SoftRedoc

System Redocumentation Tool

Version: 2.1
Date: Dec. 2008

Harry M. Sneed

Harry.Sneed@anecon.com

Software Design und Beratung G.m.b.H.
Alserstraße 4 / Hof 1
A-1090 Wien

1. Purpose of the SoftRedoc Tool

SoftRedoc is a tool to aide in the post documentation of existing software systems. As such it can be considered to be a reverse engineering tool. Its purpose is to generate documents from source code as well as to extract information for creating a software repository. The target person is in both cases the maintenance engineer. Maintenance productivity is highly dependent on the knowledge, the maintenance engineer has about the system being maintained. He needs to know what entities the system is composed of and what kinds of relationships they have to each other. In addition, he should know in what order they are executed and how they depend on one another. A documentation tool should provide this information in both textual and graphical form. SoftRedoc provides the textual documents and the information required to produce the graphics. As such it has a twofold purpose

- generating text documents and
- extracting information for a software repository

(*see figure 1: Twofold Purpose of SoftRedoc*)

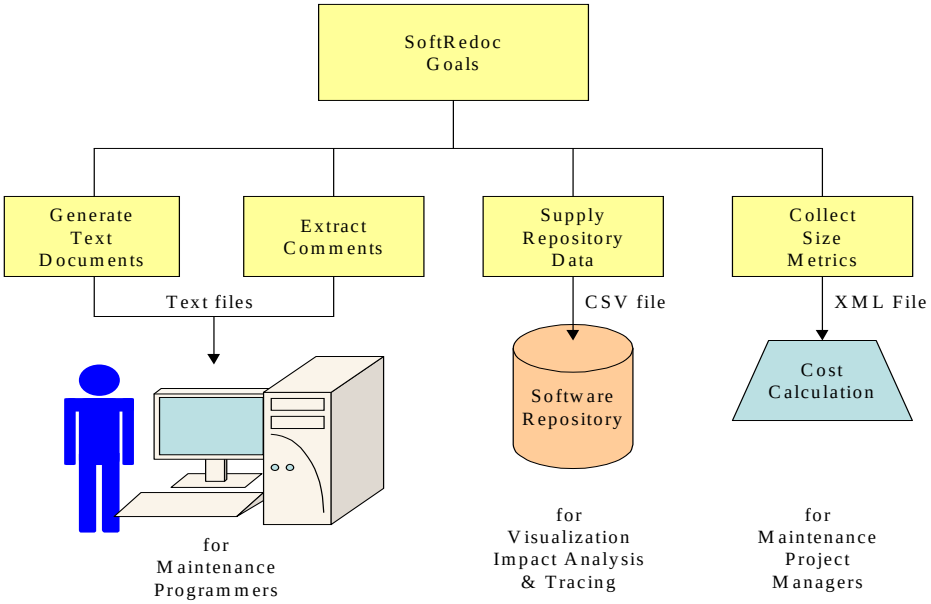


Figure 1: Goals of SoftRedoc

1.1. Generating Text Documents

The purpose of the text documents is to provide the maintenance engineer with detailed information about individual components. Each document presents another view of the component source code. Some documents outline the source structure. Others document the source contents, i.e. the data and procedural elements. Still others depict the relationship between the elements. What view, or views, a maintenance engineer needs, depends on the task at hand. He may be searching the source of an error or looking for the best place to insert a change. Since there are so many different maintenance tasks to be performed, tasks such as changing data types, inserting new data items, changing logical conditions and inserting additional statements, several complementary views are required.

The views presented by SoftRedoc depend on the software artifact type and the programming methodology. Data structures such as interface definitions, user interface panels and database tables require another type of view than code modules. The view of code module depends on how the module is implemented. Object oriented modules require other views then procedural modules.

The objective of ***SoftRedoc*** is to provide an appropriate view of each artifact based on type and construction methodology. The views are intended to make the source code more transparent to the maintenance engineer, so that he can perform his job quicker, better and safer.

1.2. Extracting Information for a Software Repository

The second objective of ***SoftRedoc*** is to populate a software repository. With the repository it is then possible to navigate through a graphical representation of the system architecture, i.e. to visualize the structure of the software, in particular to perform impact analysis and to assure the consistency and completeness of the system. Thus, in this respect SoftRedoc is really performing an indirect service. It acts as a data feed for the tool SoftRepo so it can accomplish its objectives. SoftRedoc is a prerequisite for using SoftRepo.

To this end, SoftRedoc must create interface tables containing all of the relevant entities and relationships contained in the source code. Such dynamic relationships as method or procedure calls, data accesses and control flow directions, and such static relationships as what modules contain what classes, what classes have which data attributes and which data files or tables are used by which procedures. This information is provided in the form of a standard CSV type table which can be further processed by any other documentation tool. A schema is provided to explain the structure and contents of this interface table.

2. Functions of the SoftRedoc Tool

In line with its purpose, SoftRedoc performs two basic functions

- it documents the source code and
- it creates an export file

2.1. Source Code Documentation

The source code documentation function includes a separate documenting function for each source code type.

- Programs
- Database schemas
- user interfaces
- system interfaces and
- JCL or WFL procedures

(*see Figure 2: Redocumentation Functions*)

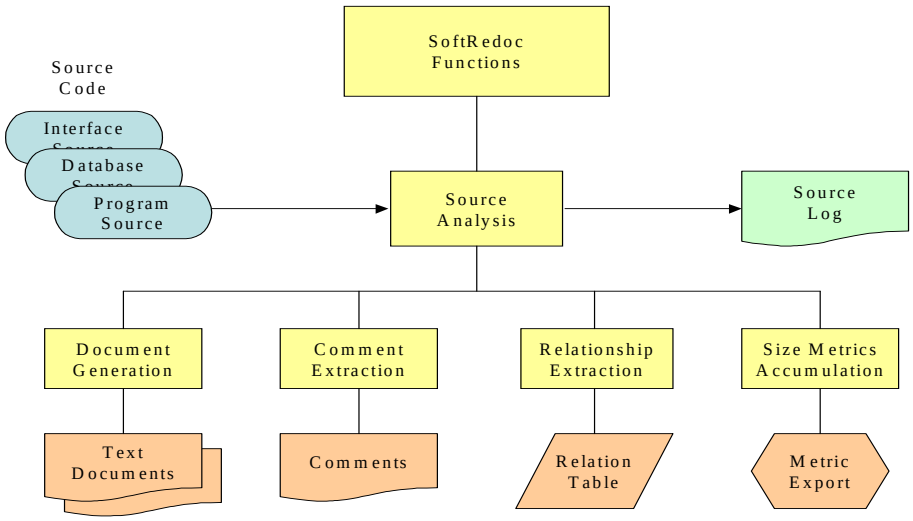


Figure 2: Functions of SoftRedoc

2.1.1. Program Code Documentation

SoftRedoc can process two different types of programs:

- procedural programs and
- object oriented programs

2.1.1.1 Procedural Program documentation

For procedural programs five different view functions are invoked

- a function to outline the contents of the program

- a function to document the program interfaces and database accesses
- a function to document the program control flow and decision logic
- a function to document the data flow thru each procedure
- a function to document the data used

2.1.1.2 Object-oriented Program Documentation

For object-oriented programs there are seven view functions offered

- one to list out the classes and methods
- one to list out the class method calls
- one to list out the class interfaces
- one to list out the class attributes declared
- one to list out the database accesses
- one to document the control flow logic of each method
- one to list out the test cases required to traverse all control branches

The user may of course select which of these view functions should be performed. The views will be printed out in a series of text documents.

2.1.2. Database Schema Documentation

The database schema documentation function produces a table of contents for each database schema, listing out the keys, attributes and cross references. The keys and attributes are listed with their types and lengths. This document is useful in giving an overview of the structure of legacy database.

2.1.3. User Interface Documentation

The user interface documentation function produces a table of contents for each user interface depicting the variables and their types and attributes. In addition, it actually creates a sample of the static screens to be printed out. For obvious reasons this cannot be done for dynamic panels and graphical user interfaces.

2.1.4. System Interface Documentation

The system interface documentation function produces a tree description of the interface structure in which the subordinate modes depict the data groups and the subordinate modes depict the data elements. Individual parameters are depicted as single modes. The data types are included adjacent to the data names.

2.1.5. JCL/WFL Procedure Documentation

The JCL documentation function creates a table of jobs in which the jobs are listed out with their steps, the programs they execute and the files and databases they access. This table

gives the user an overview of the order in which the programs are executed and which data stores are used by which programs.

2.2. Repository Input Export

The repository export function is actually a byproduct of the documentation functions. When a documentation view is created, the entities and relationships exposed by that view are passed on to a table generator routine which writes each relationship into a comma separated value file. The relationships are ordered by type and sequence. This table can be used for cross referencing between programs and data as well as for identifying where entities are located. The main purpose is to populate a repository for depicting the overall system architecture.

Altogether five types of export files are produced, one for each artifact type:´

- a procedural program export file
- an object-oriented program export file
- a database schema export file
- an interface schema export file
- a process control-JCL-export file

All of the export file types have the same basic structure. There is a row for every binary relationship between two entities with five columns.

1. Col = Base Entity Type
2. Col = Base Entity Name
3. Col = Relationship Type
4. Col = Target Entity Type
5. Col = Target Entity Name.

3. Inputs to the SoftRedoc Tool

The principle inputs to the SoftRedoc tool are the batch export files of a UML tool or the code members of a source library. These source members can be:

- Programs
- Classes
- Database schemas
- User interface maps
- Interface definitions and
- JCL procedures

In addition to these source inputs, SoftRedoc accepts a number of parameters from the user interface including the document selection criteria and the macro/function definition list. (*see Figure 3: SoftRedoc Inputs*)

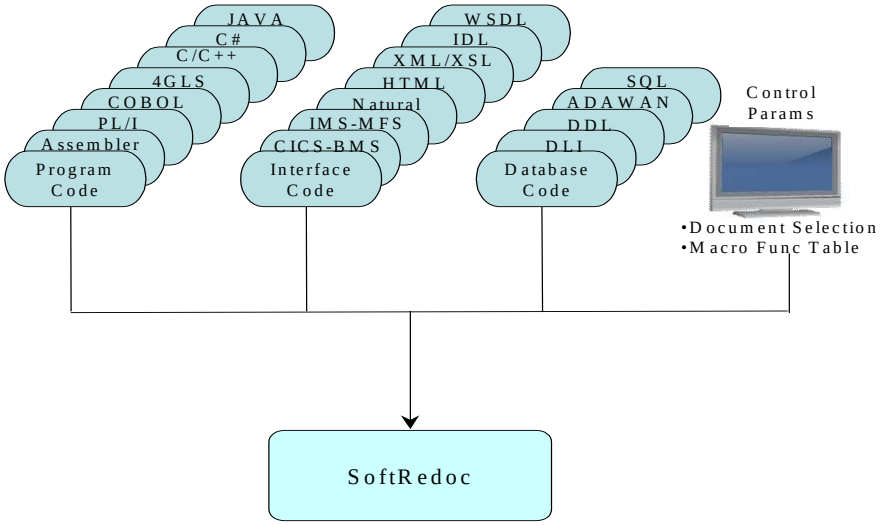


Figure 3: Inputs to SoftRedoc

3.1. Source Text Files

The Source text files can be situated in different source libraries. They should be in a standard Microsoft text format with the Microsoft PC-Standard line end and file end characters. Lines should be no longer than a maximum of 1000 characters.

SoftRedoc presupposes a specific directory hierarchy in order to distinguish between systems, components and modules. The top of a given hierarchy is a system. It is identified by the system name. The next level under the system is the component or program level. Each component or program is a subdirectory of the system directory. Under the component level is the module level, a module being the lowest directory level under which there are only source files. In the module sub-directory are all of the sources belonging to that module including its header or macro files. A component sub-directory may have any number of module sub-directories. If the source library has more than three levels, SoftRedoc has a feature which

will flatten the directories by moving all of the sources lower than level three up to the third or module level. This feature is referred to a directory flattening. It is necessary for making systems with multiple languages compatible with one another and for deriving a normalized documentation.

In the case of procedural systems the components will be either online programs or batch jobs. The modules will correspond to online transactions or batch job steps. The modules may include one or more sources.

In the case of object-oriented systems, the modules correspond to build units or execs. A module here may include several source files, for instance the class files and the header files.

Procedural systems have common copy, include and macro libraries from which source members are copied into the program source members at compile time. These libraries should be included as sub directories under the system library. They are processes separately by SoftRedoc so as not to bloat the program documentation with excessive data definitions. Only the references to the copies are documented. Copy or include members can also reference other copy or include members. The reference to individual items- procedures or data fields – within the copies are resolved within the repository by the tool SoftRepo.

This same strategy is applied to object-oriented systems written in C++, C# and Java. Instead of copy and include members these systems use header files and methods. These too are processed separately and are linked to the inheriting classes in the SoftRedoc repository.

SoftRedoc distinguishes between source file types depending on their extensions. Accepted program extensions are:

- asm/ass/mac = Assembler programs and macros
- plI/pl1/inc = PL/I programs & includes
- cob/cbl/cpy = COBOL programs & copies
- nat = Natural programs
- asp = ASP Programs
- csp/esf = CSP Programs
- del = Delta programs
- ezt = Easytrieve Programs
- aba/abap = ABAP Programs
- c/cbe/hbe/h,res = C programs and header files
- cpp/cxx/h = C++ classes & header files
- css = C# classes
- jav/java = Java classes
- res = Resource Files

Accepted interfaces extensions are:

- mfs = IMS-MFS maps
- bms = CICS-BMS maps
- nam = Natural maps
- scr = COBOL screens
- idl = IDL interfaces
- html = HTML web page definitions
- xsl = XSL templates

- xsd = XML schemas
- xml = XML interface files
- wsdl = web service interface definitions

Accepted database extensions are:

- dli/dbd/psb = IMS databases
- ddl = CODASYL databases
- ada = ADABAS databases
- sql = SQL databases

3.1.1. Program Sources

Program sources contain executable statements in a procedural or object-oriented language. In case of a procedural language, one source file is generally equivalent to a module. A module can be expanded by including other source members from a common copy or module library. A pointed out above, these sources are processed separately. In case of object-oriented languages, one source member is usually equivalent to a class. Classes are expanded by importing other classes. In SoftRedoc each source is processed separately meaning that each class is processed separatel, meaning that each class is processed separately. C++ has header files which fulfill the same role as include members in procedural languages. Program sources can be written in any one of the following languages:

- IBM Assembler or Macro Assembler
- PL/I
- COBOL-74, COBOL-85 or Object COBOL
- C or C++
- C#
- Java
- Natural
- APS
- CSP
- Delta
- ABAP

3.1.2. Interface Definition Sources

Interface sources contain definitions of user or system interfaces. User Interfaces are maps for 3270 type terminals, screens for COBOL online programs such as for the I series, GUIs, or pages for web applications. System interfaces are protocols for exchanging data between distributed systems. They can be either RPC pr message type interfaces. Interfaces can be defined in any one of the following languages:

- IMS Message Format Service (MFS)
- CICS Basic Map Service (BMS)
- Natural Map Service (NAT)
- COBOL Screen Definition Language (SCR)
- CORBA Interface Definition Language (IDL)

- Hyper Text Markup Language (HTML)
- Extended Markup Style Sheet (XSL)
- XML Schema Language (XSD)
- Extended Markup Language (XML)
- Web Service Definition Language (WSDL)

3.1.3. Database Description Sources

Database sources encompass the schema of a particular database. The schemas differ for hierarchical, networked and relational databases. Hierarchical databases are defined in terms of physical and logical definitions with a hierarchy of segments. Networked databases are networks of record types divided up into areas. Relational databases are tables of attributes with keys and indexes. The database sources must be in one of the following database description languages.

- IMS DLI
- CODASYL DDL
- ADABAS – ADAWAN
- SQL - DB2, Oracle, SQL-Driver

3.2. GUI Parameters

The ***SoftRedoc*** user interface gives the user the opportunity to set the control parameters, to select documents to produce and to edit the macro / function table. He may also determine which exports should be produced.

3.2.1. Control Parameters

The Control parameters to be submitted by the user include the following names and directories:

- Name of the software product
- Name of the application system
- Output directory for storing the results
- List of source files to be processed
- List of Options for the documents to be produced

In addition to these names and directories, there are three yes/no selections:
- Output directory for storing the results
- To printout a list of all comments
- To Export a SCV file for the expository tool SoftRepo
- To export a XML file for the estimator tool SoftCalc

3.2.2. Document Selection List

Depending upon the language being processed, the user can select which documents should be generated. A list of the possible documents is displayed with check boxes and the user can check which documents he wants to have.

3.2.3. Macro / Function Table

The macro/function table contains a row for each macro or function with 3 columns. For the procedural languages Assembler, PL/I, COBOL, APS, Natural and ABAP the rows are macro operations. For the object–oriented languages C, C++, C# and Java the rows are standard functions for I/O operations.

The first column is a four letter code for the type of macro or functions. The second column is an 8 or 40 character string giving the name of the macro or function. The third column is used only by ***SoftAudit***. It is the number of function–points associated with each I/O operation or file type. It can range from 3 to 6 for Input operations, from 4 to 7 for Output operations, from 5 to 10 for Files and from 7 to 15 for Databases.

In the case of macros the name of the macro is only 8 characters, followed by the number of parameters for the macro and the position of the parameter with the name of the object being processed by the macro, i.e. the file, record datagroup or map.

READ GETNEXT, PARAMS = 3, OBJECT = 1

The 25 valid macro or function types are:

- CALL = A procedure or method is called
- CLOS = A file or database ir opened
- DB = A database is declared
- DECL = A datatype is declared
- DELT = A file or database record is deleted
- ENTR = An entry to the module
- EXIT = An exit from the module
- FILE = A file is declared
- FUNC = A function to compute a value
- GETS = A data storage is allocated
- INCL = Another source is copied in
- INPT = An input operation
- ISRT = A database, record is inserted
- LIST = A report is printed
- MASK = A user interface is declared
- MESS = A message is declared
- OPEN = A file or database is opened
- OUTP = An output operation
- PROC = A procedure is declared
- RECV = A user interface or message is received
- READ = A file record is read
- SELT = A database record is queried
- UPDT = A database record is updated
- WRIT = A file record is written
- SEND = A user interface or message is sent.

One sample macro table and one sample function table are given in Samples A und B. *(See Sample 6.1 – Macro-Table und Sample 6.2 – Function-Table).* As can be seen, it is impossible to identify the input/output, DB and TP operations without the help of such tables. Therefore it is not possible to count function–points without them. Audits can be run using the standard default tables for each language but the results will not be as accurate as when the user fills the tables with his own specific macro and function names and assigns them the proper type.

In SoftRedoc the macro / function table is used to identify entries, exits, inputs, outputs, calls to standard framework functions and database accesses. This is important for documenting the component interfaces.

