Moduls

Lokale Variable                              - Diese gelten nur bei der Ausführung einer Funktion

Die globalen Variablen sind in der Bibliothek

SysB.VarGlobal

hinterlegt.

Der Abschnitt heißt GLOBAL SYSTEM.

Diese werden beim Start der Applikation initialisiert und behalten bis zum Abmelden des Benutzers ihre Gültigkeit.

Die globalen Variablen für das Modul befinden sich ebenfalls in der Bibliothek

SysB.VarGlobal.

Der Abschnitt heißt: GLOBAL MDI.

Diese werden in den Eigenschaften des MDI-Frames eingetragen und von Conzept16 automatisch initialisiert.

Es befinden sich hier die Variablen, die für jedes Modul benötigt werden.

Modulspezifische Variablen werden ebenfalls in den Eigenschaften der MDI-FRAMES eingetragen. Sie befinden sich entweder in der Bibliothek SysB.VarGlobal oder in der modulspezifischen Definition

(Siehe Abschnitt Programmierung Module).

Lokale Variable werden in den Prozeduren in jeder Funktion deklariert und bei Aufruf der Funktion initialisiert.

Sind sind dort jeweils  im Abschnitt LOCAL{} definiert.

***3.Programmierung/ Prozeduren / Struktur***

3.1.4 Struktur

Derzeit sind ca. 1100 Prozeduren in DB-System enthalten. Die Vergabe der Prozedurbezeichnung ist für das Auffinden von Funktionen sehr wichtig und wurde wie folgt geregelt:

**Bibliotheken**  beginnen immer mit  **sysB.**  und erhalten dann eine Bezeichnung für ihren Inhalt (siehe Beschreibung Bibliotheken).  Sie enthalten Funktionen, die in verschieden Modulen genutzt werden.

**Systemprozeduren** beginnen immer mit sysP. und erhalten dann eine Bezeichnung für ihren Inhalt (siehe Beschreibung Systemprozeduren). Sie steuern Bereiche, die in allen Modulen gleich sind. (z.B. Datensatzoperationen (sysP.RecIo)

Die Anlage von Systemprozeduren und Bibliotheken haben den Vorteil, dass bei Änderungen der Funktionalität, diese nur an einer Funktion vornehmen muss. Zu beachten ist hierbei, das bei Programmierfehler auch alle Module

davon betroffen sind.

Modulspezifische Prozeduren  beginnen immer mit dem Dateikürzel. Die Steuerung des Stammdatenmoduls Geschäftspartner beginnen entsprechend der Datei (paD.Partner)   mit paP. und der Bezeichnung des Inhaltes.

Durch diese Regelung können bei Problemen oder Erweiterungen  schnell die benötigten Prozeduren gefunden werden.

***3.Programmierun g / Module***

**3.2 Module**

Als Module werden Programmbereiche in DB-System bezeichnet.

z.B. Stammdaten-Modul Geschäftspartner für die Stammdatenpflege der Geschäftspartner

 oder Modul Auftrag für die Auftragsbearbeitung.

Alle Module sind nach meinem entwickelten System gleich aufgebaut.

**3.2.1 Struktur**

Jedes Modul besteht aus:

1. Hauptfenster - MDI.Main
2. 5 Prozeduren - xxP.Main, xxP.InitMdi,xxP.Def,xxP.RecIo,xxP.Print

Jedes Hauptfenster besteht aus folgenden Objekten:

* + - 1. Reclist - RLM\_MAIN
			2. Groupbox - GB\_SEARCH
			3. Toolbar - Toolbar
			4. NoteBook - NB\_DATEN
			5. Statusanzeige - Windowbar6

Mit dieser einheitlichen Strukturierung ist es möglich, viele Abläufe mit globalen Funktionen zu steuern.

Der Aufbau der Prozeduren und Objekte wird in den nachfolgenden Kapiteln beschrieben.

***3.Programmierun g / Module / Prozeduren***

**3.2.2 Prozeduren**

Für die Dialogmaske wird ein MDI-Frame verwendet. MDI-Frames können mehrmals aufgerufen werden und besitzen einen eigenen Speicherbereich für Variable und Puffer für Datensätze.

