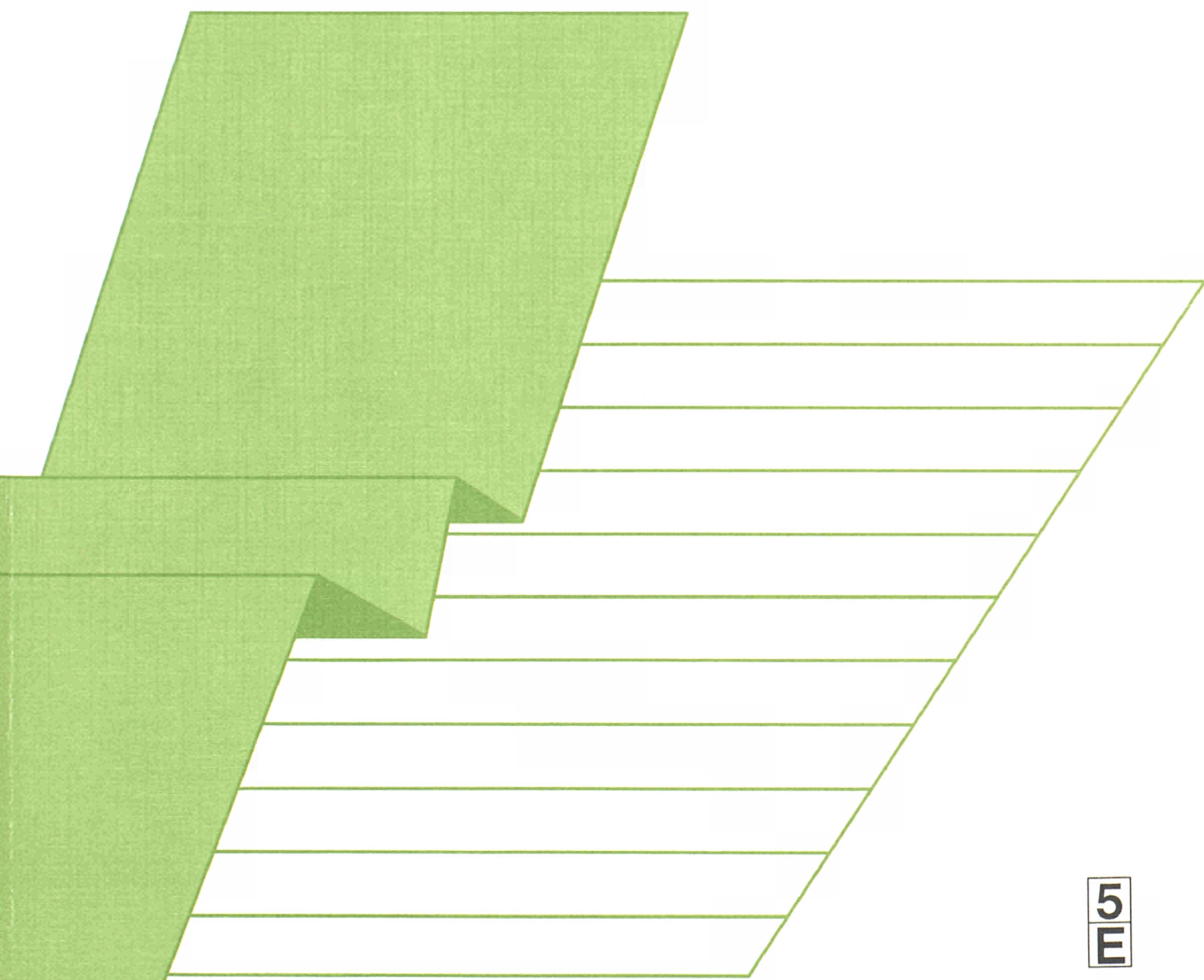


# SPEL System

Technical documentation





STATISTISCHES AMT DER EUROPÄISCHEN GEMEINSCHAFTEN  
STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES  
OFFICE STATISTIQUE DES COMMUNAUTÉS EUROPÉENNES

L-2920 Luxembourg — Tél. 43 01-1 — Télex: Comeur Lu 3423  
B-1049 Bruxelles, bâtiment Berlaymont, rue de la Loi 200 (bureau de liaison) — Tél. 235 11 11

Eurostat hat die Aufgabe, den Informationsbedarf der Kommission und aller am Aufbau des Binnenmarktes Beteiligten mit Hilfe des europäischen statistischen Systems zu decken.

Um der Öffentlichkeit die große Menge an verfügbaren Daten zugänglich zu machen und Benutzern die Orientierung zu erleichtern, werden zwei Arten von Publikationen angeboten: Statistische Dokumente und Veröffentlichungen.

Statistische Dokumente sind für den Fachmann konzipiert und enthalten das ausführliche Datenmaterial: Bezugsdaten, bei denen die Konzepte allgemein bekannt, standardisiert und wissenschaftlich fundiert sind. Diese Daten werden in einer sehr tiefen Gliederung dargeboten. Die Statistischen Dokumente wenden sich an Fachleute, die in der Lage sind, selbständig die benötigten Daten aus der Fülle des dargebotenen Materials auszuwählen. Diese Daten sind in gedruckter Form und/oder auf Diskette, Magnetband, CD-ROM verfügbar. Statistische Dokumente unterscheiden sich auch optisch von anderen Veröffentlichungen durch den mit einer stilisierten Graphik versehenen weißen Einband.

Die zweite Publikationsart, die Veröffentlichungen, wenden sich an eine ganz bestimmte Zielgruppe, wie zum Beispiel an den Bildungsbereich oder an Entscheidungsträger in Politik und Verwaltung. Sie enthalten ausgewählte und auf die Bedürfnisse einer Zielgruppe abgestellte und kommentierte Informationen. Eurostat übernimmt hier also eine Art Beraterrolle.