4. Outputs from the SoftRedoc Tool

The outputs from SoftRedoc are of four types:

- A set of text documents´
- A list of comments
- CSV export files for the product repository
- XML export files for the project calculation

(*see figure 4: SoftRedoc Outputs*)

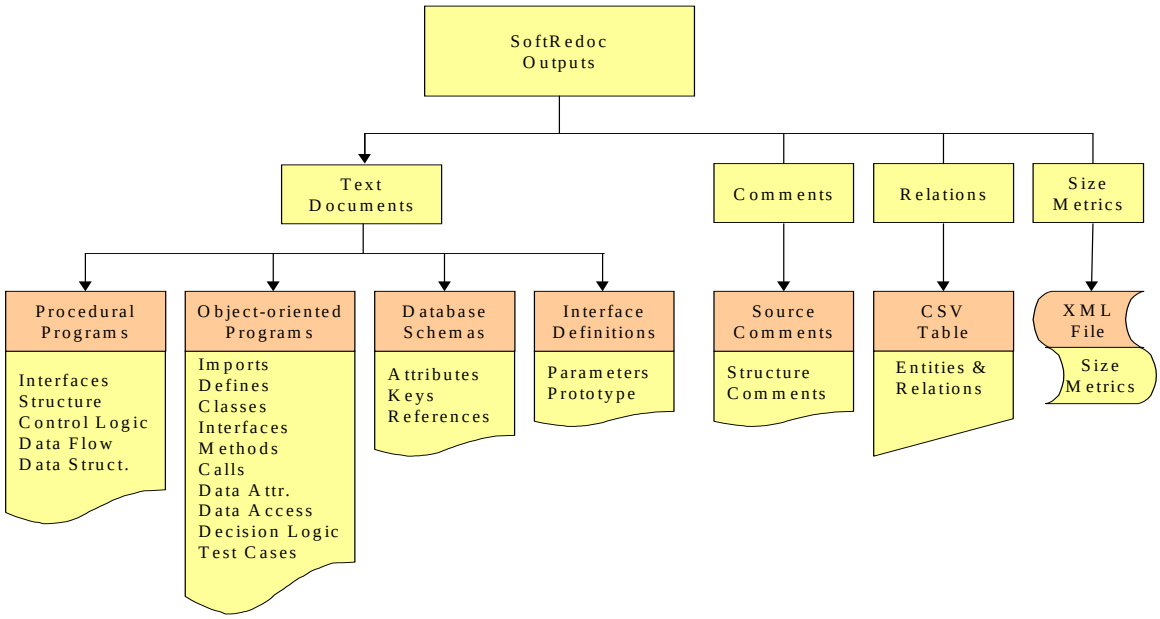


Figure 4: Outputs from SoftRedoc

4.1. Text Documents

The text documents provide views on the internal structure of the software. The views differ depending on the type of the software. For procedural and 4GL programs there are five views, for object-oriented programs there are ten views, for database descriptions there are two views and for interface definitions there are also two views. Each view is an abstract of the code, highlighting one particular substructure, similar to a computer tomography which brings out one aspect of the human organism and blends out the others.

The views are all presented as a structured list of selected code elements representing that particular view of the code. That document is referred to here as a program specification.

4.1.1. Procedural Program Specification

The five procedural program specifications for the language Assembler, PL/I, COBOL, Natural, etc. are:

- Interface List
- Structure List
- Decision logic tree
- Data flow table
- Data structure List

4.1.1.1 Procedural Interface List

The procedural interface list is a list of all interactions between the program and its environment. These include all entries and exits, all calls to external programs, all input/output operations and all database accesses module the DPI, CODEASYL and SQL commands. This documents is intended to give the user an overview of the dependencies a program has on its run time environment, i.e. its links to the TP framework, to the files and to other programs. This is very important when it comes to extracting a program from its environment for reuse in another environment, e.g. for wrapping (*see sample 6.3*).

4.1.1.2 Procedural Structure List

The procedural structure list is a list of all procedural building blocks of the program under consideration. These include the labels, paragraphs, sections, subroutines and internal procedures as well as the macro, copy and include references to foreign source members. Not only are the program construction blocks given, but also all of the references to those blocks – the branches, GOTOs, PERFORMs and internal calls. This document is intended to provide the user an insight into internal structure of the program, i.e. which procedural units it consists of and where these procedural units are referenced to (*see sample 6.4*).

4.1.1.3 Decision Logic Tree (Struktogramme)

The decision tree is a nested structure of the program control flow depicting all of the executable statements in the form of sequences, alternative branches and repetition loops. Sequentially executed statements are listed out under each other. Alternative statements are intended and marked with a "@" sign to indicate that they can be executed if their guard condition is fulfilled. Loop statements are intended and marked with a "*" sign to indicate that they are repeated if the loop conditions is fulfilled. The tree is intended at each alteration and repetition level up to a maximum of 10 nesting levels. This document is intended to give the user an overview of the program control logic in terms of the Michael Jackson structured programming technique. If a change has to be made to the code, he can see here where to best place the change without affecting the remaining control logic. Here the user can trace execution paths through the program tree (*see sample 6.5*).

4.1.1.4 Data Flow Table (HIPO Charts)

The data flow table depicts the flow of data through the procedural construction blocks. Each construction block – labeled code sequence, paragraph, main routine, sub routine, internal procedure – is represented as an input, process, output, diagram with three columns. The input variables are listed out in the left column, the predicates or decision variables are listed in the middle column, the output variables are listed out in the right column. This document is intended to give the user a view of the use of the data by the individual procedural blocks, i.e. which data is used by the individual procedural blocks, i.e. which data is used by which procedure in which way. Together with the procedure structure list it fulfills the requirements of the HIPO method as applied to documenting programs (*see sample 6.6*).

4.1.1.5 Data Structure List

The data structure list is an extract from the data definitions of a module showing only the declarations of data that is actually used. The declared data is signed as an input, an output or both. For locating the individual data fields, the upper level data structure names are also listed out even if they are not used. The usage of each data item supplements the data characteristics like name, type length and location. This document is intended to give the user an overview of the data required by a module to accomplish its tasks (*see sample 6.7*).

4.1.2. Object-oriented Program Specification

The ten views of an object-oriented component are:

- Include/import list
- Defines list
- Class list
- Method list
- Method reference list
- Interface list
- Class attribute list
- File/database access list
- Condition logic specification
- Test case specification

4.1.2.1 Include/Import List

The include/import list is a list of all source members included or imported into the source code at compile time. It is intended to provide an overview of the static dependencies a source has from other code sources (*see sample 6.8*).

4.1.2.2 Defines List

The defines list pertains in C and C++ to the compile time options. It is a list of all the compile time definitions. C Code can be compiled or left out depending on these options and static variables can be assigned a constant value. Rather than use the number, the code can refer to the symbolic names assigned via the status of the compiler options. In C# and Java it contains the static constants and enumerations (*see sample 6.9*).

4.1.2.3 Class List

The class list is a list of all classes defined within the source and from which base or super classes they are derived from. In C++ and C# the base classes are proceeded by a colon. In Java the word “extends” denotes the super class. This list is intended to give the user an

oversight of the classes contained within a source member and what higher level classes they are dependent on (*see sample 6.10*).

4.1.2.4 Class Method List

The class method list is a list of all the methods or procedures declared within a class. Under each class definition the methods of that class are listed out together with their parameters. Methods which can be invoked from anywhere are marked as public. Private methods which are available only to other methods within the same class are proceeded by their data type. Methods available only to lower level classes are protected. This list gives the user an overview of the functions available within the classes of this source (*see sample 6.11*).

4.1.2.5 Class Method Reference List

The class method reference list is a list of all foreign methods, i.e. methods outside of this source, invoked by the classes within this source. Each time a foreign method is called it is listed here together with the line number where it is called. Since the same method can be invoked many times, there can be many duplicates in the list. This is intentional to show the user the degree to which a class is dependent on functions in other classes. This list documents the runtime dependencies of the classes. It reflects the class sequence diagram *(see sample 6.12*).

4.1.2.6 Class Interface List

The class interface list is a list of all interfaces defined within the source. In C and C++ the interface are usually defined within the header file sources. They can be real or virtual. In C# and Java interfaces are defined within the main code body. Like methods, interfaces can be public or protected. This list gives an overview of the interfaces thru which this component can be accessed. The number of interface is an indicator of the degree of coupling (*see sample 6.13*).

4.1.2.7 Class Attribute List

The class attribute list is a list of all data types declared within the source by class and by method. The data attributes are given with their type and their accessibility. Like methods, data attributes can be public, protected or private. Public attributes are available only to methods in sub classes, private attributes are restricted to methods in the class where they are declared. The number of private attributes relative to the number of methods is an indicator of the class cohesion (*see sample 6.14*).

4.1.2.8 File/Database Access List

The file/database access list is a list of all file and database access operations. These can be direct input/output commands like read, write, open and close, SQL statements like SELECT, INSERT, DELETE and UPDATE or indirect function calls like ODBC and JDBC method invocations. This list is intended to give the user an overview of the dependencies a

module has on external data stores. Input/Output operations and database accesses are also an indicator of the number of function-points a component has (*see sample 6.15*).

4.1.2.9 Condition Logic Specification

The condition logic specification contains a structogram for each method in which the decision structure of the method is depicted. If, case and loop conditions are indented and placed in boxes to make them easier to view. In effect, this specification amounts to a pseudo code notation to aide in comprehending the detailed code of the individual *methods (see sample 6.16*).

4.1.2.10 Class Test Case List

The class test case specification lists out which conditions have to execute all paths through each method. A test case corresponds to a path through a method. Each test case has a number of uniquely identity in it. All of the conditions which it has to fulfill are listed out next to the test case number. This specification is intended to aide the user in preparing a class test whose goal it is to execute every possible path thru the code (see sample 6.17).

4.1.3. Database Description Texts

Two documents are produced from each database schema. The schemas can be in DLI, in CODASYL COBOL, in ADABAS-ADAWAN or in SQL. The two documents are

- A list of the database attributes
- A list of the database cross references

4.1.3.1 Database Attribute List

The database attribute list contains a list of all data elements defined within each segment, record or table. The elements are listed out in a table with their names, types and lengths. In three additional columns, there are indicators to mark weather the elements are primary keys, foreign keys or alternate keys and indexes. The list is intended to give the user an overview of the contents of the database in a normalized form for all database types. It is possible to search thru the list to find where a given data attribute or key is located, i.e. to which segment, record, or table it belongs to (*see sample 6.18*).

4.1.3.2 Database Cross Reference List

The database cross reference list is a list of all interconnections between database entities. Interconnections exists when there is a reference from one entity to another. These references can be in the form of logical keys, set pointers or foreign keys. The base entities - record, segment or able – are listed in the left column, the connecting data elements are given in the middle column, the target entities are listed in the right column. This document provides the user an overview of the database connections and which data keys serve to establish these

connections. When changing a database, it is important to know the dependencies between the database entities (*see sample 6.19*).

4.1.4. Interface Definition Texts

Two documents are also produced from each interface definition. The interface definitions can be map descriptions in CICS-BMS, IMS-MFS, Natural, COBOL or HTML or system interfaces in IDL, XML or WSDL. In all cases, the interfaces contain data elements, i.e. parameters and have a particular layout. Consequently the two documents are

- A list of the interface contents
- A sample of the interface layout

4.1.4.1 Interface Content List

The interface content list is a list of the variables or data elements contained within the interface. They are listed out by name. To the right of them their types and lengths are given. In addition there are indicators of weather the fields are used as an input, an output or both.
This list is intended to give the user an overview of the contents of the interface in a normalized format for all interface types. The user can search for selected variable names in the list to find in which interfaces they appear and how they are used (*see sample 6.20*).

4.1.4.2 Sample Interface Layout

The sample interface layout is a prototype of the interface showing how it will appear to the user. In the case of user interfaces, it is a map of the panel with the titles and representation values. In the case of system interfaces, it is a set of test data in an IDL stack, an XML file or a web service interface. It can be used to test the respective interface,
The sample interfaces are intended to aide in testing a system. They give the tester an insight into the interfaces to be tested and what combinations of test data are required. They can even be used as an input to a system testing tool such as ***DataTest*** (*see sample 6.21*).

4.2. Comment Lists

The comment list is an abstract of all comments extracted from the source code of a program, a database schema or an interface definition. Every comment line is listed out here together with those lines which identify the source segment. In procedural programs the source segments are identified by the high level data structure lines and the label lines of procedures, paragraphs, sections, subroutines, etc. In object-oriented programs the source segments are the class and the method definitions. In the case of a database schema, the segments are identified by the record or table definition lines, for interfaces the interfaces definition for the mask, page or header are listed out together with the comments.

Provided the source is well commented, this document can help the maintenance programmer to understand what the program is all about. He can see the structure of the program with its individual code segments and gather from the comments what purpose the systems are

serving. For this reason every data structure, procedure, subroutine and method should have a comment header describing what that code segment is intended to do (see sample 6.22).

4.3. CSV Export File

The CSV Export file is created to export the code entities and their relationships to a software repository. There is one made for every subsystem, database and interface set analyzed. As an CSV file it contains a line for every relationship starting with the product which owns that subsystem, database or interface set going down to the level of each procedure or method which uses a particular data item. Each line has five fix length columns separated by semicolons.

Col.1 = Base Entity Type
Col.2 = Base Entity Name
Col.3 = Base to Target Relationship Type
Col.4 = Target Entity Type
Col.5 = Target Entity Name

The base entity is always pointing to the target entity, e.g. a system owns a program, a program owns a class, a class owns a method and a method owns another method. Relationships can be of a horizontal or a vertical nature.
Vertical relationships imply that the base entity is owner of the target entity or that the base entity belongs to the target entity as when one class inherits from another. Horizontal entities imply that the base entity uses the target entity in some way, i.e. it calls it, it reads it, it writes it, it includes it, etc. A specification of all entity and relationship types is to be found in annex A.
This CSV file serves as a bridge from the SoftRedoc source analysis to the SoftRepo repository, where the code relationships delivered here are combined with the relationships extracted from the requirements and test documents. The goal is to be able to connect all software elements to each other and to be able to trace requirement elements to code elements and code elements to test elements as well as visa versa. The repository should be the ultimate source of information for a maintenance job (see sample 6.23).

4.4. XML Export File

The XML export file is created to export size metrics to a project calculation tool. There is one created for each subsystem, database and interface set analyzed. Besides the usual header information with the product, system source and date, the file contains a data group for each code entity. A code entity can be a module, data object or an interface. The size metrics given depend on the entity type.
A module has:

- Code lines
- Statements
- Functions
- Variables

A data object has:

- Relations
- Methods
- Keys
- Attributes

An interface has:

- Receivers
- Operations
- Arguments
- Results

The file is in a standard XML format so that it can also be processed by any other tool. The purpose of these size metrics extracted from the source is to estimate the costs of maintaining, migrating, reengineering or integrating the source. These are exactly those size metrics required by the SoftCalc project estimation tool (*see sample 6.24*).

5. SoftRedoc Usage

The tool ***SoftRedoc*** offers the user a graphical user interface from which he can invoke the redocumentation functions. The user interface will come up when the SoftRedoc.exe is clicked on. Prior to starting the tool, the user should have organized the input source files by product, system, component and module and set up an output directory for each product. For measurement purposes it is very important to have the source files grouped by component and module. All of the sources of a product should be collected together under one main directory with sub directories for systems, components and modules. SoftRedoc allows only three subdirectories

- Systems
- Components and
- Modules

If there are more than the lower level sources will be automatically brought up to the module level – directory flattening.

When the SoftRedoc.exe is clicked the user is shown the language selection panel. Here all of the documentable languages are displayed with a check button

- Procedural Languages = Assembler, PL/I, COBOL
- 4GL Languages = Delta, APS, CPS, Natural, ABAP
- Object–oriented Languages = C,C++, C#, Java
- Database Languages = DLI, DDL, ADABAS, SQL
- Conventional Interface Languages = BMS, MFS, Natural
- Contemporary Interface Languages = HTML, XSL, XML, XSD, WSDL, IDL

The user may select only one language to process at a time. If a system is using different languages then the documentation job must be repeated for each language using the same output directory. In this case each language group will be treated as a separate subsystem and documented in parallel. The union of the different language components occurs only in SoftRepo where all of the documented entities and their relationships are joined together.

The SoftRedoc user interface has two menu bars at the top and an alternating user panel. The first menu bar contains the general purpose selections. The second menu bar has the redoc specific selections.

The content of the user panel depends on the selection in the second menu bar. The default content is the language selection panel.

5.1. General Selection Bar

In the general selection bar the user can choose between the general purpose functions

- File
- Actions
- Viewers
- Logs
- Help.

5.1.1. File Operations

In selecting the menu item „Files”, a pulldown menu appears with 7 standard functions

- open existing job
- new job
- save job
- save job as
- import a word list
- export a word list
- exit.

The „open existing job” option causes a pop–up window to appear in which the user can search for a file containing the redoc job parameters of a previous run in a standard windows directory.

The „new job” option brings up the user panel for setting the redoc job parameters.

The „save job” option causes the current job parameters to be restored in the file where they had been before.

The „save job as” option causes a pop–up window to appear with a standard windows directory in which the user can find a place to store the redoc job parameters. They are stored in a file whose name is composed from the product and the system name with the extension .jsfr.

The „import a word list” option causes a pop–up window to appear in which the user can search for a valid macro or function table from which to load the function/macros for the current language.

The „export a word list” option causes a pop–up window to appear with a standard windows directory in which the user can store the current macro table.

The „exit” option will cause the SoftRedoc tool to be closed.

5.1.2. Actions

In selecting the menu item „Actions” a pulldown menu comes down with the options:

- Add files to the job
- Add complete directories
- Start prescanner
- Start analysis
- Reset output
- Skip rest of processing.

The option „add files to the job” gives the user a view of the windows directories from which he can select individual source text files to be audited.

The option „add complete directories” gives the user a view of the windows directories from which he can select whole subdirectories of source files to be audited.

The option „start prescanner” triggers the process to scan thru the selected source files for unknown statements and to store them in the macro file for the user to define.

The option „start–analysis” triggers the auditing process to check and measure the selected source files.

The option „reset output” will cause the output directory contents to be deleted. It should be used with caution.

The option „skip rest of processing” causes the currently running audit process to be interrupted and the results to be reset.

5.1.3. Viewers

The selection of the menu item „Viewer” results in a pull–down menu with the options:

- Open a file with the general viewer
- Open a deficiency report
- Show metrics
- Call external editor.

The first option displays the contents of the output directory and allows the user to select any file in the output directory for viewing.

The second option displays the contents of the SPECS subdiractory and allows the user to view the selected component specification lists, list by list.

The third option displays the contents of the COMMENTS subdirectory and allows the user to view a selected comment list at the module, level.