Für jedes Modul  gibt es mindestens die 5 Standardprozeduren:

MAIN        -  Steuerung der Maske und Eingaben

INIT          -  Initialisierung der Maske, Steuerung der Berechtigungen, Heranziehen der

 verknüpften Datensätze

DEF          -  Definitionen für Funktionsaufrufe und spezielle Variable

PRINT      -  Ausgabesteuerung für den Ausdruck oder Anzeige von Listen

RecIO     -  Steuerung der Datenbankoperationen, speichern, löschen etc.

Für das Stammdatenmodul Geschäftspartner heißen die Prozeduren entsprechend:

paP.Main

paP.Init

paP.Def

paP.Print

paP.RecIo

Die Prozedurkennung für das Modul wird im Mdi-Frame in der Eigenschaft Custom eingetragen (paP.)

Bei der Initialisierung in der Prozedur SysP.InitMDI  wird diese in die Globale Variable (Global MDI) Prozedur übertragen und steht so jeder Funktion während der Bearbeitung des Moduls zur Verfügung.

Eintrag in SysP.InitMDi:EvtInit :

Prozedur # aEvt:Obj->wpCustom; // ProzedurKürzel



Durch diesen Eintrag wissen externe Funktionen welche Prozedur aufgerufen werden muss.

***3.Programmierun g / Module / Prozeduren***

So wird z.B. die Toolbar zentral für alle Module gesteuert.

Beispiel: Auswahlbutton  wird in der Toolbar geklickt.

Funktion # Prozedur+'Main:';              (Inhalt:  Prozedur = paP.     Funktion = paP.Main:)
Funktion # Funktion+'Auswahl';          (Inhalt:  Funktion = paP.Main:Auswahl)
Call(Funktion);                                   Hiermit wird die Funktion  in paP.Main ausgeführt. In dem

 Fall die Routine für die Anzeige von Auswahllisten)

In den Eigenschaften des MDI-Frames  werden die globalen Variablen für das Modul hinterlegt.

GLOBAL MDI ist für jedes Moduls notwendig und muss dort zugeordnet werden.

In der Eigenschaft DBVar wird immer die Anzahl der zugeordneten Variabledeklarationen angezeigt.

Abb. Globale Variable für Modul



***3.Programmierung/ Module / Prozeduren***

**Für ein MDI-Frame werden folgende Systemprozeduren eingesetzt:**

Für die Anzeige des Frames                                                  - sysP.InitMdi

- hier werden die Fenstergröße, Beschriftungen, Anzeigeoptionen  gesteuert

Für die Anzeige der Übersichtsliste

(Bezeichnung : RLM  = Rec-List-Main )   - sysP.RecLst

 - Steuerung der Anzeige, Spaltenansicht, Sortierung, etc.

Toolbar                                                                                - sysP.initMenu

-  Anzeigensteuerung, Funktionstasten, Tipptexte

MAIN - Prozedur

In der Main-Prozedur befindet sich die Hauptsteuerung für die Moduleigenschaften und - ereignisse.

In Abhängigkeit der Objekt-Ereignisse sind die Funktionen in der Prozedur eingerichtet:

Folgende Ereignisse sind für das Mdi.Frame definiert:

EvtInit                   -  Die angegebene Prozedur wird ausgeführt, wenn das Objekt geladen wurde.

EvtTerm                -  Die angegebene Prozedur wird ausgeführt, wenn das Objekt entladen

 wurde. Dieser Eintrag erfolgt durch die Initialisierung in der sysP.initMDI.

 Die EvtTerm-Funktion ist dort ebenfalls definiert. Sie ist für alle Module

 gleich.

EvtClose               -  Die angegebene Prozedur wird ausgeführt, wenn der Benutzer das Fenster

 schließen möchte. Dieser Eintrag erfolgt durch die Initialisierung in der

 sysP.initMDI.