Für einen breiteren Benutzerkreis gibt Eurostat Jahrbücher und periodische Veröffentlichungen heraus. Diese enthalten statistische Ergebnisse für eine erste Analyse sowie Hinweise auf weiteres Datenmaterial für vertiefende Untersuchungen. Diese Veröffentlichungen werden in gedruckter Form und in Datenbanken angeboten, die in Menütechnik zugänglich sind.

Um Benutzern die Datensuche zu erleichtern, hat Eurostat Themenkreise, d. h. eine Untergliederung nach Sachgebieten, eingeführt. Daneben sind sowohl die Statistischen Dokumente als auch die Veröffentlichungen in bestimmte Reihen, wie zum Beispiel „Jahrbücher“, „Konjunktur“, „Methoden“, untergliedert, um den Zugriff auf die statistischen Informationen zu erleichtern.

Y. Franchet  
Generaldirektor

It is Eurostat's responsibility to use the European statistical system to meet the requirements of the Commission and all parties involved in the development of the single market.

To ensure that the vast quantity of accessible data is made widely available, and to help each user make proper use of this information, Eurostat has set up two main categories of document: statistical documents and publications.

The statistical document is aimed at specialists and provides the most complete sets of data: reference data where the methodology is well established, standardized, uniform and scientific. These data are presented in great detail. The statistical document is intended for experts who are capable of using their own means to seek out what they require. The information is provided on paper and/or on diskette, magnetic tape, CD-ROM. The white cover sheet bears a stylized motif which distinguishes the statistical document from other publications.

The publications proper tend to be compiled for a well-defined and targeted public, such as educational circles or political and administrative decision-makers. The information in these documents is selected, sorted and annotated to suit the target public. In this instance, therefore, Eurostat works in an advisory capacity.

Where the readership is wider and less well defined, Eurostat provides the information required for an initial analysis, such as yearbooks and periodicals which contain data permitting more in-depth studies. These publications are available on paper or in Videotext databases.

To help the user focus his research, Eurostat has created 'themes', i.e. a subject classification. The statistical documents and publications are listed by series: e.g. yearbooks, short-term trends or methodology in order to facilitate access to the statistical data.

Y. Franchet  
Director-General

Pour établir, évaluer ou apprécier les différentes politiques communautaires, la Commission des Communautés européennes a besoin d'informations.

Eurostat a pour mission, à travers le système statistique européen, de répondre aux besoins de la Commission et de l'ensemble des personnes impliquées dans le développement du marché unique.

Pour mettre à la disposition de tous l'importante quantité de données accessibles et faire en sorte que chacun puisse s'orienter correctement dans cet ensemble, deux grandes catégories de documents ont été créées: les documents statistiques et les publications.

Le document statistique s'adresse aux spécialistes. Il fournit les données les plus complètes: données de référence où la méthodologie est bien connue, standardisée, normalisée et scientifique. Ces données sont présentées à un niveau très détaillé. Le document statistique est destiné aux experts capables de rechercher, par leurs propres moyens, les données requises. Les informations sont alors disponibles sur papier et/ou sur disquette, bande magnétique, CD-ROM. La couverture blanche ornée d'un graphisme stylisé démarque le document statistique des autres publications.

Les publications proprement dites peuvent, elles, être réalisées pour un public bien déterminé, ciblé, par exemple l'enseignement ou les décideurs politiques ou administratifs. Des informations sélectionnées, triées et commentées en fonction de ce public lui sont apportées. Eurostat joue, dès lors, le rôle de conseiller.

Dans le cas d'un public plus large, moins défini, Eurostat procure des éléments nécessaires à une première analyse, les annuaires et les périodiques, dans lesquels figurent les renseignements adéquats pour approfondir l'étude. Ces publications sont présentées sur papier ou dans des banques de données de type vidéotex.

Pour aider l'utilisateur à s'orienter dans ses recherches, Eurostat a créé les thèmes, c'est-à-dire une classification par sujet. Les documents statistiques et les publications sont répertoriés par série — par exemple, annuaire, conjoncture, méthodologie — afin de faciliter l'accès aux informations statistiques.

Y. Franchet  
Directeur général

# SPEL System

Technical documentation

Theme  
Agriculture, forestry and fisheries  
Series  
Methods

5
E

Cataloguing data can be found at the end of this publication.

Luxembourg: Office for Official Publications of the European Communities, 1992

ISBN 92-826-4498-7

© ECSC-EEC-EAEC, Brussels • Luxembourg, 1992

Reproduction is authorized, except for commercial purposes, provided the source is acknowledged.

*Printed in Belgium*



SEKTORALES PRODUKTIONS- UND EINKOMMENSMODELL DER LANDWIRTSCHAFT  
SECTORAL PRODUCTION AND INCOME MODEL FOR AGRICULTURE  
MODELE DE LA PRODUCTION ET DES REVENUS DU SECTEUR AGRICOLE  
MODELO DE LA PRODUCCION Y DE LAS RENTAS DEL SECTOR AGRARIO  
INOTÆGTS- OG PRODUKTIONSMODEL FOR LANDBRUGSSEKTOREN  
ΜΟΝΤΕΛΟ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΕΙΣΟΔΗΜΑΤΟΣ ΤΟΥ ΑΓΡΟΤΙΚΟΥ ΤΟΜΕΑ  
MODELLO DELLA PRODUZIONE E DEI REDDITI DEL SETTORE AGRICOLA  
MODEL VAN DE PRODUKTIE EN DE INKOMSTEN VAN DE LANDBOUWSECTOR  
MODELO DA PRODUCAO E DOS RENDIMENTOS DO SECTOR AGRICOLA

The concept of the SPEL-System was developed at the  
Institut für Agrarpolitik, Marktforschung und  
Wirtschaftssoziologie of the University of Bonn  
by W. Henrichsmeyer, W. Wolf and H. J. Greuel.