The fourth option allows the user to use any other external editor to view the results of the audit, e.g. notepad, wordpad, etc.

5.1.4. Logs

When the menu item „Logs” is selected, the log of the last audit or scanner run will be displayed.

5.1.5. Help

By selecting the menu item „Help” the user is shown the table of contents of this user documentation and can select the procedures or methods he wants to read. With the search topic option he can submit a key word and see where this word is used in the documentation.

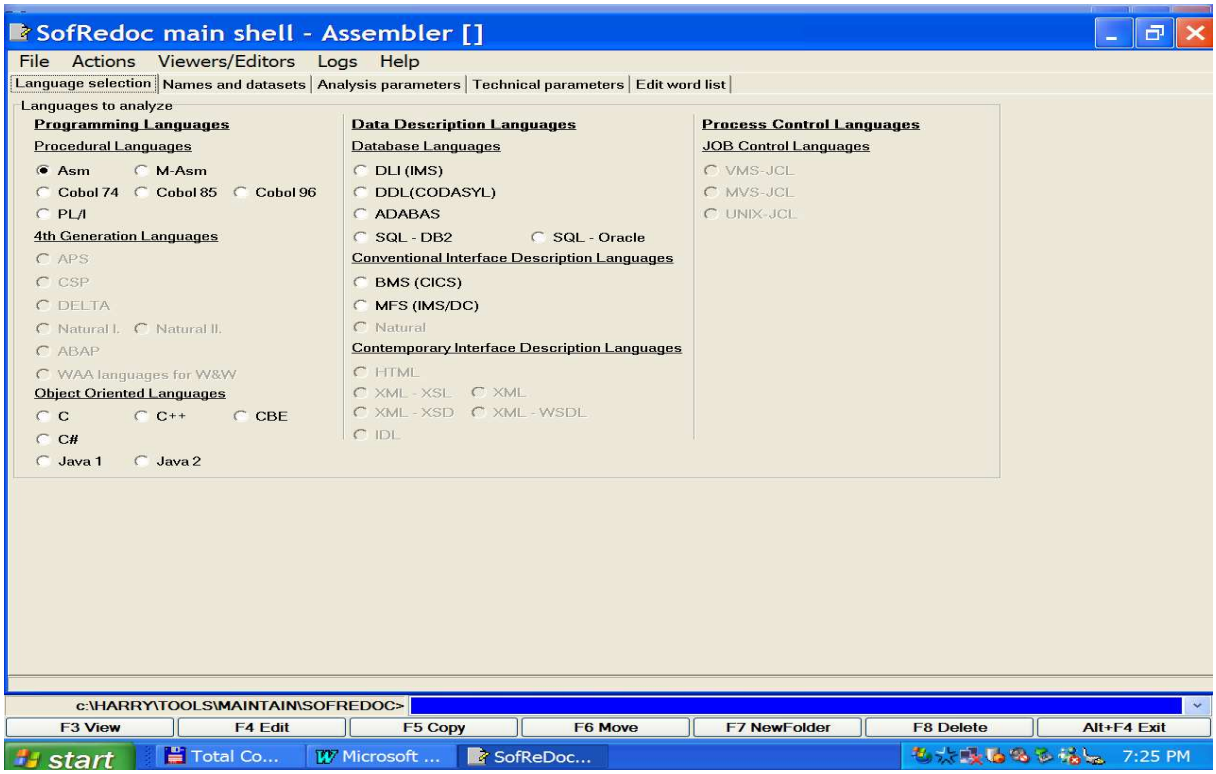
5.2. Specific Selection Bar

The specific selection bar contains five buttons, one for each of the preprocessing functions for SoftRedoc. The user can choose to:

- make a language selection
- set the names and data sets
- set the redoc parameters
- set the technical parameters
- edit the word list.

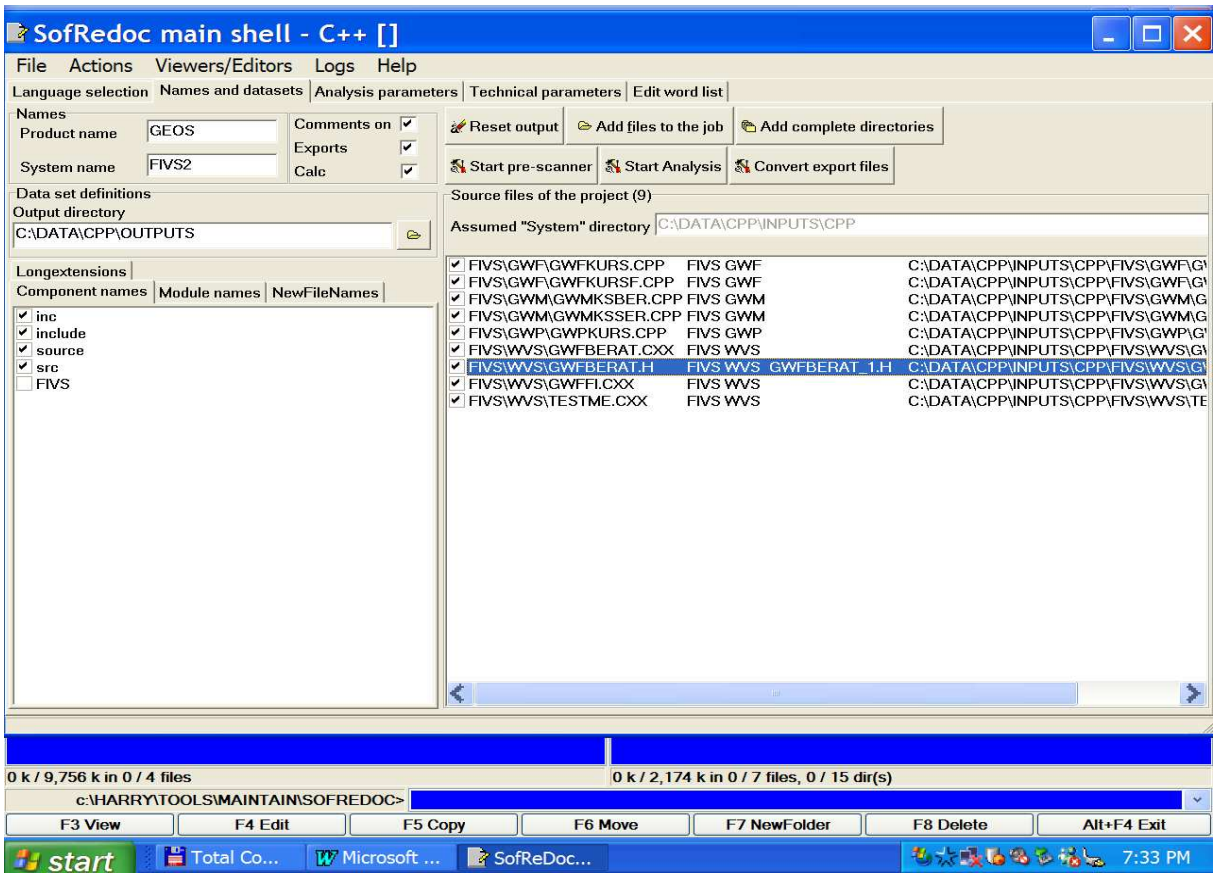
5.2.1. Language Selection

The language selection option brings up the user panel for the selection of a language to process. This panel offers all of the languages available with a check button. The user may select any one language by clicking on its button. They can be selected in any order. SoftRedoc is constructed to process one language at a time independently of each other. *(see Screenshot 1: Language Selection)*



5.2.2. Names and Data Sets

Selecting the „Names and Data Sets" option will bring up the panel for setting the basic job parameters of an audit run. At the top are the product and system names, the audit parameters and the audit functions. At the bottom the user can select an output directory on the left side and the target sources on the right size. *(see Screenshot 2: Selecting Inputs)*



The product and system names are 8 character strings intended to uniquely identify the product and system being audited. A product may have several subsystems. In this case there would be several audit runs for one product each with a different system name. These product and system names are used to identify the summary reports and the xml export files.

The three parameter buttons are

- comments,
- exports and
- Calc

When the comments button is on, the comments in the source code will be listed out in a separate report together with the structural statements, i.e. data structures, procedures, classes, methods and so on. When it is off, the no comment list will be produced.

When the exports button is on, the export files to SoftRepo will be generated. This is a prerequisite to the graphical systems documentation and the impact analysis of system changes.

When the calc button is on, an interface file to the cost estimation tool SoftCalc will be generated to load the calculation metric database.

The 6 possible SoftRedoc functions are

- Reset Output
- Add files to the job
- Add complete directories
- Start pre–scanner

- Start Analysis
- Convert documentation

Reset Output causes the contents of the output directory to be deleted.
Add files to the job allows the user to select individual source files from an input directory.
Add complete directories allow the user to select all of the source files in a given directory.
Start Prescanner starts a background process to identify user specific macros and methods in the source.
Start Analysis starts a background process to redocument all of the selected source files.
Convert documentation starts a background process to convert procedural documents over into object-oriented ones.

5.2.3. Analysis Parameters

The option „Analysis Parameters" gives the user the option of selecting which documents he would like to have. The option depends on the language type being documented. For procedural languages the possible documents are:

- External Reference List
- Procedural Structure List
- Decision Logic Structure
- Data Flow Tables
- Data Usage Table.

For object-oriented languages the possible documents are:

- Includes List
- Defines List
- Class List
- Method List
- Method call List
- Method Interface List
- Attribute List
- Files & Database Access List
- Conditional Structure List
- Test Case List

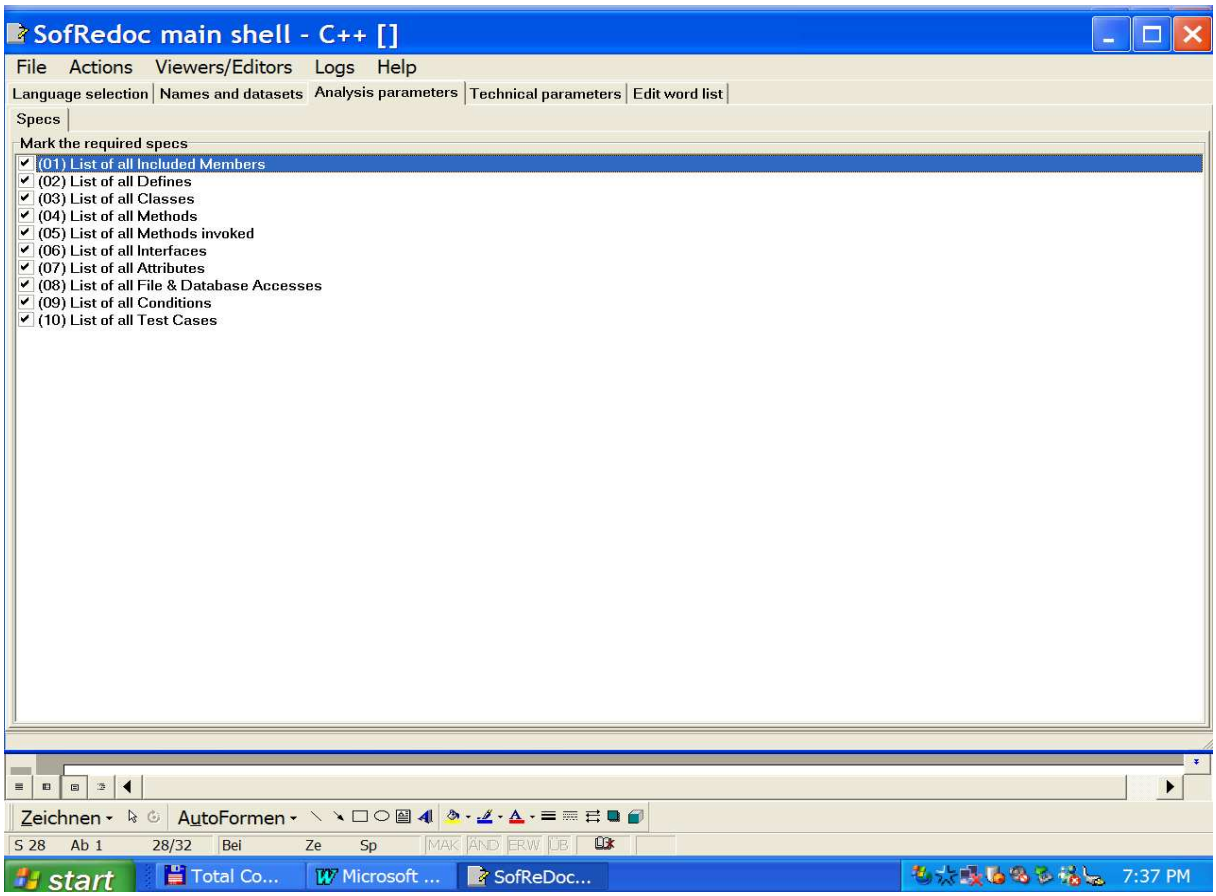
For database schemas the possible documents are:

- List of keys & attributes
- Cross reference List

For user interfaces the possible documents are:

- List of screen contents
- Screen prototype

The contents of these documents have been described in Chapter 4 *(see Screenshot 3: Selecting Documents)*



5.2.4. Edit Word List

The option „Edit Word List" will cause the current macro or function table for the target language to be displayed in two columns

- internal type and
- macro / function name.

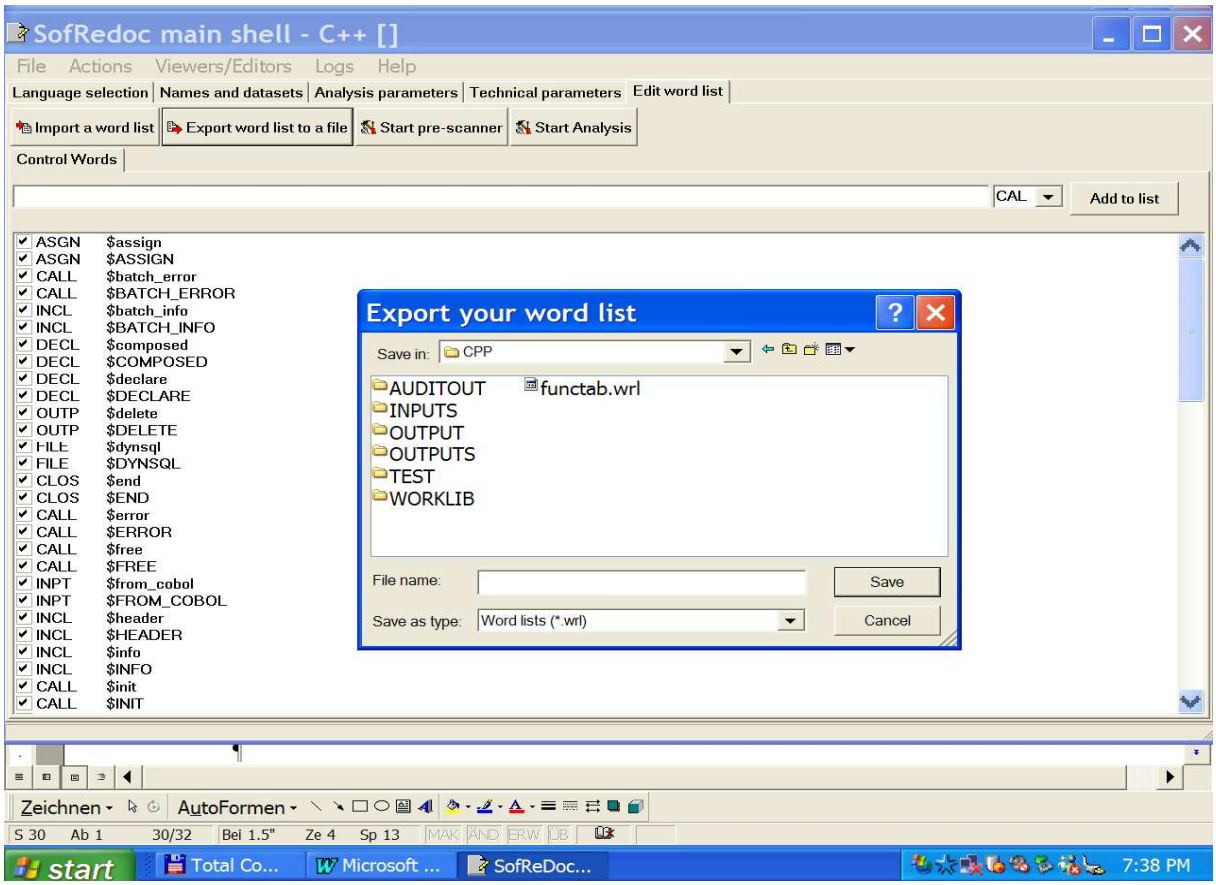
After a prescanning run, the macros and functions are displayed but their internal type will be unknown. The names of all macros will be listed. It will be up to the user to assign them one of the 30 predefined internal types.

Otherwise the contents of the existing standard table of functions will be displayed. The user can overwrite the macro or function names. He may also add additional macros or functions and assign them one of the predefined internal types. If the user wants to use an existing table from a previous analysis he may import it using the import function „Import Word List".

When the table is ready the user can export it by selecting the button „Export Word List". He will then be given a view of the windows directory and a box for assigning the name. The default name is FuncTab. It is recommended to store the FuncTab in the input directory together with the sources to be audited.

Later this table can be read in again by clicking the button „ Import a word list". Otherwise the existing standard table of functions will be loaded.