EvtMdiActivate     -  Die angegebene Prozedur wird ausgeführt, wenn das Frame den Focus

 bekommt. Dieser Eintrag erfolgt durch die Initialisierung in der

 sysP.initMDI. Die Funktion ist dort ebenfalls definiert. Sie ist für alle

 Module gleich.

***3.Programmierung/ Module / Prozeduren***

Für die verschiedenen Objekte innerhalb eines Mdi-Frame sind folgende Funktionen definiert:

Objekt Ereignis Funktion in MAIN

Button                      EvtClicked Button Steuerung der Aktionen, wenn

 auf einen Button mit der Maus

geklickt wird

Eingabefelder            EvtFocusInit EvtFocusInit         Initialisierungen wenn ein

 Eingabefeld den Focus erhält

                                  EvtFocusTerm  FocusTerm           Steuerung der Aktionen beim

 Verlassen  des Objektes

                                 EvtChanged         EvtChanged          Steuerung bei Veränderungen im

 Objekt

***3.Programmierung/ Module / Prozeduren***

Modulsteuerung

Es erfolgt hier die sehr grobe Beschreibung der Abläufe vom Start - bis Schließen eines Moduls und soll so die Beziehungen zu den Standardprozeduren verdeutlichen:

Aktion durch Benutzer Prozedur Funktion

Start des Moduls durch

Auswahl im Menü sysB.Tools Ermittlung des Frames, Einstellung der

 benutzerdefinierten Framegröße

xxP.Main:EvtInit  Vorbereitungen zum Modul, Filter für

 die Auswahllisten setzten

Aufruf von xxP.InitMdi

xxP.InitMDI:InitRechte  Berechtigung des Anwenders einlesen

 und ggf. Objekte ausblenden

xxP.InitMDI:InitMdi Objekte auf Basis des Bearbeitungs-modus und

 Berechtigung einstellen

Datensatz wird in der

Auswahlliste aktiviert xxP.InitMdi:Refresh Auf Basis des ausgewählten Datensatzes werden Verknüpfungen gelesen und die Maske

 aktualisiert

 Auf die Toolbar wird

Neuanlage geklickt sysP.InitMenu:EvtMenuCommand Aufruf der Funktion xxP.Recio:NEW

                                           xxP.RecIo:New Leeren der Datensatzpuffer, Log-Daten setzten,

Aufruf von xxP.InitMdi

xxP.InitMdi:InitMdi Objekte auf Basis des Bearbeitungsmodus und der Berechtigung einstellen

Eine Eingabe erfolgt xxP.Main:EvtInit Das Objekt, das den Focus erhält wird initialisiert,

 eventuelle Auswahl aktivieren, Wertebereiche setzten etc.

                                          xxP.Main:EvtChanged Jede Änderung des Objektes wird geprüft

Feld wird verlassen xxP.Main:FocusTerm Prüfung der Eingabe und entsprechende Aktionen

Toolbar speichern

Anklicken sysP.InitMenu:EvtMenuCommand Aufruf der Funktion xxP.Recio:Save

                                          xxP.RecIo:Save Prüfen der Eingabe, Speichern des Datensatzes.

Aufruf der Funktion xxP.initMdi

xxP.InitMDI:InitMdi Objekte auf Basis des Bearbeitungsmodus und Berechtigung einstellen und mit dem neuen Datensatz füllen

***3.Programmierung/ Module / Objekte***

**3.2.3 Objekte**

Ein Modul basiert auf einem MDI-Frame und erhält je nach Umfang eine Vielzahl an Objekten.

 Zum Standard gehören:

Beispiel: Modul Ansprechpartner



 Diese Objekte sind in jedem Modul vorhanden und erhalten denselben Bezeichner.