The following Technical Documentation was composed by  
A. Zintl and H. J. Greuel at the same institute and was edited  
in cooperation with C. Loux, Eurostat.

For further details:  
F. Pfähler  
Eurostat, F-1  
Tel. 4301-7239



# **TABLE OF CONTENTS**

## **PART 1: Basics**

- 1. INTRODUCTION**
- 2. IMPLEMENTATION**
- 3. FORMAL STRUCTURING AND CODING**
- 4. DATA MANAGEMENT**
- 5. USER INTERFACE**
- 6. SPEL PROGRAMS**
- 7. INDEX**

## **PART 2: SPEL/EC Model**

- 1. INTRODUCTION**
- 2. MODEL COMPONENTS AND IMPLEMENTATION**
- 3. DATA PREPARATION**
- 4. MODEL CALCULATIONS**
- 5. UTILITIES**
- 6. EXPLOITATION**
- 7. ANNEX A: ERROR MESSAGES AND WARNINGS**
- 8. INDEX**





# **PART 1**

Operator/User's Guide, Basics



# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>13</b>
<b>2. IMPLEMENTATION.....</b>	<b>14</b>
2.1. Users and operators.....	14
2.2. System software requirements .....	16
2.3. System initialization and environment.....	16
<b>3. FORMAL STRUCTURING AND CODING .....</b>	<b>17</b>
3.1. SPEL-Data dimensions and SPEL table subkeys .....	17
3.1.1. Region .....	19
3.1.2. Sub-region .....	19
3.1.3. Year.....	19
3.1.4. Periodicity .....	20
3.1.5. Status .....	20
3.1.6. Model area.....	20
3.1.7. Table column .....	20
3.1.8. Table row .....	21
3.2. CODES files currently in use.....	21
<b>4. DATA MANAGEMENT .....</b>	<b>22</b>
4.1. External data sources importation .....	22
4.2. SPEL standard data file formats.....	22
4.2.1. SPELIMP files.....	22
4.2.2. SDATA files .....	24
4.2.3. TABFILE files.....	26
<b>5. USER INTERFACE.....</b>	<b>29</b>
5.1. General program control .....	29
5.1.1. General control files (PARAM files) .....	29
5.1.2. Panels .....	30
5.1.3. On-line help information .....	34
5.1.4. Parameter file link panels .....	35
5.2. Special control files .....	36
5.2.1. ASSIGN files.....	36
5.2.2. REGSTE files .....	37
5.2.3. STEER files .....	39
5.2.4. SDATA files .....	39
<b>6. SPEL PROGRAMS .....</b>	<b>40</b>
6.1. Multi-Purpose Programs.....	40
6.1.1. Importing standardized sequential files into TABFILE work files (ORIGIN).....	40
6.1.2. Trend estimations (TREND) .....	47
6.2. Utilities.....	55

6.2.1.	General estimation and evaluation program (EV)	55
6.2.1.1.	Application abstracts	55
6.2.1.2.	Types of data files	56
6.2.1.3.	Control language structure	56
6.2.1.4.	Control language details	57
6.2.1.5.	A typical EV session	82
6.2.2.	Data format conversion (DATCON)	86
6.2.3.	Selection and conversion of sequential standard file formats (FILTER)	86
6.2.4.	Maintenance of TABFILE files (DASERV)	94
6.2.5.	Print and export data from TABFILE file (DAOUT)	133
<b>7.</b>	<b>INDEX</b>	<b>143</b>

## LIST OF FIGURES

Figure 1 : User and operator level.....	15
Figure 2 : SPEL data dimensions.....	18
Figure 3 : Structure of TABFILE files.....	28
Figure 4 : ORIGIN program flow chart.....	41
Figure 5 : TREND program flow chart.....	48
Figure 6 : FILTER program flow chart.....	88
Figure 7 : DASERV program flow chart.....	95
Figure 8 : DAOUT program flow chart.....	134



## 1. INTRODUCTION

The SPEL approach is characterized by mutual interaction between model-builders and statisticians or policy-makers. There are different models to deal with different questions, e.g. the SPEL/EC Model for the agricultural sector of the EC Member States and the EC as a whole.

In general the SPEL system offers a systematically structured data system to be used for

- checking the consistency of agricultural statistics,
- monitoring the present situation in agricultural sector,
- ex-post analyses of sectoral developments, and
- forecasts and policy simulations of the effects of alternative policies from short-term and medium-term viewpoints.

The set of statistical data is not taken as final but subjected to consistency checks and critical investigations, which may lead to the jointly agreed revision of existing statistics and proposals for amendments and conceptual changes for new statistics. It is not assumed either that policy-makers are able to specify target variables before taking a closer look at the problems involved. Instead, the fundamental idea is that target priorities emerge during the process of policy dialog on trade-offs between target variables.

The principle technical approach for the whole SPEL system is transparency and flexible user interface. Transparency means that each data element can be traced back to the basic data sources and the underlying assumptions. The flexible user interface facilitates dialogue between policy-makers and the model.

For the methodological aspects see the methodological documentation.

The technical documentation of SPEL system consists of the following parts :

Basic information

- Part 1: Operator/ User's Guide : Basics

Specific information for the SPEL/EC Model

- Part2: Operator/ User's Guide : SPEL/EC Model

The technical documentation is designed to be a guide for persons familiar with the VM/CMS system on an IBM or compatible mainframe.

## **2. IMPLEMENTATION**

The SPEL software is designed to be implemented on the CMS dialog system on the Amdahl mainframe at Eurostat in Luxembourg. The programming language is FORTRAN 77.

### **2.1. Users and operators**

To become familiar with the SPEL system it is important to know about the SPEL system structure and the work levels for SPEL operators and users.