The check box to the left of the table entries allows the user to remove macros and functions by unchecking them and clicking the right mouse button to remove them from the list. Care should be taken in keeping the tables of macros and functions consistent with the sources they refer to. *(See Screenshot 4: Setting up the Macro/Function table)*



5.3. Scanning for Macros and IO Functions

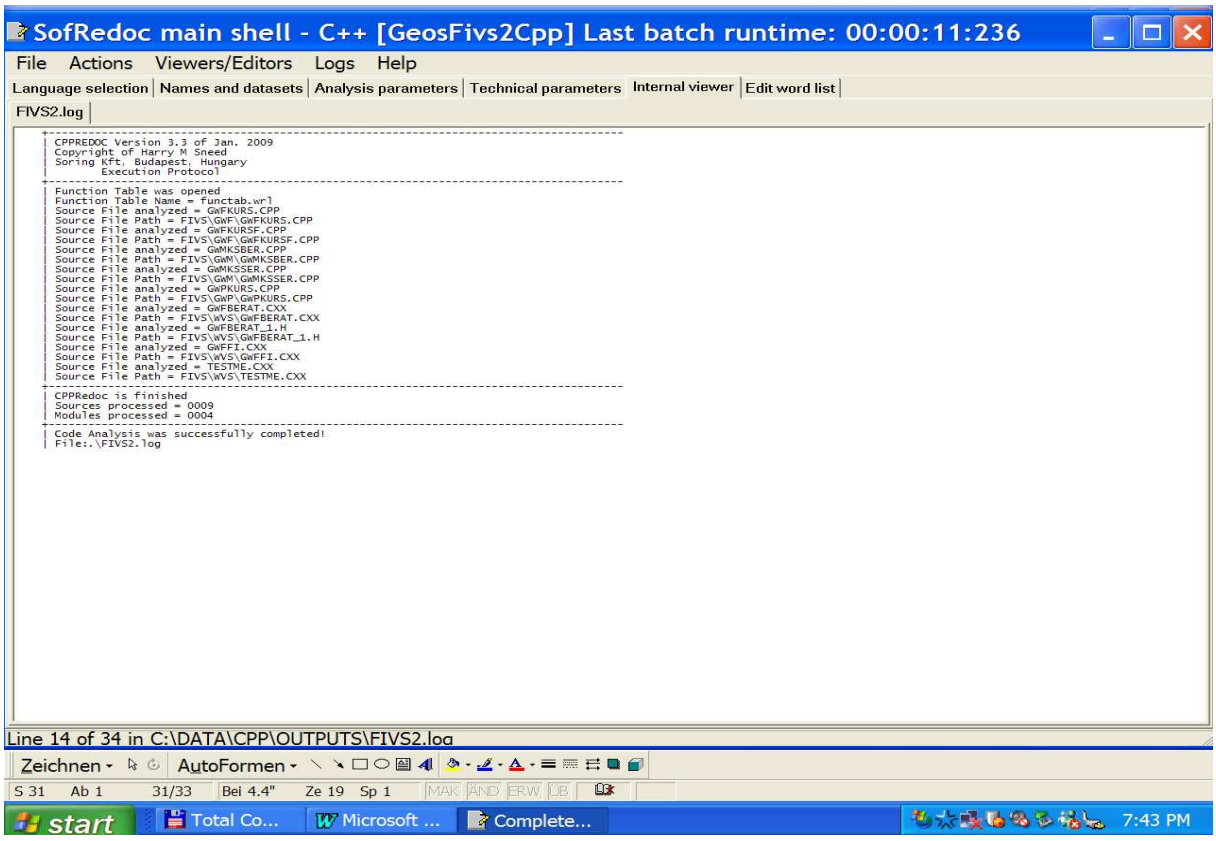
If the user is not sure what user specific macros or functions are contained in the source code of the programs, he should run the sources through the prescanner before beginning a redocumentation run. This is done by selecting either individual source files or complete source file directories. The selected files will be listed out in the text box of the names and datasets panel. Files can be removed by unchecking them.

The user then clicks the button „Start Prescanner”. A background task in a black–box will be started which goes through all of the sources of the target language type and inserts the unknown statements into the macro table for the user to define. When the job is finished the log file is displayed. The freshly created macro table is now the current one. The internal types are marked as ????. They should be replaced by standard codes set by the user prior to the redocumentation run. The purpose of this scan is to identify the I/O and database access operations for the interface documentation. If the source contains special macros or methods for these operations then they must be predefined so that they can be recognized as such. Otherwise they will be handled as standard procedure or method calls making the documentation incomplete.

5.4. Running a Redocumentation Job

To run a redocumentation job the user begins by electing the files or directories to be documented. Those files selected will be listed out in the text box of the names and datasets panel. The user can select individual files or entire directories. Normally the system directory will be selected including all subdirectories. Selected files can be removed by checking them out.

When the user clicks the button „Start Analysis" a batch process is triggered. It processes all of the source text files selected, generating the selected text documents and repository export tables for each source. Instead of aggregating metric data as in SoftAudit, here the code entities and their relationships are collected together in one CSV table pro system. If an error occurs the job is interrupted and the user informed. At the end the user is requested to acknowledge the termination by pressing the enter key. This will cause the log file to be displayed in which all of the documented sources and the possible error messages are listed out. *(see Screenshot 5: Job protocol)*



5.5. Viewing the Redocumentation Results

The last step in using SoftRedoc is to redocument the sources is to view the documents generated. All of them are stored in the output directory assigned by the user. The user can view them by using his own editor or by using one of the SoftRedoc viewers.

To view the results via the SoftRedoc viewer, the user should click the option „Viewers" in the upper menu bar. A pulldown menu will appear which will offer two options

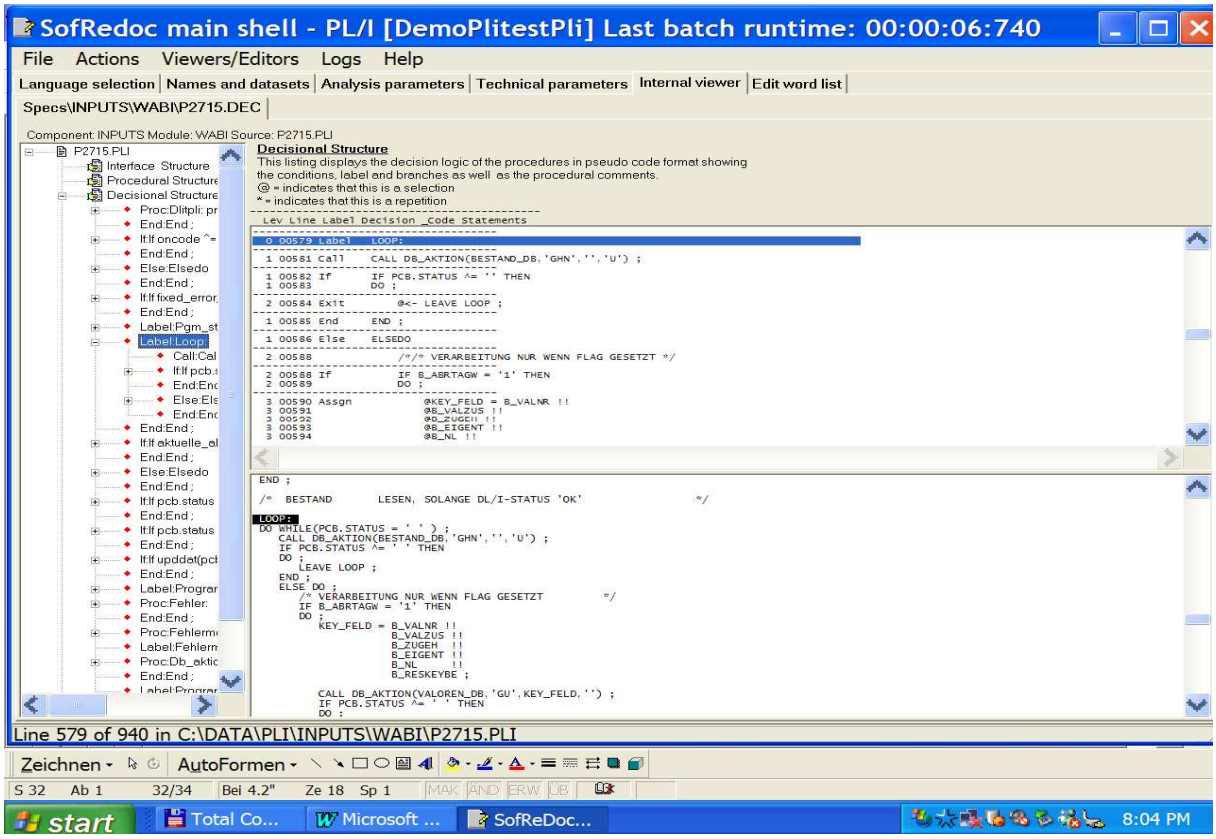
- View a document with the general internal
- View a specification listing.

To look at any output the standard viewer can be selected. It will begin by displaying the contents of the output directory. There are three subdirectories in the output directory:

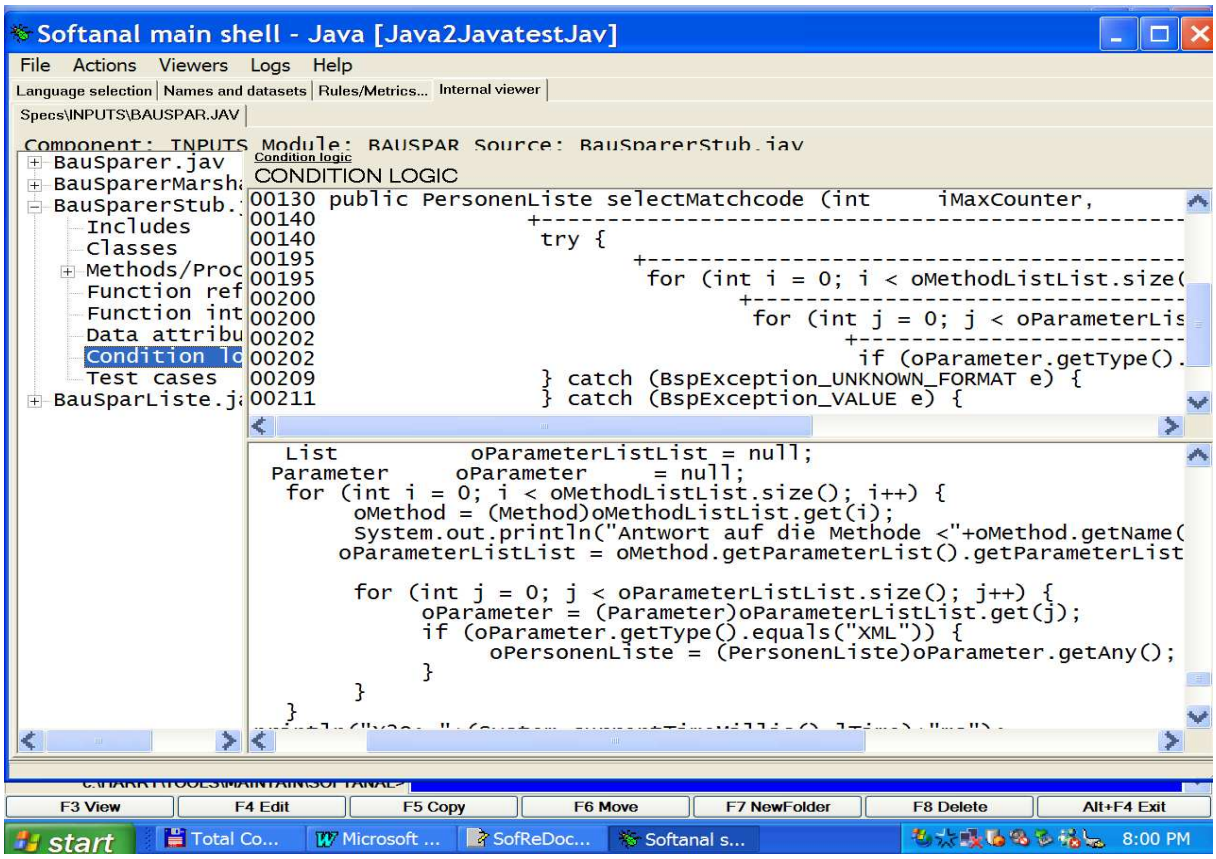
- Comments,
- Exports and
- Specs.

In the Comment subdirectory the user will find the comment texts – one per program, component, database or interface. A file is opened by clicking it on. In the Exports subdirectory are the CSV tables with the entity relationships. These file can also be viewed by clicking them on. In the same directory is an XML metric export file for each system containing a group of key size metrics for every source entity analyzed. In the Specs subdirectory are the actual program documents. For the procedural languages there will be 5 separate text files for each program. For the object-oriented languages there will be one text file for each module containing up to 10 selected documents.

There is a special syntax driven viewer for the program documents in the specs subdirectory. With it the user can select a program and see all five views of that program – the interfaces, the procedural structure, the control flow, the data flow and the data usage. In the left margin is a list of the interfaces, procedures, control commands and data structures contained in that document. The main screen is split in two parts. In the upper part is the generated document. In the lower part is the original source code. Upon clicking a name or command type in the left margin the cursor is positioned on that procedure or those command types in the document in the upper screen as well as on the corresponding statements in the original source. In this way the user can navigate through the source code based on labels and command types. *(see Screenshot 6: Viewing a procedural Program)*



The specification viewer for object-oriented code displays the names of the modules documented in the left margin. When one is selected it lists out the documents generated for that particular module. Upon selecting a document the contents of that document are displayed in the upper screen of the split screen whereas the original source is displayed in the lower screen. When the user selects a particular class, interface, method or method call the cursor is positioned to the location of that selected entity in the document as well as in the original source code. Here too the user can navigate through the source in search of where functions are defined and called and where data attributes are defined and used. This dynamic documentation viewing is a key feature of the SoftRedoc tool. used. definedd rach at claccinte. metric viewer displays all of the metric reports. When the user selects one, its contents are displayed. The user is presented with a view of the quantity, the complexity and the quality metrics of that particular entity. This viewer can, of course, also be used to view the contents of the system and product summary reports. *(see Screenshot 7: Viewing an object-oriented Program)*



5.6. Redocumenting multiligual Systems

Almost all software systems consist of components in different languages. If they use a database, they will have a database language like SQL or DLI. If they have user interfaces they will use a GUI language like HTML, XHTML, BMS or MFS. If they interact with other systems they must employ an interface language like IDL or XML. Finally, the server components are usually in a different source language than the client components. Very often the client or front end components are in an object-oriented language while the server or backend components are in a procedural language. That means that the redocumentation of such systems must also be made multiligual.

SoftRedoc addresses this problem in two ways:

- First by fitting all languages of the same type into a common documentation frame work.
- Secondly, by converting the procedural documents over into an object-oriented structure.

The document types are the same for all procedural languages – Assembler, PLI and COBOL – as well as for the 4th Generation languages. They all have the same interface structure, the same procedural outline, the same decision control structure, the same data flow depiction and the same data usage tables. The document types are also the same for all object-oriented languages – C++, CSS and Java. They all have a list of imports, a list of defines or enumerations, a list of classes, a list of interfaces, a list of methods, a list of attributes, a list of file and database accesses, a pseudo code description of each method and a table of test cases. The documents fit the code into standard patterns.

Before exporting the program entities and relationships to the software repository the procedural documents can be converted into an object-oriented structure. This is done by selecting the function “Convert export files”. If this function is selected it converts the individual procedures over into classes each with it’s own data attributes. It then rearranges the procedures that the procedures at the bottom of the procedural calling hierarchy become the base classes at the top of the class hierarchy whereas the procedures at the top of the procedural hierarchy are placed at the bottom of the class inheritance tree. The other procedures are percolated either up or down depending upon what other procedures invoke them. The final result is a class hierarchy for each documented program. This documentation can then be merged together with the class hierarchies of original object-oriented systems to create a unified object-oriented model. This solves not only the multilingual problem but also the multiparadigm problem. There is one common repository model for all languages. The process for converting procedural to object-oriented structures has been documented in a paper for the 18th International Conference on Software Maintenance. (see Figure 5: Merging program structures into a common model)

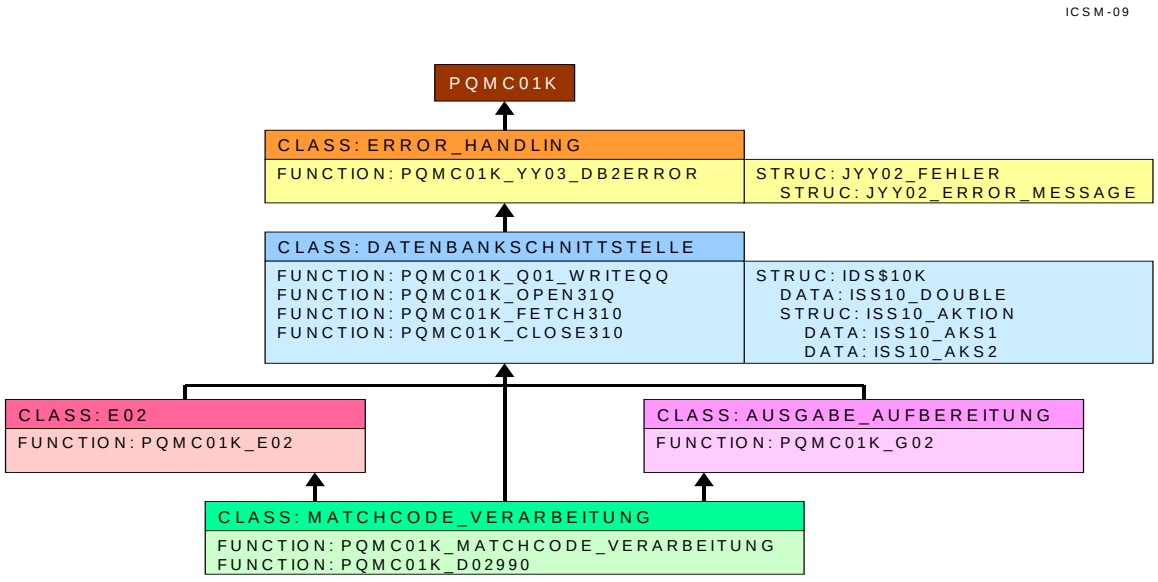


Figure 5: Derived Class Structure

6. SoftRedoc Documentation Samples

6.1. Sample Macro Table

Lnr Lng Code Name
0001 04 PROC NOPR
0002 07 LINK OBJNAME
0003 06 FUNC YABVAN
0004 07 COPY YABZBER
0005 07 PROC YAD1401
0006 08 FUNC YADRESS1
0007 08 FUNC YADRESSE
0008 08 LINK YAGRCALL
0009 08 LINK YAGRCALT
0010 05 DB YAIBD
0011 04 FUNC YAKV
0012 05 CNTR YALIN
0013 07 FUNC YANREDE
0014 04 FUNC YAUS
0015 05 FUNC YBAUS
0016 05 FUNC YBEIN
0017 06 FUNC YBEIN1
0018 05 PROC YBMIX
0019 04 FUNC YBOP
0020 05 FUNC YBPPZ
0021 07 PROC YBTCALL
0022 05 CALL YCALL
0023 05 FUNC YCAUS
0024 05 FUNC YCEIN
0025 04 FUNC YCLC
0026 05 FUNC YCLEN
0027 05 FILE YCLOS
0028 05 FUNC YCMIX
0029 07 LINK YCOCALL
0030 06 COPY YCOMRG
0031 07 COPY YCOMRG2
0032 06 ENTR YCSANF
0033 06 EXIT YCSEND
0034 07 FUNC YCTIME0
0035 07 FUNC YCTIME1
0036 05 FUNC YDATE
0037 06 ENTR YDCANF
0038 04 FILE YDCB
0039 06 EXIT YDCEND
0040 07 FUNC YDFTRAC
0041 04 FUNC YDIV
0042 07 MASK YDLCALL
0043 06 DB YDLPCB

6.2. Sample Function Table

Lnr Lng Code Name
0001 10 FILE open
0002 10 FILE Open
0003 10 FILE Connect
0004 10 FILE FileStream
0005 10 FILE OpenWrite
0006 10 FILE OpenFile