Objekt Bezeichner Prozeduren/Funktionen Beschreibung

1. Toolbar          Toolbar                sysp.initMenu:InitToolbar               Initialisierung der Toolbar-Button

(Aktivieren, Tipptext, F-Key, Beschriftung)

sysp.initMenu:InitTBModus Initialisierung der Toolbar-Button nach

 Bearbeitungsmodus,(Anzeige, Erfassung,

 Korrektur)

sysp.initMenu:EvtMenuCommand Funktionsaufruf nach Aktivierung eines

Toolbarbutton

2. WindowBar Windowbar6  Enthält die Anzeigen für Filter und Modus

3. RecList RLM\_MAIN RLM\_Main ist der Definitionsname. Diesem

 wird entsprechend dem Modul die Bezeichnung

zugewiesen (RLM\_Partner für die RecList der

Geschäftspartner)

sysP.RecLst:EvtInit Aufbau der RecList( Spaltenbreite,- bezeichnung, - anzeige, Sortierung,

 Formatierungen)

***3.Programmierung/ Module / Objekte***

sysP.RecLst:EvtTerm Beim Schließen des Moduls werden eventuelle

 Änderungen in der Ansicht der Recliste

 anwenderbezogen gespeichert.

sysP.Reclst:EvtMouseItem Auswertung der Mausaktivität

                                                         sysP.recLst:EvtMenuinitPopup Initialisierung des Context-Menüs,

 Menüoptionen

sysP.RecLst:

EvtMenuCtxtCommand Auswertung der Menüaktionen

sysP.RecLst:EvtLstSelect Aktionen wenn ein Datensatz aktiviert wird (zB.

 Maskenrefesh)

4. Groupbox GB\_SEARCH sysp.InitMenu:

EvtMenuInitPopup\_MDI Initialisierung des Context-Menüs für das

MDI-Frame, Menüoptionen

sysp.InitMenu:

EvtMenuCommand\_MDI     Auswertung der Menüaktionen

 5. Notebook NB\_Daten sysp.InitMenu:

EvtMenuInitPopup\_MDI      Initialisierung des Context-Menüs für das MDI-

 Frame, Menüoptionen

sysp.InitMenu:

EvtMenuCommand\_MDI     Auswertung der Menüaktionen

sysp.initmdi:EvtPageSelect  Auswertung der gewählten Seite des Notebooks

6. NotebookPage NP\_Daten xxP.Main:EvtMouse Auswertung der Mausaktionen

sysp.InitMenu:

EvtMenuInitPopup\_MDI       Initialisierung des Context-Menüs für das MDI-

 Frame, Menüoptionen

sysp.InitMenu:

EvtMenuCommand\_MDI     Auswertung der Menüaktionen

 7. MDI\_Frame MDI\_MAIN MDI\_Main ist der Definitionsname. Diesem wird

 entsprechend dem Modul die Bezeichnung

 zugewiesen (MDI\_Partner für das Frame der

 Geschäftspartner)

sysP.InitMdi:EvtInit Initialisierung von Online-Hilfe,

Systemvariable und der EvtInit-Funktion der

Main-Prozedur

sysP.InitMdi:EvtClose Speichern von Datenveränderungen,

 Einstellungen, Schließen von eventuellen für das

 Modul gestarteten automatischen Prozessen

***3.Programmierung/ Module / Objekte***

sysp.InitMdi:EvtMdiActivate Auswertung bei erhalt des Fokus

sysP.InitMenu:

EvtMenuCommand Auswertung der Menüaktionen

Je nach Umfang des Moduls wird der Standard entsprechend erweitert.

Verknüpfte Daten werden in weiteren RecListen angezeigt. Diese tragen dann den Bezeichner RLS.xxxxx

Alle anderen Objekte behalten Ihren Bezeichner und erhalten nur einen sprechenden Namen. Der Bezeichner ist wichtig, da die globale Initialisierung diese auswertet.