Working with the SPEL system can be done on two levels (see figure 1) :

- Operator level
- User level.

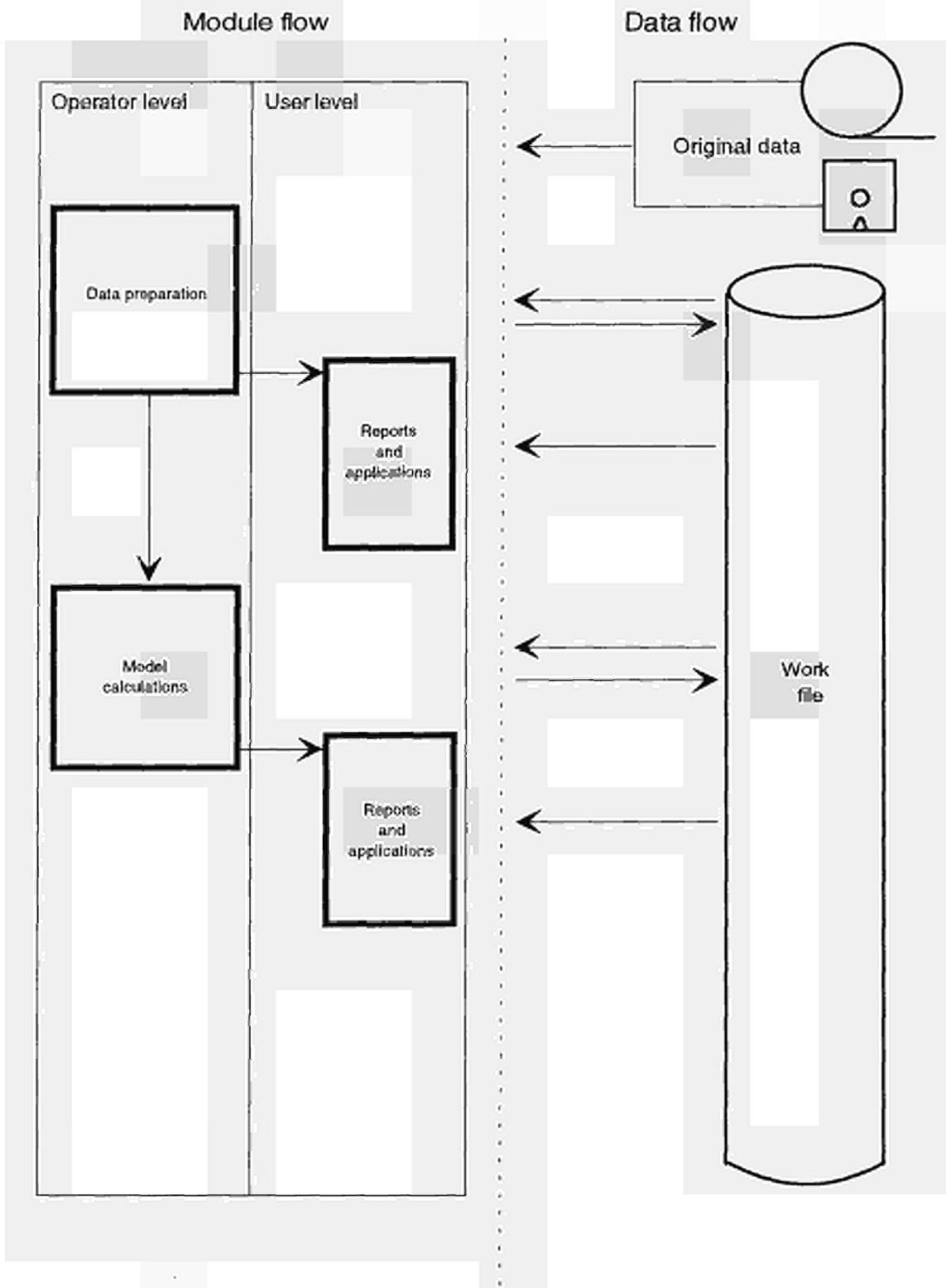
Operators are responsible for preparing original data and doing specific model calculations in line with the methodology.

Ex-post data and model results are offered to the users for reports and applications. A set of user-friendly programs is made available to users, who do not need to have detailed computer experience.

All development and revisions of SPEL software will be done by programmers.



Figure 1 : User and operator level



## 2.2. System software requirements

System software required:	IBM VS FORTRAN	Compiler
	IBM GDDM	Graphical data display manager
	VMSORT	Sort program
	IBM REXX	Restructured Extended Executor Language
	IBM XEDIT	System Product Interpreter
	MINOS	System Product Editor
		Large-scale optimization system (linear and non-linear)

## 2.3. System initialization and environment

To set up the SPEL software and SPEL-Data for one of the SPEL models, a special procedure must be called before using any SPEL program. This procedure is implementation-dependent and is installed and maintained in Luxembourg by the Luxembourg team. For users the initialisation procedure starts a program selection shell as a user-friendly guide through the work steps.

For further detailed information see the documentation of the SPEL models (e.g. SPEL/EC Model)

### 3. FORMAL STRUCTURING AND CODING

In line with methodological concepts and data flow analysis the data in the SPEL system (SPEL-Data) are logically allocated to tables, each identified by a unique alphanumeric key. This key is divided into subkeys, each directly related to six of the SPEL-Data dimensions (see Figure 2). A table is defined by two dimensions, the column and row. All in all, SPEL-Data are characterized by eight dimensions.

Each table element is thus defined by its column and row.

The number of rows and columns, their sequence and definition are called the table structure.

There are two table structures : basic table structure and complementary table structure.