0007 10 FILE BhBelegSAPFile
0008 10 FILE BhBeleg
0009 00 FILE Disconnect
0010 00 FILE close
0011 00 FILE Close
0012 00 FILE CloseFile
0013 06 READ Read
0014 06 READ ExecuteXmlReader
0015 06 READ ExecuteXmlTextReader
0016 06 READ ReadXml
0017 06 READ ReadLine
0018 06 READ ReadByte
0019 06 READ ReadtoEnd
0020 06 READ StreamReader
0021 04 INPT GetData
0022 04 INPT Seek
0023 04 INPT Load
0024 07 OUTP Transform
0025 07 OUTP SaveAs
0026 07 WRIT makeProtokoll
0027 07 WRIT StreamWriter
0028 07 WRIT Write
0029 07 WRIT WriteLine
0030 07 WRIT WriteXmlSchema
0031 07 WRIT WriteThru
0032 07 WRIT WriteXml
0033 07 WRIT writeSap
0034 07 OUTP Copy
0035 07 OUTP Flush
0036 07 OUTP AppendText
0037 07 OUTP Create
0038 07 OUTP Move
0039 07 OUTP Info
0040 07 OUTP Warn
0041 07 OUTP Error
0042 07 OUTP Fatal
0043 06 GET ReadMitglied

6.3. Procedural Program Interfaces

```
/ Post Documentation of Program:COBOLD.CBL                                  Date: 09.02.18 |
+-------------------------------------------------------------------------------------------+
| Interface  Structure COBOLD.CBL                                                           |
| This listing displays the interfaces of the program to the external environment.          |
| These are the sub program calls (CALL), the TP operations (CICS/IMSDC), the DB operations|
|(DLI/SQL) and the file accesses as well as the COPY, INCLUDE and MAcro references.        |
+-------------------------------------------------------------------------------------------+
+-----+------+--------+-----------------------------------------------------------------+
| Line| Type | Access | Interface Statement                                             |
+-----+------+--------+-----------------------------------------------------------------+
|00221|      |ENTRY   |  COBOLD                                                         |
+-----+------+--------+-----------------------------------------------------------------+
|00224|      |        |INITIALIZATION.                                                  |
|00234|OUTPUT|WRITE   |    PRT-LINE                                                     |
|00235|OUTPUT|DISPLAY |    MESSAGE                                                      |
+-----+------+--------+-----------------------------------------------------------------+
|00239|OUTPUT|        |READ-ORDERS.                                                     |
|00240|INPUT |READ    |    ORDER-FILE                                                   |
+-----+------+--------+-----------------------------------------------------------------+
|00244|INPUT |        |READ-CUSTOMER.                                                   |
|00248|INPUT |READ    |    CUSTOMER-FILE                                                |
+-----+------+--------+-----------------------------------------------------------------+
|00257|      |        |PROCESS-ORDER.                                                   |
|00270|INPUT |READ    |    ARTICLE-FILE                                                 |
```

```
|00281|OUTPUT|REWRITE |    ARTICLE-RECORD                                                    |
+-----+------+--------+----------------------------------------------------------------------+
|00285|      |        |WRITE-OPEN-POSITIONS.                                                 |
|00291|OUTPUT|WRITE   |    POSITION-RECORD                                                   |
+-----+------+--------+----------------------------------------------------------------------+
|00295|      |        |WRITE-DISPATCH.                                                       |
|00310|OUTPUT|WRITE   |    DISPATCH-RECORD                                                   |
+-----+------+--------+----------------------------------------------------------------------+
|00312|OUTPUT|        |REPORT-ORDER.                                                         |
|00331|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00337|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00339|OUTPUT|WRITE   |    PRT-LINE                                                          |
+-----+------+--------+----------------------------------------------------------------------+
|00343|      |        |REPORT-ERROR.                                                         |
|00357|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00364|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00368|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00374|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00376|OUTPUT|WRITE   |    PRT-LINE                                                          |
+-----+------+--------+----------------------------------------------------------------------+
|00383|      |        |PRINT-SUMMARY.                                                        |
|00392|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00398|OUTPUT|WRITE   |    PRT-LINE                                                          |
|00400|OUTPUT|WRITE   |    PRT-LINE                                                          |
+-----+------+--------+----------------------------------------------------------------------+
|00405|      |        |TERMINATION.                                                          |
|00406|OUTPUT|DISPLAY |    MESSAGE                                                           |
+-----+------+--------+----------------------------------------------------------------------+
```

6.4. Procedural Program Structure

```
/ Post Documentation of Program:PQMC01K.CBL                                  Date: 09.02.18 |
+-------------------------------------------------------------------------------------------+
| Procedural Structure PQMC01K.CBL                                                          |
| This listing shows the overall structure of the program with its sections and procedures |
|as well as the branches - Performs, Calls and GOTOs - within the procedures.               |
+-------------------------------------------------------------------------------------------+
+-----+--+--------+-----------------------------------------------------------------------+
| Line|Nr| Type   |  Label    Calls/Performs/GOTOs                                        |
+-----+--+--------+-----------------------------------------------------------------------+
|02406| 0|MODULE  |PQMC01K                                                                |
|02087| 0|ENTRY   |PROCEDURE DIVISION.                                                    |
+-----+--+--------+-----------------------------------------------------------------------+
|02110| 1|SECT    |EINGABE-PRUEFUNG SECTION                                               |
+-----+--+--------+--+--------------------------------------------------------------------+
+-----+--+--------+--+--------------------------------------------------------------------+
|02114| 2|PARA    |  |B02                                                                 |
|02128| 3|CALL    |  |   'UDAS01K' USING PZ-BEREICH                                       |
|02137| 3|CALL    |  |   'UDAS01K' USING PZ-BEREICH                                       |
+-----+--+--------+--+--------------------------------------------------------------------+
|02153| 2|PARA    |  |B02200                                                              |
+-----+--+--------+--+--------------------------------------------------------------------+
|02185| 2|PARA    |  |B02990                                                              |
+-----+--+--------+-----------------------------------------------------------------------+
|02213| 1|SECT    |MATCHCODE-VERARBEITUNG SECTION                                         |
+-----+--+--------+--+--------------------------------------------------------------------+
|02224| 3|PERFORM |  |   E02                                                              |
|02253| 3|PERFORM |  |   R01-OPEN310                                                      |
|02257| 3|PERFORM |  |   R01-FETCH310                                                     |
|02274| 3|PERFORM |  |   R01-CLOSE310                                                     |
|02288| 3|PERFORM |  |   G02                                                              |
|02302| 3|PERFORM |  |   R01-FETCH310                                                     |
|02313| 3|PERFORM |  |   R01-CLOSE310                                                     |
|02315| 3|PERFORM |  |   G02                                                              |
|02319| 3|PERFORM |  |   G02                                                              |
+-----+--+--------+--+--------------------------------------------------------------------+
|02341| 2|PARA    |  |D02990                                                              |
+-----+--+--------+-----------------------------------------------------------------------+
|02371| 1|SECT    |E02 SECTION                                                            |
+-----+--+--------+--+--------------------------------------------------------------------+
+-----+--+--------+--+--------------------------------------------------------------------+
|02372| 2|PARA    |  |E02100                                                              |
+-----+--+--------+--+--------------------------------------------------------------------+
|02428| 2|PARA    |  |E02200                                                              |
|02438| 3|CALL    |  |   'UDAS06K' USING                                                  |
```

```
|02438| 3|        |  |   JQAS06K                                                                |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02473| 2|PARA    |  |E02300                                                                    |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02511| 2|PARA    |  |E02900                                                                    |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02522| 2|PARA    |  |E02990                                                                    |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02545| 1|SECT    |AUSGABE-AUFBEREITUNG SECTION                                                 |
+-----+--+--------+--+--------------------------------------------------------------------------+
+-----+--+--------+--+--------------------------------------------------------------------------+
|02549| 2|PARA    |  |G02                                                                       |
|02562| 3|PERFORM |  |   Q01-WRITEQ                                                             |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02599| 2|PARA    |  |G02800                                                                    |
|02607| 3|PERFORM |  |   Q01-WRITEQ                                                             |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02614| 2|PARA    |  |G02900                                                                    |
|02634| 3|PERFORM |  |   Q01-WRITEQ                                                             |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02636| 2|PARA    |  |G02990                                                                    |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02668| 1|SECT    |DATENBANKSCHNITTSTELLE SECTION                                               |
+-----+--+--------+--+--------------------------------------------------------------------------+
+-----+--+--------+--+--------------------------------------------------------------------------+
|02672| 2|PARA    |  |DB-INITIALISIERUNG                                                        |
|02679| 3|SQL     |  |   SET                                                                    |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02683| 2|PARA    |  |Q01-WRITEQ                                                                |
|02692| 3|PERFORM |  |   R01-COUNT                                                              |
|02714| 3|CALL    |  |   'TSQUEUE' USING MQMC02K                                                |
|02725| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
|02729| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02737| 2|PARA    |  |R01-OPEN310                                                               |
|02741| 3|SQL     |  |   OPEN                                                                   |
|02752| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02757| 2|PARA    |  |R01-CLOSE310                                                              |
|02761| 3|SQL     |  |   CLOSE                                                                  |
|02769| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02774| 2|PARA    |  |R01-FETCH310                                                              |
|02788| 3|SQL     |  |   FETCH                                                                  |
|02796| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02801| 2|PARA    |  |R01-COUNT                                                                 |
|02814| 3|CALL    |  |   'PQTA09K' USING IQTA01K.                                               |
|02832| 3|SQL     |  |   SELECT                                                                 |
|02844| 3|PERFORM |  |   YY03-DB2ERROR                                                          |
+-----+--+--------+--+--------------------------------------------------------------------------+
|02848| 2|PARA    |  |YY03-DB2ERROR                                                             |
|02853| 3|COPY    |  |   RDXX5100                                                               |
|02854| 3|COPY    |  |   RDYY0101                                                               |
|02855| 3|COPY    |  |   FUER                                                                   |
|02856| 3|COPY    |  |   RDYY0300                                                               |
+-----+--+--------+--+--------------------------------------------------------------------------+
```

6.5. Procedural Decision Logic

```
/ Post Documentation of Program:PQMC01K.CBL                                  Date: 09.02.18 |
+-----------------------------------------------------------------------------------------+
| Decisional Structure PQMC01K.CBL                                                        |
| This listing displays the decision logic of the procedures in pseudo code format showing |
| the conditions, label and branches as well  as the procedural comments.                 |
| @ = indicates that this is a selection                                                  |
| * = indicates that this is a repetition                                                 |
+-----------------------------------------------------------------------------------------+
| Lev Line Label Decision & Code Statements                                               |
+-----------------------------------------------------------------------------------------+
+--+-----+-------+-------------------------------------------------------------------------+
| 0|02087|Label  |PQMC01K                                                                  |
+--+-----+-------+-------------------------------------------------------------------------+
| 0|02110|Label  |EINGABE-PRUEFUNG SECTION                                                 |
+--+-----+-------+-------------------------------------------------------------------------+
```

```
| 0|02114|Label  |B02                                                                    |
+--+-----+-------+-----------------------------------------------------------------------+
| 1|02116|       |MOVE SPACES TO NEXT-METHOD-ID.                                         |
| 1|02117|       |MOVE MMC02-QGRDNR2 TO WMC01-GRDNR8.                                    |
+--+-----+-------+-----------------------------------------------------------------------+
| 1|02118|If     |IF WMC01-GRDNR7 NOT NUMERIC                                            |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02120|       |   |@MOVE 'B02200' TO NEXT-METHOD-ID                                   |
+--+-----+-------+---+-------------------------------------------------------------------+
| 1|02121|End    |END-IF                                                                 |
+--+-----+-------+-----------------------------------------------------------------------+
| 1|02123|If     |IF NEXT-METHOD-ID = SPACES                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02124|       |   |@MOVE WMC01-GRDNR8 TO J01-QGRDNRI                                  |
| 2|02125|       |   |@MOVE '2' TO J01-QPZACOD                                           |
| 2|02126|       |   |@CALL 'UDAS01K' USING PZ-BEREICH                                   |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02128|If     |   |IF J01-QPZRCOD NOT = '0'                                           |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02129|       |   |   |@MOVE 'AU029' TO WAS00-QMELDNR                                 |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02132|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02132|       |   |@MOVE WMC01-GRDNR8 TO J01-QGRDNRI                                  |
| 2|02133|       |   |@MOVE '1' TO J01-QPZACOD                                           |
| 2|02135|       |   |@CALL 'UDAS01K' USING PZ-BEREICH                                   |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02137|If     |   |IF J01-QPZRCOD NOT = '0'                                           |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02138|       |   |   |@MOVE 'NV020' TO WAS00-QMELDNR                                 |
| 3|02139|       |   |   |@MOVE -1 TO MMC02-QGRDNR2L                                     |
| 3|02140|       |   |   |@MOVE '1' TO INV01-QFEHLER                                     |
| 3|02142|       |   |   |@MOVE 'B02990' TO NEXT-METHOD-ID                               |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02144|Else   |   |ELSE                                                               |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02144|       |   |   |@MOVE J01-QGRDNR TO INV01-QGRDNR                               |
| 3|02145|       |   |   |@MOVE ZERO TO INV01-QUKNR                                      |
| 3|02146|       |   |   |@MOVE '99' TO INV01-QSTEUB                                     |
| 3|02147|       |   |   |@MOVE '1' TO IAU02-NEUER-KEY                                   |
| 3|02149|       |   |   |@MOVE 'B02990' TO NEXT-METHOD-ID                               |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02151|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 1|02151|End    |END-IF                                                                 |
+--+-----+-------+-----------------------------------------------------------------------+
| 0|02153|Label  |B02200                                                                 |
+--+-----+-------+-----------------------------------------------------------------------+
| 1|02155|If     |IF NEXT-METHOD-ID = 'B02200'                                           |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02157|If     |   |IF MMC02-QMCNAME1L NOT = ZERO                                      |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02158|       |   |   |@MOVE ISS02-QATRP TO MMC02-QMCNAME1A                           |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02161|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02161|If     |   |IF MMC02-QMCNAME1 NOT = IAU02-QNAME1                               |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02162|       |   |   |@MOVE MMC02-QMCNAME1 TO IAU02-QNAME1                           |
| 3|02163|       |   |   |@MOVE '1' TO WMC01-NEUER-KEY                                   |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02166|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02166|If     |   |IF MMC02-QMCVORN1L NOT = ZERO                                      |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02167|       |   |   |@MOVE ISS02-QATRP TO MMC02-QMCVORN1A                           |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02170|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02170|If     |   |IF MMC02-QMCVORN1 NOT = IAU02-QVORN1                               |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 3|02171|       |   |   |@MOVE MMC02-QMCVORN1 TO IAU02-QVORN1                           |
| 3|02172|       |   |   |@MOVE '1' TO WMC01-NEUER-KEY                                   |
+--+-----+-------+---+---+---------------------------------------------------------------+
| 2|02175|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02175|If     |   |IF MMC02-QMCPLZ1L NOT = ZERO                                       |
+--+-----+-------+---+---+---------------------------------------------------------------+
```

```
| 3|02176|       |   |   |@MOVE ISS02-QATRP TO MMC02-QMCPLZ1A                                |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 2|02179|End    |   |END-IF                                                                 |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02179|If     |   |IF MMC02-QMCPLZ1 NOT = IAU02-QPLZ1                                     |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 3|02180|       |   |   |@MOVE MMC02-QMCPLZ1 TO IAU02-QPLZ1                                 |
| 3|02181|       |   |   |@MOVE '1' TO WMC01-NEUER-KEY                                       |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 2|02183|End    |   |END-IF                                                                 |
+--+-----+-------+---+-------------------------------------------------------------------+
| 1|02183|End    |END-IF                                                                     |
+--+-----+-------+-------------------------------------------------------------------+
| 0|02185|Label  |B02990                                                                     |
+--+-----+-------+-------------------------------------------------------------------+
| 1|02186|       |MOVE SPACES TO NEXT-METHOD-ID                                              |
| 1|02187|       |EXIT.                                                                      |
+--+-----+-------+-------------------------------------------------------------------+
| 0|02213|Label  |MATCHCODE-VERARBEITUNG SECTION                                             |
+--+-----+-------+-------------------------------------------------------------------+
| 1|02219|       |MOVE SPACES TO NEXT-METHOD-ID.                                             |
| 1|02220|       |MOVE ZERO TO IAU02-QBLFDNR                                                 |
| 1|     |       |IAU02-MAPRUECK.                                                            |
| 1|02224|       |PERFORM E02.                                                               |
+--+-----+-------+-------------------------------------------------------------------+
| 1|02226|If     |IF INV01-QFEHLER NOT = '0'                                                 |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02228|       |   |@MOVE 'D02990' TO NEXT-METHOD-ID                                   |
+--+-----+-------+---+-------------------------------------------------------------------+
| 1|02229|End    |END-IF                                                                     |
+--+-----+-------+-------------------------------------------------------------------+
| 1|02233|If     |IF NEXT-METHOD-ID = SPACES                                                 |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02234|       |   |@MOVE J06-QMCNAME TO WMC01-MCNAME                                  |
| 2|02235|       |   |@MOVE J06-QMCVORN TO WMC01-MCVORN                                  |
| 2|02236|       |   |@MOVE J06-QMCPLZ TO WMC01-MCPLZ                                    |
| 2|02237|       |   |@MOVE J06-QMCNAM1 TO WMC01-MCNAM1                                  |
| 2|02238|       |   |@MOVE J06-QMCVOR1 TO WMC01-MCVOR1                                  |
| 2|02239|       |   |@MOVE J06-QMCPLZ1 TO WMC01-MCPLZ1                                  |
| 2|02243|       |   |@PERFORM R01-OPEN310                                               |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02253|Loop   |   |PERFORM UNTIL NEXT-METHOD-ID NOT = SPACES                          |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 3|02255|       |   |   |@*PERFORM R01-FETCH310                                         |
| 3|02257|       |   |   |@*MOVE SPACES TO NEXT-METHOD-ID                                |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 3|02258|If     |   |   |IF SQLCODE = ZERO                                              |
| 3|     |       |   |   |AND INV01-QAKS1(2) = '02'                                      |
| 3|     |       |   |   |AND S310-MCBKZ NOT = 'A'                                       |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 4|02262|       |   |   |   |@*CONTINUE                                                 |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 3|02264|Else   |   |   |ELSE                                                           |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 4|02264|       |   |   |   |@*MOVE 'D02360' TO NEXT-METHOD-ID                          |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 3|02266|End    |   |   |END-IF                                                         |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 2|02267|End    |   |END-LOOP                                                               |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02267|       |   |@MOVE SPACES TO NEXT-METHOD-ID                                     |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02272|If     |   |IF SQLCODE = 100                                                   |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 3|02273|       |   |   |@PERFORM R01-CLOSE310                                          |
| 3|02274|       |   |   |@MOVE 'NV057' TO WAS00-QMELDNR                                 |
| 3|02276|       |   |   |@MOVE -1 TO MMC02-QMCNAME1L                                    |
| 3|02277|       |   |   |@MOVE '1' TO INV01-QFEHLER                                     |
| 3|02278|       |   |   |@MOVE ZERO TO WAS00-FFMAP                                      |
| 3|02280|       |   |   |@MOVE 'D02990' TO NEXT-METHOD-ID                               |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 2|02283|End    |   |END-IF                                                             |
+--+-----+-------+---+-------------------------------------------------------------------+
| 2|02283|If     |   |IF NEXT-METHOD-ID = SPACES                                         |
+--+-----+-------+---+---+-------------------------------------------------------------------+
| 3|02284|       |   |   |@ADD 1 TO WMC01-NUM                                            |
| 3|02286|       |   |   |@PERFORM G02                                                   |
| 3|02288|       |   |   |@MOVE 16 TO IAU02-NUM                                          |
```

```
+--+-----+-------+---+---+-----------------------------------------------------------------------+
| 3|02298|Loop   |   |   |PERFORM UNTIL NEXT-METHOD-ID NOT = SPACES                              |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 4|02300|       |   |   |   |@*PERFORM R01-FETCH310                                             |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 4|02302|If     |   |   |   |IF SQLCODE = ZERO                                                  |
| 4|     |       |   |   |   |AND INV01-QAKS1(2) = '02'                                          |
| 4|     |       |   |   |   |AND S310-MCBKZ NOT = 'A'                                           |
+--+-----+-------+---+---+---+---+---------------------------------------------------------------+
| 5|02306|       |   |   |   |   |@*CONTINUE                                                     |
+--+-----+-------+---+---+---+---+---------------------------------------------------------------+
| 4|02311|Else   |   |   |   |ELSE                                                               |
+--+-----+-------+---+---+---+---+---------------------------------------------------------------+
| 5|02311|If     |   |   |   |   |IF SQLCODE = 100                                               |
+--+-----+-------+---+---+---+---+---+-----------------------------------------------------------+
| 6|02312|       |   |   |   |   |   |@*PERFORM R01-CLOSE310                                     |
| 6|02313|       |   |   |   |   |   |@*MOVE 1 TO WMC01-BSP-WECHSEL                              |
| 6|02314|       |   |   |   |   |   |@*PERFORM G02                                              |
+--+-----+-------+---+---+---+---+---+-----------------------------------------------------------+
| 5|02316|Else   |   |   |   |   |ELSE                                                           |
+--+-----+-------+---+---+---+---+---+-----------------------------------------------------------+
| 6|02316|If     |   |   |   |   |   |IF WMC01-NUM NOT > IAU02-NUM                           |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 7|02318|       |   |   |   |   |   |   |@*PERFORM G02                                      |
| 7|02319|       |   |   |   |   |   |   |@*ADD 1 TO WMC01-NUM                               |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02322|End    |   |   |   |   |   |END-IF                                                 |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02322|If     |   |   |   |   |   |IF INV01-QFEHLER NOT = '0'                             |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 7|02324|       |   |   |   |   |   |   |@*MOVE 'D02990' TO NEXT-METHOD-ID                  |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02327|End    |   |   |   |   |   |END-IF                                                 |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02327|If     |   |   |   |   |   |IF WMC01-BSP-WECHSEL = 1                               |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 7|02329|       |   |   |   |   |   |   |@*MOVE 'D02990' TO NEXT-METHOD-ID                  |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02332|Else   |   |   |   |   |   |ELSE                                                   |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 7|02332|       |   |   |   |   |   |   |@*CONTINUE                                         |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 6|02334|End    |   |   |   |   |   |END-IF                                                 |
+--+-----+-------+---+---+---+---+---+---+-------------------------------------------------------+
| 5|02335|End    |   |   |   |   |END-IF                                                         |
+--+-----+-------+---+---+---+---+---+-------------------------------------------------------+
| 4|02336|End    |   |   |   |END-IF                                                             |
+--+-----+-------+---+---+---+---+-----------------------------------------------------------+
| 3|02338|End    |   |   |END-LOOP                                                               |
+--+-----+-------+---+---+---+-------------------------------------------------------------------+
| 2|02339|End    |   |END-IF                                                                     |
+--+-----+-------+---+---+-----------------------------------------------------------------------+
| 1|02339|End    |END-IF                                                                         |
+--+-----+-------+-------------------------------------------------------------------------------+
```