***Programmierung / Auswertung/Statistik / Bibliotheken***

***3.3.1 Bibliotheken***

Für die Ausgabe von Auswertungen und zur Steuerung von Drucker sind folgende Bibliotheken im Einsatz:

sysB.Printer

Diese Bibliothek enthält alle Funktionen  zur Druckerinitialisierung:

  1. Standarddrucker - Vorgegebener Drucker wird geprüft und ggf. eine Auswahl angezeigt
  2. Seitenumbrüche          - Prüfungen und Erzwingen von Seitenumbrüchen
  3. Logdaten                - Eintragung von Druckkennzeichen, Datum Anzahl
  4. Ausgabe initialisieren - Ausgabedevice über den Druckernamen ermitteln
  5. Datenaufbereitung       - Verknüpfungen zu Druckausgaben ermitteln
  6. Formularermittlung      - Ermittelt mandantenspezifische Formulare
  7. Druckerschacht           - Schachtauswahl von Drucker ermitteln
  8. Firmenlogo                  - Ausgabe von Logos

Die nächste Bibliothek enthält  alle Funktionen zum Lesen und Speichern von Datensätzen in der Datei stD.TempStatistik (985).

SysB.Statistik

ST\_Clr : sysB.Statistik:StClrFile();                                   - Datei für den ST-COUNT  leeren
ST\_ClrFz : sysB.Statistik:StClrFile(0,352); - Datei für den ST-COUNT  und zus. Kriterien leeren
ST\_ClrPosH(a) : sysB.Statistik:StClrFile(a);                    - Datei für den ST-COUNT  und zus. Kriterien leeren
ST\_RecTest : sysB.Statistik:StRecTest();                         - Prüft, ob ein Datensatz vorhanden ist
ST\_Insert : sysB.Statistik:StInsert();                                 - Datensatz einfügen
ST\_ReadEqL : sysB.Statistik:StReadEqL();                     - Datensatz lesen und sperren
ST\_ReadEq : sysB.Statistik:StReadEq();                           - Datensatz lesen
ST\_ReadFirst : sysB.Statistik:StReadFirst();                     - ersten Datensatz lesen
ST\_ReadFirstPosH(a) : sysB.Statistik:StReadFirst(0,a);  - Ersten Datensatz mit zusätzlichen Optionen lesen
ST\_ReadFirstKey(a) : sysB.Statistik:StReadFirst(a);       - Ersten Datensatz mit zusätzlichen Optionen lesen
ST\_ReadNext : sysB.Statistik:StReadNext();                   - Nächsten Datensatz lesen
ST\_ReadNextPosH(a) : sysB.Statistik:StReadNext(0,a);  - Nächsten Datensatz mit zusätzlichen Optionen lesen
ST\_ReadNextKey(a) : sysB.Statistik:StReadNext(a);       - Nächsten Datensatz mit zusätzlichen Optionen lesen

***Programmierung / Auswertung/Statistik / Bibliotheken***

ST\_ReadPrev : sysB.Statistik:StReadPrev(); - Vorherigen Datensatz lesen
ST\_SELRun : sysB.Statistik:SEL\_RUN(); - Selektion kopieren und starten
ST\_SELClose : sysB.Statistik:SEL\_CLOSE(); - Selektion löschen
ST\_ClrCaption : sysB.Statistik:ClrCaption(); - In Objekte mit dem Namen Text wird der Inhalt geleert ST\_SetDivider : sysB.Statistik:Divider(True); - Linie werden eingebelndet
ST\_SetFont(a) : sysB.Statistik:SetFont(a);                  - Schriftart setzen, Übergabe Printform
ST\_SucheON : sysB.Statistik:St\_Suche(TRUE);        - Anzeige für Suche einschalten
ST\_SucheOFF : sysB.Statistik:St\_Suche(FALSE);    - Anzeige für Suche ausschalten
ST\_SetStatus(a) : sysB.Statistik:St\_STATUS(a);        - Anzeige für Suche mit Text füllen

Für Statistiken wurden 2 Dateien angelegt. Diese dienen zum Einen zur Ausgabe von Überschriften und Zeilen und zum Anderen zur Zwischenspeicherung der für die Auswertung gefundenen Daten.

Da mehrere Benutzer diese Datei  gleichzeitig nutzen können, wird die Eindeutigkeit der Daten für den Benutzer durch die Variable ST\_COUNT  hergestellt. Diese eindeutige Nummer wird bei Start eines Moduls generiert

Die Variable wird bei den Funktionen automatisch berücksichtigt.