The basic table structure represents the modelling ideas. It is used for any results.<sup>1</sup>

The complementary table structure is more differentiated, responding to the original data and used to constitute the basic table structure.<sup>2</sup>

#### 3.1. SPEL-Data dimensions and SPEL table subkeys

Each of the eight dimensions is assigned a methodological differentiation criterion.

The size of each dimension depends on the SPEL model; for example, the number of countries in the SPEL/EC Model is different from the number of countries in the SPEL/EC-ROW model. SPEL-Data dimensions and SPEL table subkeys correspondance is described in Figure 2.

---

<sup>1</sup> See methodological documentation SPEL/EC model Annex 1

<sup>2</sup> See methodological documentation BASE model Annex 4

Figure 2 : SPEL - Data dimensions

	Dimension	Translation rules	Table subkey	
1	Region		Region	1
2	Sub-region		Sub-region	2
3	Year		Year	3
4	Periodicity		Periodicity	4
5	Status ←	{ Base year Origin and treatment Table structure type                     }	→ Base year	5
			→ Type	6
6	Model area		Model area	7
7	Table column			
8	Table row			

The last two dimensions, table columns and table rows, are the dimensions of the SPEL tables.

The sequence and definition of the table rows and columns depend on the model.

N.B.:

In future technical revisions the dimension "Status" will be represented by a single table subkey "Status".

### **3.1.1. Region**

Regions are defined by alphanumeric codes, each of up to three characters .

For the SPEL/EC Model the official Community codes are used for the Member States .

For example :

D Germany

IRL Ireland

F France

E12 EUR 12

For the SPEL/EC-ROW model the FAO-country numbers are used.

For example :

231 USA

059 Egypt

### **3.1.2. Sub-region**

Two characters are used for the code. Currently the whole region is always used.

For example :

00 - the whole region

### **3.1.3. Year**

In the SPEL system current years are used for ex-post data and projection years are used for simulation results.

The last two digits of the year numbers are used for the code.

Years 1941 to 1999 are coded as 41 to 99. Years 2000 to 2040 are coded as 00 to 40

For example:

- 85 year 1985

- 01 year 2001

### **3.1.4. Periodicity**

Two characters are used for the code.

00 whole year

01 ... 12 - January ... December

Q1 ... Q4 - first quarter ... fourth quarter

H1, H2 -First half of a year, second half of a year

Currently the whole year is always used.

### **3.1.5. Status**

The status is defined by six-character codes. The first two characters stand for the last two digits of the base year of simulation results. For the ex-post period, the code "NN" (no base year) is always used. The next three characters specify origin and treatment, the last character identifies the table structure type : B=basic table structure, C=complementary table structure.

For example :

NNZPAB Ex-post data from ZPA1 domain and belonging to table structure B

NNEXPC Data from experts and belonging to table structure C

### **3.1.6. Model area**

Data which are specific to one model area can be distinguished by this subdimension. One character is used for the code.

For example :

D Demand (SPEL/EC)

S Supply (SPEL/EC)

T Trade (SPEL/EC-ROW)

### **3.1.7. Table column**

Table columns are defined by alphanumeric codes, each of four characters.

For example :

SWHE production activity soft wheat

PRIC producer/purchase prices (unit value)

### **3.1.8. Table row**

Table rows are defined by alphanumeric codes, each of four characters.

For example :

BARL      product barley

LEVL      production activity level

## **3.2. CODES files currently in use**

All lists of codes (except the codes for years, which are self-evident) are included in a CODES file. For the user this file is just a further item of information and can not be modified, because it is also used to control a preprocessor for the FORTRAN 77 programs.

The complete listings of the CODES files and the methodological background for the formal structuring and coding is described in the methodological documentation for each SPEL model.

## 4. DATA MANAGEMENT

For data management SPEL system distinguishes between non SPEL data file formats and SPEL standard data file formats identified by standard file name extensions. All data which are to be imported into the SPEL system have to be converted from their own original format to a SPEL standard format.

### 4.1. External data sources importation

Most of the external data used in SPEL are originally stored on magnetic tapes and each originator has its own data format.

The formats are :

Originator	Format
CRONOS	LRECL 22 (Code 18 bytes Data 4 bytes, binary floating)
FAO	FAO STANDARD FORMAT (see AGROSTAT user tape description)

### 4.2. SPEL standard data file formats

Within the SPEL system only a few file formats are used and all external file formats are converted. In the CMS operating system the following file name extensions stand for the standard formats :

SPELIMP  
SDATA  
TABFILE

Files of those formats are further called SPELIMP files, SDATA files and TABFILE files.

External data on magnetic media coming from external originators is first converted to SPELIMP format, whilst data coming from listings, printed publications, experts' findings, etc are directly edited into SDATA files. ALL SPEL-Data are finally stored in a TABFILE file, except those stored in SDATA files and being used for special program control (e.g. scenario parameters).

#### 4.2.1. SPELIMP files

The SPELIMP file is the standard type for importing external data into the system. It is optimized for fast item selection. SPELIMP files are the result of the format conversion program DATCON. Optionally the amount of data can be reduced by the FILTER program.



Data from SPELIMP files can be directly imported into the model work file(TABFILE file) by the ORIGIN import program.

### File structure :

SPELIMP files are sequential files of fixed record length with a specific record structure which is unique for all records:

```
rrrssyymmdddcccccccfvvvvvvvvvvvvvv
```

where

pos. 01-03	rrr	Region code
pos. 04-05	ss	Sub-region code
pos. 06-07	yy	Current year code
pos. 08-09	mm	Periodicity code
pos. 10-13	dddd	Originator type code (original domain name)
pos. 14-22	cccccccc	Remaining data code (up to nine characters, left aligned)
pos. 23	f	Data status flag as specified by originator if existent, otherwise blank
pos. 24-37	vvvvvvvvvv	Numerical data anywhere in the field <sup>3</sup> (e.g. 1.23456 -1234.5 12.3 E-10 123.E03 2E10 means $2 \times 10^{10}$ )

### Example of SPELIMP file:

```
-----1-----2-----3-----4--
D 008000ZPA10002345 185.
F 008000ZPA10002345 180.
...
NL 008000ZPA10002345 180.4
-----1-----2-----3-----4--
```

The records are sorted according to columns 01-22 in character ascending order.