6.6. Procedural Data Flow Table

```
/ Post Documentation of Program:COBOLD.CBL                                  Date: 09.02.18 |
+-------------------------------------------------------------------------------------------+
| Data Flow  Structure COBOLD.CBL                                                           |
| This table displays for each program procedure, i.e. Section or Paragraph, the input,    |
|output and control variables - predicates -  in the form of a HIPO diagram.               |
+-------------------------------------------------------------------------------------------+
+-----------------------------+---------------------------+-----------------------------+
|  PROCEDURE/METHOD: INITIALIZATION                                                       |
+-----------------------------+---------------------------+-----------------------------+
|  INPUTS/ARGUMENTS           |  PREDICATES               | OUTPUTS/RESULTS             |
+-----------------------------+---------------------------+-----------------------------+
|PAGE-COUNT                   |                           |PAGE-COUNT                   |
|HEADER-PRT-LINE              |                           |PAGE-COUNT-NO                |
|                             |                           |PRT-LINE                     |
+-----------------------------+---------------------------+-----------------------------+
+-----------------------------+---------------------------+-----------------------------+
|  PROCEDURE/METHOD: READ-CUSTOMER                                                        |
+-----------------------------+---------------------------+-----------------------------+
|  INPUTS/ARGUMENTS           |  PREDICATES               | OUTPUTS/RESULTS             |
```

ORDER-RECORD.CUST-NO		ERROR-TYPE
		CUST-KEY
		POS
		TOTAL-ITEMS-FULFILLED
		TOTAL-CUST-PRICE

PROCEDURE/METHOD: PROCESS-ORDER		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
ORDER-RECORD.ITEM-QUAN (POS)	POS	ERROR-TYPE
	ORDER-RECORD.ITEM-NO (POS)	POS
	ORDER-RECORD.ITEM-QUAN (POS)	ARTICLE-RECORD.ART-QUAN
	ARTICLE-RECORD.ART-QUAN	

PROCEDURE/METHOD: WRITE-OPEN-POSITIONS		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
ORDER-RECORD.CUST-NO		POSITION-RECORD.CUST-NO
ORDER-RECORD.ORDER-NO		POSITION-RECORD.ORDER-NO
ORDER-RECORD.ORDER-ITEM (POS)		POSITION-RECORD.DISPATCH-ITEM
		POSITION-RECORD.REC-TYPE

PROCEDURE/METHOD: WRITE-DISPATCH		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
ORDER-RECORD.ORDER-NO	CUST-CREDIT	TOTAL-ITEMS-FULFILLED
ORDER-RECORD.CUST-NO		DISPATCH-RECORD.ORDER-NO
CUSTOMER-RECORD.CUST-NAME		DISPATCH-RECORD.CUST-NO
CUSTOMER-RECORD.CUST-ADDRESS		DISPATCH-RECORD.CUST-NAME
ORDER-RECORD.ORDER-ITEM (POS)		DISPATCH-RECORD.CUST-ADDRESS
		DISPATCH-RECORD.DISPATCH-ITEM
		PAYMENT
		DISPATCH-RECORD.REC-TYPE

PROCEDURE/METHOD: REPORT-ORDER		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
CUSTOMER-RECORD.CUST-NO	ERROR-TYPE	DATA-PRT-LINE
CUSTOMER-RECORD.CUST-NAME	LINE-COUNT	DATA-PRT-LINE.CUST-NO
ARTICLE-RECORD.ART-NO		DATA-PRT-LINE.CUST-NAME
ARTICLE-RECORD.ART-NAME		DATA-PRT-LINE.ART-NO
ORDER-RECORD.ITEM-QUAN (POS)		DATA-PRT-LINE.ART-NAME
ARTICLE-RECORD.ART-PRICE		DATA-PRT-LINE.ART-QUAN
ORDER-RECORD.ITEM-QUAN (POS)		DATA-PRT-LINE.ART-PRICE
PRICE		ARTICLE-RECORD.ART-PRICE
DATA-PRT-LINE		PRICE
PAGE-COUNT		TOTAL-PRICE
HEADER-PRT-LINE		TOTAL-CUST-PRICE
UNDER-LINE		PRT-LINE
		LINE-COUNT
		PAGE-COUNT
		PAGE-COUNT-NO

PROCEDURE/METHOD: REPORT-ERROR		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
ORDER-RECORD.ORDER-NO	ERROR-TYPE	DATA-PRT-LINE
ORDER-RECORD.CUST-NO	LINE-COUNT	ERROR-PRT-LINE
ORDER-RECORD.ART-NO (POS)		DATA-PRT-LINE.ORDER-NO
ARTICLE-RECORD.ART-NAME		DATA-PRT-LINE.CUST-NO
ARTICLE-RECORD.ART-QUAN		DATA-PRT-LINE.ART-NO
DATA-PRT-LINE		DATA-PRT-LINE.ART-NAME
ERROR-PRT-LINE		DATA-PRT-LINE.ART-QUAN
PAGE-COUNT		PRT-LINE
HEADER-PRT-LINE		ERROR-MESSAGE
UNDER-LINE		LINE-COUNT
		PAGE-COUNT

		PAGE-COUNT-NO
PROCEDURE/METHOD: PRINT-SUMMARY		
INPUTS/ARGUMENTS	PREDICATES	OUTPUTS/RESULTS
CUSTOMER-RECORD.CUST-NO	LINE-COUNT	SUMMARY-PRT-LINE.CUST-NO
CUSTOMER-RECORD.CUST-NAME		SUMMARY-PRT-LINE.CUST-NAME
POS		NO-ITEMS-ORDERED
TOTAL-ITEMS-FULFILLED		NO-ITEMS-FULFILLED
TOTAL-CUST-PRICE		TOTAL-PRICE-FOR-CUST
SUMMARY-PRT-LINE		PRT-LINE
PAGE-COUNT		LINE-COUNT
HEADER-PRT-LINE		PAGE-COUNT
UNDER-LINE		PAGE-COUNT-NO

6.7. Procedural Data List

/ Post Documentation of Program:COBOLD.CBL Date: 09.02.18

Used Data Structure COBOLD.CBL

This listing contains the declarations of all data structures, groups and individual data elements and indicates how they are used, either as inputs, outputs, both or not at all.

Usage	Lev		Data	Type	Comment
UNUSED	01	01	CUSTOMER-RECORD.		
OUTPUT	02	02	REC-TYPE	PIC XX.	
INOUT	02	02	CUST-NO	PIC 9(8) COMP.	
INOUT	02	02	CUST-NAME	PIC X(20).	
INOUT	02	02	CUST-ADDRESS.		
INPUT	02	02	CUST-CREDIT	PIC 9.	
INPUT	01	01	ARTICLE-RECORD.		
OUTPUT	02	02	REC-TYPE	PIC XX.	
INOUT	02	02	ART-NO	PIC 9(8) COMP.	
OUTPUT	02	02	ART-QUAN	PIC S9(5) USAGE COMP.	
OUTPUT	02	02	ART-PRICE	PIC S999V99 COMP-3.	
INOUT	02	02	ART-NAME	PIC X(40).	
INOUT	01	01	ORDER-RECORD.		
OUTPUT	02	02	REC-TYPE	PIC XX.	
INOUT	02	02	ORDER-NO	PIC 9(8) COMP.	
INOUT	02	02	CUST-NO	PIC 9(8) COMP.	
INOUT	02	02	ORDER-ITEMS.		
INOUT	03	03	ORDER-ITEM	OCCURS 9 TIMES.	
INPUT	04	04	ITEM-NO	PIC 9(3).	
INOUT	04	04	ART-NO	PIC 9(8) COMP.	
INPUT	04	04	ITEM-QUAN	PIC 9(5) USAGE COMP.	
INPUT	01	01	DISPATCH-RECORD.		
OUTPUT	02	02	REC-TYPE	PIC XX.	
INOUT	02	02	ORDER-NO	PIC 9(8) COMP.	

```
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NO                    PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NAME                  PIC X(20).                                |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-ADDRESS.                                                        |
+------+---+-----------------------------------------------------------------------------+
|UNUSED| 02|    02  ORDER-ITEMS.                                                         |
+------+---+-----------------------------------------------------------------------------+
|UNUSED| 03|        03  DISPATCH-ITEM.                                                   |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 04|            04  ITEM-NO            PIC 9(3).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 04|            04  ART-NO             PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 04|            04  ITEM-QUAN          PIC 9(5) USAGE COMP.                      |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  PAYMENT                    PIC X.                                    |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 01|01  POSITION-RECORD.                                                         |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  REC-TYPE                   PIC XX.                                   |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  ORDER-NO                   PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NO                    PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  ORDER-ITEMS.                                                         |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 03|        03  DISPATCH-ITEM.                                                   |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 04|            04  ITEM-NO            PIC 9(3).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 04|            04  ART-NO             PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 04|            04  ITEM-QUAN          PIC 9(5) USAGE COMP.                      |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 01|01  PRT-LINE                       PIC X(133).                               |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 77|77  CUST-KEY                       PIC 9(8) COMP.                            |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 77|77  POS                            PIC 99 USAGE COMP.                        |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 77|77  ERROR-TYPE                     PIC 99.                                   |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 77|77  PAGE-COUNT                     PIC 999 COMP VALUE 0.                     |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 77|77  LINE-COUNT                     PIC 99 COMP VALUE 0.                      |
+------+---+-----------------------------------------------------------------------------+
|INPUT | 77|77  PRICE                          PIC 99999V99.                             |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 77|77  TOTAL-CUST-PRICE               PIC 99999V99.                             |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 77|77  TOTAL-ITEMS-FULFILLED          PIC 99 COMP VALUE 0.                      |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 01|01  HEADER-PRT-LINE.                                                         |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  PAGE-COUNT-NO              PIC 9999.                                 |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 01|01  UNDER-LINE.                                                              |
+------+---+-----------------------------------------------------------------------------+
|UNUSED| 01|01  DATA-PRT-LINE.                                                           |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  ORDER-NO                   PIC 9(8).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NO                    PIC 9(8).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NAME                  PIC X(20).                                |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  ART-NO                     PIC 9(8).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  ART-NAME                   PIC X(20).                                |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  ART-QUAN                   PIC 9(8).                                 |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  ART-PRICE                  PIC ZZZZ9.99.                             |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  TOTAL-PRICE                PIC ZZZZ9.99.                             |
```

```
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 01|01  ERROR-PRT-LINE.                                                          |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  ERROR-MESSAGE              PIC X(40).                                |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 01|01  SUMMARY-PRT-LINE.                                                        |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NO                    PIC 9(8).                                 |
+------+---+-----------------------------------------------------------------------------+
|INOUT | 02|    02  CUST-NAME                  PIC X(20).                                |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  NO-ITEMS-ORDERED           PIC 99.                                   |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  NO-ITEMS-FULFILLED         PIC 99.                                   |
+------+---+-----------------------------------------------------------------------------+
|OUTPUT| 02|    02  TOTAL-PRICE-FOR-CUST       PIC ZZZZ9.99.                             |
+------+---+-----------------------------------------------------------------------------+
```

6.8. Class Includes/Imports

```
+-----------------------------------------------------------------------------+
|         S O F T R E D O C   M O D U L E   S P E C I F I C A T I O N           |
+-----------------------------------------------------------------------------+
| SOURCE TYPE:CSS                                          DATE: 26.09.05     |
|                                                                             |
| MODULE NAME:Aussendung                                 SYSTEM: WEBGUI       |
|                                                                             |
| SOURCE NAME:AussendungJob.aspx.cs                                           |
+-----------------------------------------------------------------------------+
| LINE :      INCLUDES/IMPORTS                                                |
+-----------------------------------------------------------------------------+
| 00001:using System;                                                         |
| 00002:using System.Collections;                                             |
| 00003:using System.ComponentModel;                                          |
| 00004:using System.Data;                                                    |
| 00005:using System.Drawing;                                                 |
| 00006:using System.Web;                                                     |
| 00007:using System.Web.SessionState;                                        |
| 00008:using System.Web.UI;                                                  |
| 00009:using System.Web.UI.WebControls;                                      |
| 00010:using System.Web.UI.HtmlControls;                                     |
| 00011:using InhouseWKOIT.Factory;                                           |
| 00012:using InhouseWKOIT.BusinessEntities;                                  |
| 00013:using InhouseWKOIT.BusinessEntities.AussendungsEntities;              |
| 00014:using InhouseWKOIT.BusinessEntities.Enumerations;                     |
| 00015:using InhouseWKOIT.UtilitiesFramework;                                |
| 00016:using InhouseWKOIT.GuWebClient.Library.ServerObjects.Common;          |
| 00017:using InhouseWKOIT.BusinessEntities.Messages;                         |
| 00018:using InhouseWKOIT.BusinessEntities.EinstufungsEntities;              |
| 00019:using InhouseWKOIT.BusinessEntities.FGEntities;                       |
| 00020:using Wko.Wsf.Declarative;                                            |
| 00021:using System.Configuration;                                           |
|                                                                             |
+-----------------------------------------------------------------------------+
```

6.9. Class Definitions

```
+-----------------------------------------------------------------------------+
| LINE : DEFINES
+-----------------------------------------------------------------------------+
| 00024:namespace InhouseWKOIT.GuWebClient.Aussendung                         |
| 00029:[WsfTargetMethodPermission(typeof(WKOIT.BizAction.Gu.ShowAussendung))]|
| 00139:#region Hauptfilter                                                   |
| 00182:#endregion                                                            |
| 00199:#region Protokoll                                                     |
| 00248:#endregion                                                            |
| 00260:#region Filelist                                                      |
| 00288:#endregion                                                            |
| 00454:#region Bereichswahl                                                  |
| 00461:#region Sparteneinschränkung                                          |
| 00489:#endregion                                                            |
```

```
| 00493:#region Fachgruppeneinschränkung                                                |
| 00519:#endregion                                                                      |
| 00523:#region Mitgliedsbereich                                                        |
| 00549:#endregion                                                                      |
| 00556:#region Alle Mitglieder wählen                                                  |
| 00561:#endregion                                                                      |
| 00566:#endregion                                                                      |
| 00622:#region Web Form Designer generated code                                        |
| 00645:#endregion                                                                      |
|                                                                                       |
+---------------------------------------------------------------------------------------+
```

6.10.Class List

```
+---------------------------------------------------------------------------------------+
| LINE : CLASSES                                                                        |
+---------------------------------------------------------------------------------------+
| 00031:public class AussendungJob : GuWebClient.ServerObjects.Common.BasePage         |
| 00031:{                                                                               |
|                                                                                       |
+---------------------------------------------------------------------------------------+
```