 Die Funktionen beginnen immer mit     ST\_

(z.B. ST\_CLR   - Löschen der Datei stD.TempStatistik für den aktuellen ST\_COUNT.

***Programmierung / Auswertung/Statistik / Variable-Stuktur***

***3.3.2 Variable***

Für die Ausgabe von Auswertungen / Statistiken sind folgende globalen Variablen-Segmente definiert:

Global gPrint      - Variablen für die Druckaufbereitung

Global Statistik  - Variablen für Statistiken

***3.3.3 Struktur***

**3.3.3.1  Elemente zur Ausgabe**

Zur Ausgabe von Druckanzeigen werden

**Druckdokumente**   (Bezeichner : prD.xxxxxx)

**Druckformate**        (Bezeichner : prF.xxxxxx)

verwendet.

**Druckdokumente** sind Seitenorientiert.

Das bedeutet, dass ein Dokument immer eine vollständige Ausgabeseite ist.DieDaten für die gesamte Seite müssen vor Ausgabe bereitgestellt werden.

**Druckformate** sind frei definierbar. Sie können Teile einer Ausgabeseite oder auch eine ganze Ausgabeseite darstellen.

Für die Ausgabe von Statistiken werden Druckformate verwendet, da diese zeilenorientiert zu einer gesamten Seite zusammengefügt werden können.

Standardformate bestehen aus 5 Elementen:

**Listenkopf**  (Ausgabe nur einmal am Anfang der Liste)   prf\_xxxxxxLH

**Seitenkopf** (Ausgabe bei jedem Seitenwechsel)             prf\_xxxxxxPH

**Zeile**          (Ausgabe für jede Zeile)                               prf\_xxxxxx

**Seitenfuß**   (Ausgabe vor jedem Seitenwechsel)             prf\_xxxxxxPF

**Listenfuß**  (Ausgabe nur einmal am Ende der Liste)        prf\_xxxxxxLF

Bei Gruppenwechsel

1. Gruppenwechsel                                                       prf\_xxxxxxGW

2. Gruppenwechsel                                                       prf\_xxxxxxGW2

***Programmierung / Auswertung/Statistik / Prozeduren***

In den Prozeduren zur Ausgabe der Druckformate (Sysp.PrintListe, sysP.PrintStatistik und sysB.Printer) werden diese Bezeichner automatisch berücksichtigt.

**3.3.3.2  Prozeduren**

Beinhaltet die Toolbar die Button Seitenansicht und Liste, so muss dieses Modul auch eine Prozedur mit dem Namen  xxP.Print  besitzen. Diese wird automatisch durch die Toolbar

aufgerufen. Für das Stammdatenmodul Geschäftspartner heißt die Prozedur z.B. paP.Print.

Diese Prozeduren haben mindesten 2 Funktionen

- SUB PRINT    -   für die Seitenansicht eines einzelne  Datensatzes

- SUB LIST      -   für  die Ausgabe einer Liste als Seitenvorschau bei Stammdatenprogrammen und bei

Vorgängen( z.B. Rechnungen, Aufträge etc.) die Ausgabe des Vorgangs auf den eingestellten

Drucker.

Auswertungen und Statistiken werden mit den folgenden Prozeduren ausgegeben:

(Kursive dargestellte Prozeduren sind für L&K nicht aktiv)

stP.AU.Main            -  Statistiken über Aufträge

stP.AVAU.Main      -   Statistiken für die Arbeitsvorbereitung

stP.BDE.Main         -   Statistiken für die Betriebsdatenerfassung

stP.BP.Main            -   Statistiken für die Begleitpapiere

stP.Fa.Main            -   Statistiken für die Fakturierung

stP.GewAbfVo       -   Statistiken für die Gewerbeabfallverordung

stP.LU.Main            -   Statistiken für die Lagerumbuchungen

stP.VT.Main            -   Statistiken für den Vertrieb

stP.LV.Main            -   Statistiken für die Lagerverwaltung

stP.WS.Main           -   Statistiken über Wiegescheine

***Programmierung / Auswertung/Statistik / Prozeduren***

Der Aufbau der Prozeduren ist einheitlich mit folgenden Funktionen programmiert::