The remaining data code, the data status flag and originator type code<sup>4</sup> are usually the same as the code used in original data sources.

<sup>3</sup> FORTRAN F14.0 format

<sup>4</sup> In FAO data, no originator type code is included. The code is set by a parameter of format conversion program (DATCON).

## 4.2.2. SDATA files

The SDATA file is used for all external data to be edited by the operator/user.

The data representation is in the form of tabulated sequential time series . SPEL table subkey codes (e.g. region , sub-region etc.) and SPEL table element codes (for column and for row) are supported. All subkey codes and table element codes are unequivocally interpreted. Additional comment lines are very helpful for the reader.

### File structure:

SDATA files are sequential files of fixed record length and record structures which differ for six record types:

### Comment records

begin with an '\*' at the first character position and may occur anywhere in the file.

### The file header record

is used to identify the file as an SDATA file. It contains

```
$STANDARD
```

at the beginning of the line and must be the first non-comment record in the file.

### Table header records

are used to mark the beginning of new time series tables.

```
$TABLE   rrrss...bbdddm
```

where :

pos. 01-06	\$TABLE	Fixed identifier
pos. 07-10		<i>Blank</i>
pos. 11-13	rrr	Region code
pos. 14-15	ss	Sub-region code
pos. 16-19	"...."	Fixed <sup>5</sup>
pos. 20-21	bb	Base year code, currently "NN" for ex-post data
pos. 22-25	dddd	Type code
pos. 26-26	m	Model area code (S=supply, T=trade, D=demand, (*) =undefined)

### Column definition records

define the codes for the data columns in the following data records. They must follow the table header record and be written before data records. Additional column definition records may occur whenever data line column codes change.

```
$COLUMNS  ccccllll cccllll cccllll cccllll  ---->
```

<sup>5</sup> Filler to have the same layout as SPEL table keys (here years and periodicities)

where :

pos. 01-08	\$COLUMNS	Fixed identifier
pos. 09-10		<i>Blank</i>
pos. 11-18	ccccIIII	Code for data column 1 cccc - model table column code IIII - model table row code
pos. 19		<i>Blank</i>
pos. 20-27	ccccIIII	Code for data column 2
...		
pos. 65-72	ccccIIII	Code for data column 7

### Data records

are used for enter numerical data.

Structure :

```
yy mm aaaaaaaaaabbbbbbbbbb ... gggggggggg -->
```

where :

pos. 01		<i>Must be blank</i>
pos. 02-03	yy	Current year code
pos. 04-05	mm	Current periodicity code
pos. 06-09		<i>Blank</i>
pos. 10-18	aaaaaaaa	Numerical data anywhere in the field must include a decimal point (e.g. 1.23456 ; -1234.5; 12.3 E-10; 123.E03; 2E10 means $2 \times 10^{10}$ ) If the field is totally blank, it is interpreted as non-existing in the file
...		
pos. 64-72	ggggggggg	Numerical data

### The end record

must be

\$END

at the beginning of the line and must be the last non-comment record in the file.

**Example of SDATA file<sup>6</sup>:**

```

-----1-----2-----3-----4-----5-----6-----7-----8
$STANDARD
*
*      Process level for Germany
*
$TABLE   D  00...00ZPAB*
$COLUMNS WHEALEVL BARLLEVL OATSLEVL
8600      99.9      99.9      99.9
8700      99.9      99.9      99.9
*
*      Process level for France
*
$TABLE   F  00...00ZPABS
$COLUMNS WHEALEVL BARLLEVL OATSLEVL MAIZLEVL OCERLEVL PARILEVL PULSLEVL
*8400
8500      99.9      99.9      99.9      99.9      99.9      99.9      99.9
8600      99.9      99.9      99.9      99.9      99.9      99.9      99.9
8700      99.9      99.9      99.9      99.9      99.9      99.9      99.9
$COLUMNS SUGBLEVL RAPELEVL SUNFLEVL
8400      99.9      99.9
8500      99.9      99.9
8600      99.9      99.9      99.9
8700      99.9      99.9      99.9
$END
-----1-----2-----3-----4-----5-----6-----7-----8

```

**4.2.3. TABFILE files**

The TABFILE files are work files and contain original data as well as results of all SPEL work steps. Storing of all intermediate results permits re-execution at any time from any point in the work step sequence.

Work files are logically divided into entries which are the "SPEL tables" described in chapter "Formal structuring and coding".

The file structure ensures an acceptable execution speed, because tables are defined to be adapted to the data flow of SPEL programs and allow minimum disk access activities and high flexibility.

Each table is identified by an unique key. A table key "UK 00800000BASB" means : Region UK, Sub-region 00 (the whole region), Current year 80 (1980), Periodicity 00 (the whole year), Base year NN (no base year), Type BASB (result of work step BASEMOD, basic table structure), Model area S (supply).

General maintenance for TABFILE files can be done by the menu-driven utility program DASERV. It offers services for creating new files, listing the keys of all entries, printing entries, reorganizing files, copying and renaming entries as well as many others. New TABFILE files must be created by DASERV because SPEL programs can only access already existing TABFILE files.