6.11.Class Methods

```
+---------------------------------------------------------------------------------------+
| LINE :  METHODS/FUNCTIONS                                                             |
+---------------------------------------------------------------------------------------+
|                                                                                       |
| 00078:private ShowAussendungAufrag fillAuftragsObject(AussendungsAuftrag auftrag)     |
|                                                                                       |
| 00094:private void displayJob(ShowAussendungAufragDetailResponseMessage response)     |
|                                                                                       |
| 00307:private void loadAussendungsJob(AussendungAuftragKey key)                       |
|                                                                                       |
| 00320:private void loadAussendungsJob(int selectedYear)                               |
|                                                                                       |
| 00334:private void Page_Load(object sender, System.EventArgs e)                       |
|                                                                                       |
| 00388:private void deleteJob()                                                        |
|                                                                                       |
| 00414:private void saveJob()                                                          |
|                                                                                       |
| 00623:override protected void OnInit(EventArgs e)                                     |
|                                                                                       |
| 00636:private void InitializeComponent()                                              |
|                                                                                       |
| 00648:private void saveJobButton_Click(object sender, System.EventArgs e)             |
|                                                                                       |
| 00653:private void deleteButton_Click(object sender, System.EventArgs e)              |
|                                                                                       |
| 00658:private void fileXml_CheckedChanged(object sender, System.EventArgs e)          |
|                                                                                       |
| 00663:private void detailAnsicht_Click(object sender, System.EventArgs e)             |
|                                                                                       |
+---------------------------------------------------------------------------------------+
```

6.12.Class Method References

```
+---------------------------------------------------------------------------------------+
| LINE : METHOD / FUNCTION REFERENCES                                                   |
+---------------------------------------------------------------------------------------+
| 00001:AussendungJob.aspx.cs();                                                        |
+-------------------------------------------------------------------------------------- |
| 00078:private ShowAussendungResponse AuftragsObject(AussendungsAuftrag auftrag)       |
| 00081:ShowAussendungAufragDetailRequestMessage                                        |
| 00084:BizAdapterFactory                                                               |
| 00085:factory.execute                                                                 |
+-------------------------------------------------------------------------------------- |
| 00094:private void displayJob(ShowAussendungAufragDetailResponseMessage response)     |
```

```
+------------------------------------------------------------------------------- |
| 00097:Options.getDateIndefinite                                                |
| 00097:response.Auftrag.WunschDurchfuehrung.ToShortDateString                   |
| 00097:response.Auftrag.IstDurchfuehrung.ToShortDateString                      |
| 00121:ListItem                                                                 |
| 00121:currentYear.ToString                                                     |
| 00123:verarbeitungsJahr.Items.Add                                              |
| 00130:response.Auftrag.AussendungsDatum.ToShortDateString                      |
| 00131:response.Auftrag.KontostandPer.ToShortDateString                         |
| 00132:response.Auftrag.AbzugVariabel.ToString                                  |
| 00135:response.Auftrag.VerzichtTyp.ToString                                    |
| 00178:areaString.Substring                                                     |
| 00185:response.Auftrag.FileTyp.IndexOf                                         |
| 00191:response.Auftrag.AussendungsTyp.ToString                                 |
| 00194:response.Auftrag.PfaendungsNummer.ToString                               |
| 00195:response.Auftrag.Bagatellgrenze.ToString                                 |
| 00196:response.Auftrag.Mahnspesen.ToString                                     |
| 00204:HtmlGenericControl                                                       |
| 00205:noProtocoll.Attributes.Add                                               |
| 00206:noProtocoll.Style.Add                                                    |
| 00210:protokollCell.Controls.Add                                               |
| 00226:liTag.Attributes.Add                                                     |
| 00239:liTag.Controls.Add                                                       |
| 00240:ulTag.Controls.Add                                                       |
| 00253:response.Auftrag.EinstufungsAuftragKey.keyToString                       |
| 00266:liTag.Style.Add                                                          |
| 00268:HtmlAnchor                                                               |
| 00273:fileList.Controls.Add                                                    |
| 00280:noFiles.Attributes.Add                                                   |
| 00281:noFiles.Style.Add                                                        |
+------------------------------------------------------------------------------- |
| 00307:private void loadAussendungsJob(AussendungAuftragKey key)                |
+------------------------------------------------------------------------------- |
| 00310:AussendungsAuftrag                                                       |
| 00312:fillAuftragsObject                                                       |
| 00315:errorWebHandler.publish                                                  |
| 00318:displayJob                                                               |
|------------------------------------------------------------------------------- |
| 00320:private void loadAussendungsJob(int selectedYear)                        |
|------------------------------------------------------------------------------- |
| 00323:AussendungsAuftrag                                                       |
| 00325:fillAuftragsObject                                                       |
| 00328:errorWebHandler.publish                                                  |
| 00331:displayJob                                                               |
|                                                                                |
+--------------------------------------------------------------------------------+
```

6.13.Class Interfaces

```
+--------------------------------------------------------------------------------+
|   FUNCTION INTERFACES:                                                         |
|     ---------------------------------------------------------------------- |
|==>> GveController.java ();                                                     |
|     +---------------------------------------------------------------------- |
|==>> public class GveController implements GveAbgfDao {                        |
|     GveAbgfDao                                                                 |
|     +---------------------------------------------------------------------- |
|==>> public class GSVController implements GSVAbgfDao {                        |
|     GSVAbgfDao                                                                 |
|                                                                                |
+--------------------------------------------------------------------------------+
```

6.14.Class Attributes

```
+--------------------------------------------------------------------------------+
| LINE : DATA ATTRIBUTES                                                         |
+--------------------------------------------------------------------------------+
| 00001:ABArtikel.jav ();                                                        |
|      :+---------------------------------------------------------------------- |
| 00009:class ABArtikel extends ABDBAccess {                                     |
| 00011:String ArtikelNummer;                                                    |
| 00013:int ArtikelMenge;                                                        |
| 00014:int Mindestmenge;                                                        |
```

```
| 00015:float ArtikelPreis;                                                    |
| 00016:String ArtikelName;                                                    |
| 00017:int LieferMenge;                                                       |
| 00018:String Status;                                                         |
|      :+----------------------------------------------------------------------- |
| 00021:public ABArtikel(String ArtikelNr) {                                   |
|      :+----------------------------------------------------------------------- |
| 00034:public void holeArtikelDaten() {                                       |
| 00035:ResultSet result ;                                                     |
| 00039:ResultSetMetaData rsmd ;                                               |
| 00040:int numCols;                                                           |
| 00048:Float Temp ;                                                           |
|      :+----------------------------------------------------------------------- |
| 00061:public boolean pruefeMengeAusreichend(int BestellMenge) {              |
| 00062:boolean MengeAusreichend ;                                             |
|      :+-------------------------------------------------------------------  |
| 00067:public void reduziereMenge(int BestellMenge) {                         |
|      :+-------------------------------------------------------------------  |
+------------------------------------------------------------------------------+
```

6.15.Class File/Database Accesses

```
+------------------------------------------------------------------------------+
| LINE : FILE/DB ACCESSES                                                      |
+------------------------------------------------------------------------------+
|                                                                              |
| 00077:Put (Menumaske);                                                       |
|                                                                              |
| 00092:Get (Auftrag);                        // Einlesen der Maskendaten.     |
|      :                                                                       |
| 00097:Put (Auftrag);                                                         |
|      :                                                                       |
| 00173:Put (Rechnung);                                                        |
|      :                                                                       |
| 00190:Put (Protokoll);                                                       |
|                                                                              |
| 00078:int i=this.executeUpdate("UPDATE `Artikel` SET ArtikelMenge="+NeueMenge+"   |
|              WHERE ArtikelNummer ="+ArtikelNummer                            |
|                                                                              |
+------------------------------------------------------------------------------+
```

6.16.Class Control Logic Diagram

```
+------------------------------------------------------------------------------+
| LINE : CONDITION LOGIC                                                       |
+------------------------------------------------------------------------------+
| 00001:GWM ();                                                                |
+------------------------------------------------------------------------------+
| 00037:GWMKurs_BereinigenDlg::GWMKurs_BereinigenDlg                           |
|      :+----------------------------------------------------------------------- |
| 00051:void GWMKurs_BereinigenDlg::Init (void)                                |
|      :+----------------------------------------------------------------------- |
| 00089:BOOL GWMKurs_BereinigenDlg::Command (USHORT usCommand)                 |
|       +----------------------------------------------------------------------- |
| 00094:|  |if ( fiSuchen.Command (usCommand) )  // Wurde das Command im FI-Suc    |
|       |  +----------------------------------------------------------------------- |
| 00101:|  |switch (usCommand)                                                   |
|       |  |   +------------------------------------------------------------------- |
| 00103:|  |   |case PB_OK:                                                      |
+------------------------------------------------------------------------------- |
| 00105:|  |   |   |if (!Validate())                                             |
+------------------------------------------------------------------------------- |
| 00108:|  |   |   |if (!OwnCheck())                                             |
+------------------------------------------------------------------------------- |
| 00119:|  |   |case PB_CANCEL:                                                  |
+------------------------------------------------------------------------------- |
| 00124:|  |   |default:                                                         |
|      :+----------------------------------------------------------------------- |
| 00142:BOOL GWMKurs_BereinigenDlg::OwnCheck (void)                            |
+------------------------------------------------------------------------------- |
| 00147:|  |if (fdaDatum >= fdaHeute)                                            |
|      :+----------------------------------------------------------------------- |
| 00168:void GWMKurs_BereinigenDlg::EditChange (USHORT usEvent)                |
```

```
+------------------------------------------------------------------------------- |
| 00172:|  |if (fiSuchen.Event (usEvent))                                        |
|      :+------------------------------------------------------------------------ |
| 00185:void GWMKurs_BereinigenDlg::ButtonSelect (USHORT usEvent)                |
+------------------------------------------------------------------------------- |
| 00189:|  |if (fiSuchen.ButtonSelect (usEvent))                                 |
+--------------------------------------------------------------------------------+
```

6.17.Class Test Cases

```
+--------------------------------------------------------------------------------+
| CASE : TEST CASES                                                              |
+--------------------------------------------------------------------------------+
| 00001:GWM ();                                                                  |
+------------------------------------------------------------------------------- |
| 00037:GWMKurs_BereinigenDlg::GWMKurs_BereinigenDlg                             |
|     2: ASSERT GWMKurs_BereinigenDlg::GWMKurs_BereinigenDlg                     |
+------------------------------------------------------------------------------- |
| 00051:void GWMKurs_BereinigenDlg::Init (void)                                  |
|     3: ASSERT void GWMKurs_BereinigenDlg::Init (void)                          |
+------------------------------------------------------------------------------- |
| 00089:BOOL GWMKurs_BereinigenDlg::Command (USHORT usCommand)                   |
|     4: ASSERT BOOL GWMKurs_BereinigenDlg::Command (USHORT usCommand)           |
|     4: ASSERT fiSuchen.Command (usCommand) ;                                   |
|     5: ASSERT !  fiSuchen.Command (usCommand) ;                                |
|     5: ASSERT usCommand == PB_OK;                                              |
|     5: ASSERT !Validate();                                                     |
|     6: ASSERT Validate();                                                      |
|     5: ASSERT !OwnCheck();                                                     |
|     7: ASSERT OwnCheck();                                                      |
|     9: ASSERT usCommand == PB_CANCEL;                                          |
|    10: ASSERT usCommand == 99;                                                 |
+------------------------------------------------------------------------------- |
| 00142:BOOL GWMKurs_BereinigenDlg::OwnCheck (void)                              |
|    11: ASSERT BOOL GWMKurs_BereinigenDlg::OwnCheck (void)                      |
|    11: ASSERT fdaDatum >= fdaHeute;                                            |
|    12: ASSERT fdaDatum <= fdaHeute;                                            |
+------------------------------------------------------------------------------- |
| 00168:void GWMKurs_BereinigenDlg::EditChange (USHORT usEvent)                  |
|    13: ASSERT void GWMKurs_BereinigenDlg::EditChange (USHORT usEvent)          |
|    13: ASSERT fiSuchen.Event (usEvent);                                        |
|    14: ASSERT ! fiSuchen.Event (usEvent);                                      |
+------------------------------------------------------------------------------- |
| 00185:void GWMKurs_BereinigenDlg::ButtonSelect (USHORT usEvent)                |
|    15: ASSERT void GWMKurs_BereinigenDlg::ButtonSelect (USHORT usEvent)        |
|    15: ASSERT fiSuchen.ButtonSelect (usEvent);                                 |
|    16: ASSERT ! fiSuchen.ButtonSelect (usEvent);                               |
|                                                                                |
+--------------------------------------------------------------------------------+
|                                                                                |
| ===>>: Minimum Number of Testcases = 0016                                      |
|                                                                                |
+--------------------------------------------------------------------------------+
```

6.18.Database Table Structure

```
+-------------------------------------------------------------------+
|   S O F T A N A L   D A T A B A S E   S P E C I F I C A T I O N   |
|   DB_TYPE: SQL                                    DATE: 21.07.05 |
|   DB_NAME: DB2                                                   |
|   DB_PATH: C:\DATA\SQL\OUTPUT\SPECS\TCHILD.sql                   |
+-------------------------------------------------------------------+
|   TABLE:   TCHILD                          PRIMARY  FOREIGN    DB  |
|   DATANAME               DATATYPE            KEY      KEY     INDEX |
+-------------------------------------------------------------------+
|==>C_PISRT_CHL            TIMESTAMP            *                *   |
|==>C_VERSION_KEY          CHAR (1)                      *           |
|==>C_VALNR                CHAR (12)                     *           |
|==>C_VALZUS               CHAR (2)                      *           |
|==>C_ZUGEH                CHAR (5)                      *           |
|==>C_EIGENT               CHAR (1)                      *           |
|==>C_NL                   CHAR (4)                      *           |
```

```
|==>C_RESKEYBE             CHAR (8)                    *           |
|==>C_PISRT_PRN            TIMESTAMP                   *       *   |
|==>C_IDATP                DATE                                    |
|==>C_NOMTIT_OWAE          DECIMAL                                 |
|==>C_ANZTIT               DECIMAL                                 |
|==>C_TOTSTI               DECIMAL                                 |
|==>C_NOM_SFR              DECIMAL                                 |
|==>C_NOM_USD              DECIMAL                                 |
|==>C_ANZBET               DECIMAL                                 |
|==>C_STIBET               DECIMAL                                 |
|==>C_KAPBET_OWEA          DECIMAL                                 |
|==>C_KAPBET_CHF           DECIMAL                                 |
|==>C_KAPBET_USD           DECIMAL                                 |
|==>C_KURS_OWAE            DECIMAL                                 |
|==>C_KURS_USD             DECIMAL                                 |
|==>C_KUDAT_OWAE           DATE                                    |
|==>C_KUDAT_USD            DATE                                    |
|==>C_BPRZST               DECIMAL                                 |
|==>C_BPRZKP               DECIMAL                                 |
+--------------------------------------------------------------------+
```