**Für die Maskensteuerung:**

SUB EvtInit             - Initialisierung des MDI.Frame für die Ausertungen

SUB EvtChanged    - Kontrolle der Eingaben in der Maske

SUB EvtFocusInit    - Aktivierung der Toolbar

SUB Auswahl          - Steuerung der Auswahl

SUB Button            - Steuerung der Buttons und der Listenauswahl

SUB FocusTerm    - Kontrolle nach Verlassen eines Eingabefeldes

**Für die Auswertung:**

SUB FillTmp                          -  Auf Basis der getroffenen Listenauswahl werden hier die Daten gesucht und für die

 Statistik aufbereitet

SUB FillSort                         -   Füllt die Felder für die Sortierung

**Für die Ausgabe:**

SUB AUSGABE\_PRINT  -  Auf Basis der getroffenen Listenauswahl werden hier die die Bezeichner für die

 Listenbezeichnung und Prozeduren für Listenkopf, Seitenkopf, Listenzeile,

 Seitenfuß, Listenfuß, Gruppenwechsel 1 und 2 definiert.

                                                Des Weiteren werden das Ausgabeformat (Querdruck) und die Ausgabeeinheit

 (Drucker, PDF Seitenansicht) der Prozedur sysP.PrintStatistik übergeben

SUB Kopf                          - Initialisierung von Listenkopf

SUB Seitenkopf                 -  Initialisierung für den Seitenkopf

SUB Zeile                          - Initialisierung der Zeilen

SUB FUSS - Initialisierung von Listenfuss/Seiitenfuss

SUB GW                           -  Initialisierung für den 1. Gruppenwechsel

SUB GW2                         -  Initialisierung für den 2. Gruppenwechsel

**sysP.PrintStatistik**        -  Diese Prozedur steuert alle Statistiken. Durch Übergabe der Parameter aus der

 Funktion AUSGABE\_PRINT werden die Daten ausgegeben.

***Programmierung / Auswertung/Statistik / Prozeduren***

Der Ablauf einer Statistik wird hier beschrieben:

* MDI.Frame (Maske) für die Auswahl der Statistik-Kriterien wird angezeigt.
* Eine Auswahl wird vom Anwender getroffen
* In der Funktion FillTMP werden die gewünschten Daten ermittelt, sortiert und in der Datei stD.Statistik temporär gespeichert.
* Anschließend erfolgt in der Funktion AUSGABE\_PRINT die Übergabe an sySP.PrintStatistik.
* Dort wird die Ausgabe Initialisiert und die Seite vorbereitet
* Der Listenkopf wird geladen
* Wenn eine Prozedur (*gProz\_LH*) übergeben wurde, wird diese aufgerufen.
* Der Listenkopf wird in die Seite eingefügt.
* Eventuelle Gruppenwechsel (*gProz\_GW*) werden geprüft und entsprechend der Seite hinzugefügt
* Das Element für die Zeilenausgabe wird geladen.
* Die Prozedur(*gProz\_Data*) für die Zeile wird aufgerufen.
* Dort werden die Daten gelesen, evtl. aufbereitet und der Prozedur  sysP.PrintStatistik über die Variable PRN\_DATA mitgeteilt, ob sie das Element hinzufügen  kann.
* Das Element wird der Seite hinzugefügt.
* Nach dem Hinzufügen von Druckformaten wird immer der restliche Platz auf der Seite überprüft(sysB.Printer)  und ggbfs. Ein Seitenwechsel eingefügt.
* Die Funktion für den Seitenwechsel(*gProz\_PF*) überprüft das Vorhandensein von Druckformaten und Prozeduren für den folgenden Seitenfuß und Seitenkopf.

Dieser grob beschriebene Ablauf erfolgt bei jeder Zeilenausgabe bis zum letzten Datensatz.