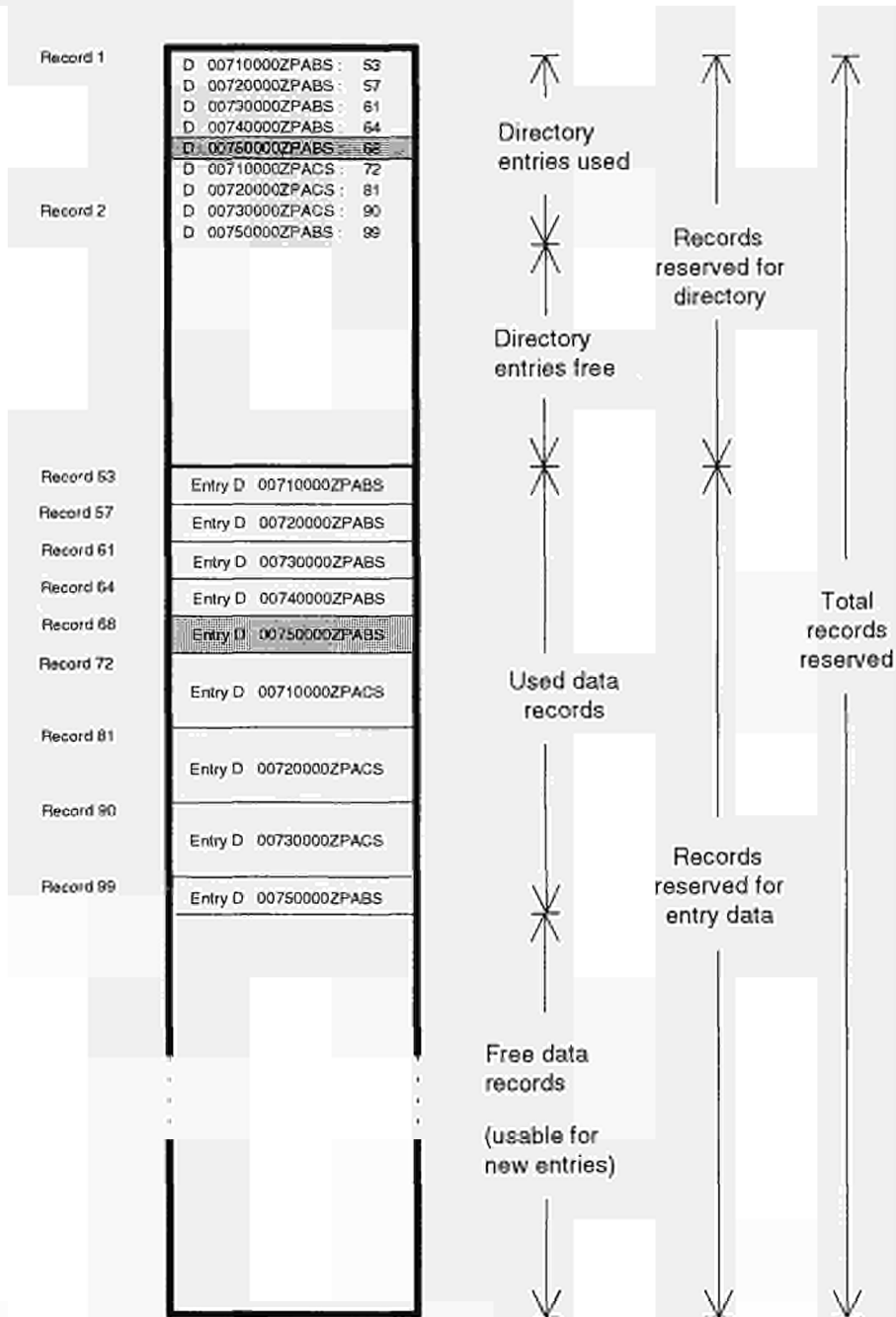
All access of programs to these TABFILE files is done by the TABFILE file manager, which forms part of the SPEL programs.

<sup>6</sup> The values in the table are for illustration only

A SPEL user never has to consider the format or internal file structure but has to consider the following:

- A file is static in its size, which is definitively fixed when created. Estimates of the maximum number of entries and the size of entries to be stored must be made when the file is created.
- Before accessing a new file by SPEL programs, the key structure should be defined or copied from another TABFILE file by using the appropriate DASERV service.
- A new entry is implicitly created during the first writing operation using the corresponding table key. The file size occupied by the entry is determined by the key size, an optional comment field size and the number of non-zero table elements. Zero elements in "sparse" tables do not occupy any disk space. The file size occupied by an entry is always continuous.
- During subsequent writing with the same entry key, the same location will be used as long as the new entry size fits the old one. If the new size exceeds the space reserved for the already existing entry, the latter is implicitly deleted and a new entry created.
- File areas occupied by entries implicitly or explicitly deleted by user action cannot be used immediately for new entries. This "unused space" must be freed by running a service of the DASERV utility.
- Utilization factors for numbers of entries in the file directory and for the stored data records are printed at the end of the log file of application programs; they can also be listed by the DASERV utility.
- To enlarge an existing table file, a new table file of sufficient capacity must first be created. The entries in the old table file can then be copied into the new file by using the DASERV utility.

Figure 3 : Structure of TABFILE files



Notes:

An entry may consist of text or numerical data or both. The numerical data part is an array of data elements. In SPEL it is a table of size rows by columns; the text part is comment (e.g. including creation data and time). The table is interpreted as a vector of length rows multiplied by columns. To minimize storage space, the vector is packed (e.g. zero data are removed) and stored as a flow without any regard for record boundaries.

The entry references in the directory formed from the entry key and the number of the first entry record) are also stored as a flow.

In this figure "deleted" entries are marked by shading.

## 5. USER INTERFACE

The user interface comprises panels, general control files and special control files.

Special control files permit large sets of control statements which cannot be modified during run-time.

### 5.1. General program control

General program control is done by parameters which can be modified during run-time by panels. Presets for all these parameters are loaded from general control files.