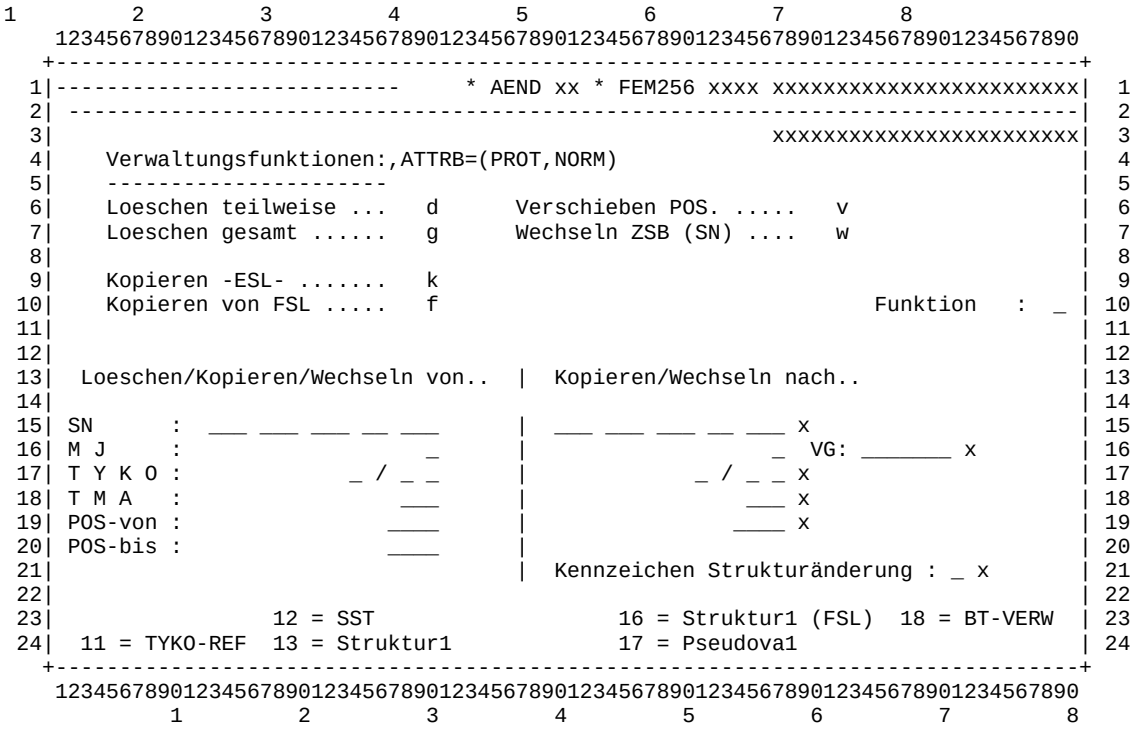
6.19.Database Cross References

6.20.User Interface Contents

```
+-----------------------------------------------------------+
|    S O F T A N A L   M A P   S P E C I F I C A T I O N    |
|   MAP_TYPE:BMS                          DATE: 27.01.09    |
|   MAP_NAME:FEM256                                         |
|   MAP_PATH:.\SPECS\FEM256.bms                             |
+-----------------------------------------------------------+
|   FIELDNAME          FIELDTYPE    LINE    POS    USAGE    |
+-----------------------------------------------------------+
|==>FEM256-001         CHAR(0020)    01      01    TITLE    |
|==>FEM256-002         CHAR(0006)    01      32    TITLE    |
|==>MSEITE             CHAR(0002)    01      39    OUTPUT   |
|==>FEM256-004         CHAR(0008)    01      42    TITLE    |
|==>MFNUMM             CHAR(0004)    01      51    OUTPUT   |
|==>MFTEXT             CHAR(0024)    01      56    OUTPUT   |
|==>FEM256-007         CHAR(0079)    02      01    TITLE    |
|==>MFTEXZ             CHAR(0024)    03      56    OUTPUT   |
|==>FEM256-009         CHAR(0025)    04      01    TITLE    |
|==>FEM256-010         CHAR(0022)    05      04    TITLE    |
|==>FEM256-011         CHAR(0026)    06      04    TITLE    |
|==>FEM256-012         CHAR(0026)    06      36    TITLE    |
|==>FEM256-013         CHAR(0026)    07      04    TITLE    |
|==>FEM256-014         CHAR(0026)    07      36    TITLE    |
|==>FEM256-015         CHAR(0026)    09      04    TITLE    |
|==>FEM256-016         CHAR(0026)    10      04    TITLE    |
|==>FEM256-017         CHAR(0012)    10      64    TITLE    |
|==>FUNKT              CHAR(0001)    10      78    INPUT    |
|==>FEM256-019         CHAR(0001)    10      80    TITLE    |
|==>FEM256-020         CHAR(0080)    11      80    TITLE    |
|==>FEM256-021         CHAR(0061)    13      02    TITLE    |
|==>FEM256-022         CHAR(0080)    13      80    TITLE    |
|==>FEM256-023         CHAR(0002)    15      01    TITLE    |
|==>FEM256-024         CHAR(0001)    15      09    TITLE    |
|==>SNTYPV             CHAR(0003)    15      12    INPUT    |
|==>SNMGRV             CHAR(0003)    15      16    INPUT    |
|==>SNENDV             CHAR(0003)    15      20    INPUT    |
|==>SNINDV             CHAR(0002)    15      24    INPUT    |
|==>SNFCOV             CHAR(0003)    15      27    INPUT    |
|==>FEM256-030         CHAR(0006)    15      31    TITLE    |
|==>SNTYPN             CHAR(0003)    15      39    INPUT    |
|==>SNMGRN             CHAR(0003)    15      43    INPUT    |
|==>SNENDN             CHAR(0003)    15      47    INPUT    |
|==>SNINDN             CHAR(0002)    15      51    INPUT    |
|==>SNFCON             CHAR(0003)    15      54    INPUT    |
|==>FEM256-036         CHAR(0001)    15      58    TITLE    |
```

```
|==>FEM256-037          CHAR(0009)    16      01    TITLE  |
|==>MJVON               CHAR(0001)    16      29    INPUT  |
|==>FEM256-039          CHAR(0006)    16      31    TITLE  |
|==>MJNACH              CHAR(0001)    16      56    INPUT  |
|==>FEM256-041          CHAR(0004)    16      58    TITLE  |
|==>VGNR                CHAR(0007)    16      63    INPUT  |
|==>FEM256-043          CHAR(0001)    16      71    TITLE  |
|==>FEM256-044          CHAR(0009)    17      01    TITLE  |
|==>TAVON               CHAR(0001)    17      23    INPUT  |
|==>FEM256-046          CHAR(0001)    17      25    TITLE  |
|==>TFVON               CHAR(0001)    17      27    INPUT  |
|==>TGVON               CHAR(0001)    17      29    INPUT  |
|==>FEM256-049          CHAR(0006)    17      31    TITLE  |
|==>TANACH              CHAR(0001)    17      50    INPUT  |
|==>FEM256-051          CHAR(0001)    17      52    TITLE  |
|==>TFNACH              CHAR(0001)    17      54    INPUT  |
|==>TGNACH              CHAR(0001)    17      56    INPUT  |
|==>FEM256-054          CHAR(0001)    17      58    TITLE  |
|==>FEM256-055          CHAR(0009)    18      01    TITLE  |
|==>TMAVON              CHAR(0003)    18      27    INPUT  |
|==>FEM256-057          CHAR(0006)    18      31    TITLE  |
|==>TMANACH             CHAR(0003)    18      54    INPUT  |
|==>FEM256-059          CHAR(0001)    18      58    TITLE  |
|==>FEM256-060          CHAR(0009)    19      01    TITLE  |
|==>POSVV               CHAR(0004)    19      26    INPUT  |
|==>FEM256-062          CHAR(0006)    19      31    TITLE  |
|==>POSVN               CHAR(0004)    19      53    INPUT  |
|==>FEM256-064          CHAR(0001)    19      58    TITLE  |
|==>FEM256-065          CHAR(0009)    20      01    TITLE  |
|==>POSBV               CHAR(0004)    20      26    INPUT  |
|==>FEM256-067          CHAR(0006)    20      31    TITLE  |
|==>FEM256-068          CHAR(0033)    21      36    TITLE  |
|==>STRNACH             CHAR(0001)    21      70    INPUT  |
|==>FEM256-070          CHAR(0001)    21      72    TITLE  |
|==>FEM256-071          CHAR(0080)    21      80    TITLE  |
|==>FEM256-072          CHAR(0008)    23      17    TITLE  |
|==>FEM256-073          CHAR(0034)    23      44    TITLE  |
|==>FEM256-074          CHAR(0029)    24      02    TITLE  |
|==>FEM256-075          CHAR(0014)    24      44    TITLE  |
|                                                          |
+----------------------------------------------------------+
```

6.21.User Interface Prototype



6.22.Extracted Comment List

```
_____________________________________________________________________________
source: GWP
_____________________________________________________________________________
// ---------------------------------------------------------------------------
// GWPKurs::GWPKurs
// author: MH
//
// Description
//
// Constructor
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectMarkt
// author: MH
//
// Description
//
// Marke lesen.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectMarkante
// author: MH
//
// Description
//
// Markante Kurse lesen.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectTagesWp
// author: MH
//
// Description
//
// Tageskurse Wertpapier lesen.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectTagesDerivat
// author: MH
//
// Description
//
// Tageskurse Derivat lesen.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectTages
// author: MH
//
// Description
//
// Tageskurse lesen.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectErfassen
// author: MH
//
// Description
//
// Select f□r Kurse ERFASSEN.
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectVorherigenExist
// author: MH
//
// Description
//
// Select f□r vorherigen Kurs. (Pr□ft ob aktueller bereits existiert)
// ---------------------------------------------------------------------------
// ---------------------------------------------------------------------------
// GWPKurs::SelectVorherigen
// author: MH
//
// Description
```

```
//
// Select f□r vorherigen Kurs.
// ------------------------------------------------------------------------
// ------------------------------------------------------------------------
// GWPKurs::SelectExist
// author: MH
//
// Description
//
// Pr□ft ob Kurs existiert
// ------------------------------------------------------------------------
// ------------------------------------------------------------------------
// GWPKurs::SelectSuchenErfassen
// author: MH
//
// Description
//
// Select f□r SUCHEN & ERFASSEN.
// ------------------------------------------------------------------------
// ------------------------------------------------------------------------
// GWPKurs::SelectVorhMark
// author: MH
//
// Description
//
// Select f□r vorherigen markanten Kurs.
// ------------------------------------------------------------------------
// ------------------------------------------------------------------------
// GWPKurs::Update
// author: MH
//
// Description
//
// Aktualisieren von Kursen (INS, UPD, DEL).
// ------------------------------------------------------------------------
// EOF/GEOS/GWPKURS.CXX
End of Source; ____________________________________________________________
```

6.23.Repository Export Table

```
Type;Base Entity                        ;Rel ;Type;Target Entity
PROD;SPARKASE                           ;OWNS;SYS ;DB
SYS ;DB                                 ;OWNS;PROG;PERSON
PROG;PERSON                             ;ISAT;LIB ;COMMENTS\DB\PERSON.txt
PROG;PERSON                             ;OWNS;MOD ;Person.jav
MOD ;Person.jav                         ;ISAT;LIB ;DB\PERSON\Person.jav
MOD ;Person.jav                         ;INCL;COPY;*.java
MOD ;Person.jav                         ;INCL;COPY;TransportObject.java
MOD ;Person.jav                         ;INCL;COPY;*.java
MOD ;Person.jav                         ;INCL;COPY;BspException.java
MOD ;Person.jav                         ;OWNS;CLAS;Person
CLAS;Person                             ;USES;CLAS;TransportObject
CLAS;Person                             ;OWNS;FUNC;Person
FUNC;Person                             ;LINK;PARM;VarChar cRole
FUNC;Person                             ;LINK;PARM;Nummer nVersion
FUNC;Person                             ;LINK;PARM;TimeStamp tsReadTime
FUNC;Person                             ;LINK;PARM;int iSeqNumber
FUNC;Person                             ;LINK;PARM;Nummer nBBZS
FUNC;Person                             ;LINK;PARM;Datum dGeburtsdatum
FUNC;Person                             ;LINK;PARM;Nummer nGrdNr
FUNC;Person                             ;LINK;PARM;VarChar cNachname
FUNC;Person                             ;LINK;PARM;VarChar cVorname
FUNC;Person                             ;LINK;PARM;VarChar cKzSparer
FUNC;Person                             ;LINK;PARM;boolean bKzArchiv
FUNC;Person                             ;LINK;PARM;VarChar cKzAusland
FUNC;Person                             ;LINK;PARM;VarChar cOrt
FUNC;Person                             ;LINK;PARM;VarChar cPlz
FUNC;Person                             ;LINK;PARM;VarChar cStrasse
CLAS;Person                             ;OWNS;FUNC;Person
FUNC;Person                             ;LINK;PARM;Nummer nBBZS
FUNC;Person                             ;LINK;PARM;Datum dGeburtsdatum
FUNC;Person                             ;LINK;PARM;Nummer nGrdNr
FUNC;Person                             ;LINK;PARM;VarChar cNachname
FUNC;Person                             ;LINK;PARM;VarChar cVorname
```

```
FUNC;Person                    ;LINK;PARM;VarChar cKzSparer
FUNC;Person                    ;LINK;PARM;boolean bKzArchiv
FUNC;Person                    ;LINK;PARM;VarChar cKzAusland
FUNC;Person                    ;LINK;PARM;VarChar cOrt
FUNC;Person                    ;LINK;PARM;VarChar cPlz
FUNC;Person                    ;LINK;PARM;VarChar cStrasse
CLAS;Person                    ;OWNS;FUNC;getBBZS
CLAS;Person                    ;OWNS;FUNC;getGeburtsdatum
CLAS;Person                    ;OWNS;FUNC;getGrdNr
CLAS;Person                    ;OWNS;FUNC;getNachname
CLAS;Person                    ;OWNS;FUNC;getKzSparer
CLAS;Person                    ;OWNS;FUNC;getKzArchiv
CLAS;Person                    ;OWNS;FUNC;getVorname
CLAS;Person                    ;OWNS;FUNC;getKzAusland
CLAS;Person                    ;OWNS;FUNC;getOrt
CLAS;Person                    ;OWNS;FUNC;getPlz
CLAS;Person                    ;OWNS;FUNC;getStrasse
CLAS;Person                    ;OWNS;FUNC;setBBZS
CLAS;Person                    ;OWNS;FUNC;setGeburtsdatum
CLAS;Person                    ;OWNS;FUNC;setGrdNr
CLAS;Person                    ;OWNS;FUNC;setNachname
CLAS;Person                    ;OWNS;FUNC;setKzSparer
CLAS;Person                    ;OWNS;FUNC;setKzArchiv
CLAS;Person                    ;OWNS;FUNC;setVorname
CLAS;Person                    ;OWNS;FUNC;setKzAusland
CLAS;Person                    ;OWNS;FUNC;setOrt
CLAS;Person                    ;OWNS;FUNC;setPlz
CLAS;Person                    ;OWNS;FUNC;setStrasse
CLAS;Person                    ;OWNS;FUNC;o2x
FUNC;o2x                       ;LINK;PARM;Writer oWriter
CLAS;Person                    ;OWNS;FUNC;Person
CLAS;Person                    ;OWNS;FUNC;getBBZS
CLAS;Person                    ;OWNS;FUNC;getGeburtsdatum
CLAS;Person                    ;OWNS;FUNC;getGrdNr
CLAS;Person                    ;OWNS;FUNC;getNachname
CLAS;Person                    ;OWNS;FUNC;getKzSparer
CLAS;Person                    ;OWNS;FUNC;getKzArchiv
CLAS;Person                    ;OWNS;FUNC;getVorname
CLAS;Person                    ;OWNS;FUNC;getKzAusland
CLAS;Person                    ;OWNS;FUNC;getOrt
CLAS;Person                    ;OWNS;FUNC;getPlz
CLAS;Person                    ;OWNS;FUNC;getStrasse
CLAS;Person                    ;OWNS;FUNC;setBBZS
CLAS;Person                    ;OWNS;FUNC;setGeburtsdatum
CLAS;Person                    ;OWNS;FUNC;setGrdNr
CLAS;Person                    ;OWNS;FUNC;setNachname
CLAS;Person                    ;OWNS;FUNC;setKzSparer
CLAS;Person                    ;OWNS;FUNC;setKzArchiv
CLAS;Person                    ;OWNS;FUNC;setVorname
CLAS;Person                    ;OWNS;FUNC;setKzAusland
CLAS;Person                    ;OWNS;FUNC;setOrt
CLAS;Person                    ;OWNS;FUNC;setPlz
CLAS;Person                    ;OWNS;FUNC;setStrasse
CLAS;Person                    ;OWNS;FUNC;o2x
FUNC;o2x                       ;CALL;FUNC;write
CLAS;Person                    ;OWNS;FUNC;Person.jav
CLAS;Person                    ;OWNS;DATA;public static final int s_iNACHNAME_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iVORNAME_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iSTRASSE_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iORT_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iPLZ_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iPLZ_FORMAT
CLAS;Person                    ;OWNS;DATA;public static final int s_iGRDNR_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iBBZS_LENGTH
CLAS;Person                    ;OWNS;DATA;public static final int s_iKZSPARER_LENGTH
CLAS;Person                    ;OWNS;DATA;protected Nummer m_nBBZS
CLAS;Person                    ;OWNS;DATA;protected Datum m_dGeburtsdatum
CLAS;Person                    ;OWNS;DATA;protected Nummer m_nGrdNr
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cNachname
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cKzSparer
CLAS;Person                    ;OWNS;DATA;protected boolean m_bKzArchiv
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cVorname
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cKzAusland
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cOrt
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cPlz
CLAS;Person                    ;OWNS;DATA;protected VarChar m_cStrasse
CLAS;Person                    ;OWNS;FUNC;Person
CLAS;Person                    ;OWNS;FUNC;getBBZS
```

```
CLAS;Person                             ;OWNS;FUNC;getGeburtsdatum
CLAS;Person                             ;OWNS;FUNC;getGrdNr
CLAS;Person                             ;OWNS;FUNC;getNachname
CLAS;Person                             ;OWNS;FUNC;getKzSparer
CLAS;Person                             ;OWNS;FUNC;getKzArchiv
CLAS;Person                             ;OWNS;FUNC;getVorname
CLAS;Person                             ;OWNS;FUNC;getKzAusland
CLAS;Person                             ;OWNS;FUNC;getOrt
CLAS;Person                             ;OWNS;FUNC;getPlz
CLAS;Person                             ;OWNS;FUNC;getStrasse
CLAS;Person                             ;OWNS;FUNC;setBBZS
CLAS;Person                             ;OWNS;FUNC;setGeburtsdatum
CLAS;Person                             ;OWNS;FUNC;setGrdNr
CLAS;Person                             ;OWNS;FUNC;setNachname
CLAS;Person                             ;OWNS;FUNC;setKzSparer
CLAS;Person                             ;OWNS;FUNC;setKzArchiv
CLAS;Person                             ;OWNS;FUNC;setVorname
CLAS;Person                             ;OWNS;FUNC;setKzAusland
CLAS;Person                             ;OWNS;FUNC;setOrt
CLAS;Person                             ;OWNS;FUNC;setPlz
CLAS;Person                             ;OWNS;FUNC;setStrasse
```

6.24.SoftCalc Metric Export File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--DOCTYPE CENTITY SYSTEM "centity.xsd"-->
 <CENTITY>
   <CENTITY_FILE
    Product = "DEMO"
    System = "COBSYST"
    Date = "18.02.09"
    Source = "SOFREDOC"
   />
   <CODE_ENTITY
    Entity = "Component"
    EName = "COBOLD;cob"
    EType = "MODU" >
     <Component_Attributes>
         <Nr_Locs>0800</Nr_Locs>
         <Nr_Stmts>0400</Nr_Stmts>
         <Nr_Functs>0032</Nr_Functs>
         <Nr_Variables>0080</Nr_Variables>
         <Change_Rate>0</Change_Rate>
     </Component_Attributes>
   </CODE_ENTITY>
   <CODE_ENTITY
    Entity = "INTERFACE"
    EName = "ORDER_FILE"
    EType = "FILE" >
     <Interface_Attributes>
         <Nr_Targets>0001</Nr_Targets>
         <Nr_Operations>0004</Nr_Operations>
         <Nr_Arguments>0008</Nr_Arguments>
         <Nr_Results>0002</Nr_Results>
         <Change_Rate>0</Change_Rate>
     </Interface_Attributes>
   </CODE_ENTITY>
   <CODE_ENTITY
    Entity = "INTERFACE"
    EName = "CUSTOMER_FILE"
    EType = "FILE" >
     <Interface_Attributes>
         <Nr_Targets>0001</Nr_Targets>
         <Nr_Operations>0004</Nr_Operations>
         <Nr_Arguments>0008</Nr_Arguments>
         <Nr_Results>0002</Nr_Results>
         <Change_Rate>0</Change_Rate>
     </Interface_Attributes>
   </CODE_ENTITY>
   <CODE_ENTITY
    Entity = "INTERFACE"
    EName = "I_O"
    EType = "FILE" >
     <Interface_Attributes>
         <Nr_Targets>0001</Nr_Targets>
```

```
        <Nr_Operations>0004</Nr_Operations>
        <Nr_Arguments>0008</Nr_Arguments>
        <Nr_Results>0002</Nr_Results>
        <Change_Rate>0</Change_Rate>
     </Interface_Attributes>
   </CODE_ENTITY>
   <CODE_ENTITY
    Entity = "INTERFACE"
    EName = "DISPATCH_FILE"
    EType = "FILE" >
     <Interface_Attributes>
        <Nr_Targets>0001</Nr_Targets>
        <Nr_Operations>0004</Nr_Operations>
        <Nr_Arguments>0008</Nr_Arguments>
        <Nr_Results>0002</Nr_Results>
        <Change_Rate>0</Change_Rate>
     </Interface_Attributes>
   </CODE_ENTITY>
   <CODE_ENTITY
    Entity = "INTERFACE"
    EName = "ARTICLE_FILE"
    EType = "FILE" >
     <Interface_Attributes>
        <Nr_Targets>0001</Nr_Targets>
        <Nr_Operations>0004</Nr_Operations>
        <Nr_Arguments>0008</Nr_Arguments>
        <Nr_Results>0002</Nr_Results>
        <Change_Rate>0</Change_Rate>
     </Interface_Attributes>
   </CODE_ENTITY>
</CENTITY>
```