#### 5.1.1. General control files (PARAM files)

General control files have the name extension PARAM, further called PARAM files. Once the SPEL system is implemented a DEFAULT PARAM file is installed by the operator. For all SPEL programs default presets for the program control parameters are included. The operator and user can create their own version with adapted parameters for special program applications. For further information see the following chapters on General program control.

PARAM files are sequential files of a fixed record length of 80 with a fixed record structure in free format.

Control parameters are grouped into parameter blocks according to their usage.

There are four different record structures:

##### **Comment records**

begin with an '\*' at the first character position and may occur anywhere in the file.

##### **The parameter block header record**

includes the name of the parameter block. It is a character string of up to 20 characters in length. It indicates the beginning of a parameter block.

##### **Parameter definition record(s)**

include(s) the assignment(s) of a parameter value to a parameter name.

$$\begin{array}{l} name_1 = value_1 \\ \cdot \\ \cdot \\ name_i = value_i \end{array}$$

where name and value are character strings. The whole assignment must not exceed 80 characters. The parameters belong to a block defined in the previous block header record.

**Parameter block end record**

The end line must be  
'END'

**Example of part of a PARAM file:**

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
*
#SPEL
System text =SPEL
Subkeys 1='Region' 'Sub-region' 'Current year' 'Periodicity' 'Base year'
'Type' 'Model area'
Subkey 2 length= 3 2 2 2 4 1
Model areas =D S T
Area text D =Database for model area DEMAND
Area text S =Database for model area SUPPLY
Area text T =Database for model area TRADE
Base year of reference = 85
First year of reference = 73
Additional parameter blocks = #DEFAULT
END
*
#DEFAULT
Protocol file =SPEL LOG C
Error file =SPEL ERROR C
Global help file =SPEL HELP *
END
*
ORIGIN
Program text =Import original data
Local help file = ORIGIN HELP *
Table file = SPEL-USR TABFILE A
Model area = S
Assign file = * ASSIGN *
Input file 1 = * SDATA *
Input file 2 = * SDATA *
Region = D F
Current year = 72 : 90
Type = ZPAB ZPAC EXPB EXPC COSC COSB PRAC PRAB
Additional parameter blocks = AREAS #SPEL
END
*
...
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----

```

**5.1.2. Panels**

Panels are screens allowing user interaction or showing messages.

Three kinds of panels can be distinguished :

- edit panels (control parameter and data edit)
- selection panels (item selection with cursor selection facilities)
- display panels (messages, help and data display).



When calling a program, the user will get a sequence of screen panels to verify and optionally update all program control parameters. Presets for input fields and top line text, except control level text, come from PARAM files. The panel layout is standardized.

The text which appears in bold in the panels can be either overtyped by the user or selected by cursor for the current run. The following examples show typical panel layout.

### Edit Panel

The top line of the panel shows on the left the purpose of the running program, in the middle the current software name and possibly the version, and on the right the control level. The control level text issued by the program may be replaced by error messages and warnings. The active function keys (PF-keys) are displayed at the bottom.

The centre of an edit panel contains parameter fields with the parameter description on the left and the parameter value on the right. The bold-printed parameter values can be updated by user input. By pressing the ENTER key the input is tested for acceptability. When errors are detected a short error message will appear at the top right and the cursor will be placed on the erroneous input for correction.

### Example: Work file selection

```

Import original data ----- SPEL ----- Work file selection
                                Select work files

Table file   => SPEL-USR TABFILE A
Assign file  => * ASSIGN *
Input file 1 => * SDATA *
Input file 2 => * SPELIMP *

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

Between the top line and the PF-key description lines additional comment lines may appear (e.g. "Select work file").

The example panel is used by the program ORIGIN to control all files used. The file names are tested for conformity and processing mode.

In any file selection panel file names may be specified using wildcard characters such as "\*". If a file name contains wildcards, a list of matching files is displayed in a special screen (anel). Selection is effected by placing the cursor on the desired file name line and pressing ENTER.

## Selection Panel

The top and bottom lines have the same structure and the same functions as in the edit panel. In the centre of a selection panel there is a list of character items; below, a list of ASSIGN files is shown. To select one of these items the user has to put the cursor on the item line and press ENTER.

### Example: Work file selection

```

Import original data ----- SPEL ----- Work file selection

                Assign file
                File name selection

                IMP      ASSIGN  B1
                SDATA    ASSIGN  B1
                SUPPLY   ASSIGN  D1
                SUPPLY   ASSIGN  D1
                DEMAND   ASSIGN  G1
                DEMOLD   ASSIGN  G1
                PORTUGAL ASSIGN  G1
                HELFEE   ASSIGN  H1
                HELFEE1  ASSIGN  H1

                Select a file by moving the cursor to any file name above

                Enter= ok  1= Help  3= Quit

```

## Display Panel

Display panels show only text. Normally they are used to display help texts (see chapter "On-line help information") or listings of protocol and error files.

### Example: Display of a log file

```

* Working for Member State: D  sub-region: 00
-
**** Get regression steering file
-
*** 240 records selected from regression steering file

-
**** Get time series
-
*      Working for type: ZPAC
-
*      Working for type: COSC
-
*      Working for type: PRAC
-
*      Working for type: EXPC
-
*      Working for type: ESTC
-

Enter= ok  3= Quit  4= Find  6= Bottom  8= Page down  11= 1/2 p.down

```

In the example above an error has occurred in running the ORIGIN program. The user has pressed the help key and the protocol file is displayed to give more detailed information.

**List of default PF-keys :**

	<b>Usage of General PF keys</b>	<b>Usage of Paging facilities PF keys</b>
PF 1	Help (call Help panel)	-
PF 3	Quit (back to previous control level)	-
PF 4	Exit (program exit)	Find (search for keyword)
PF 5	-	Top (top of file)
PF 6	-	Bottom (bottom of file)
PF 7	-	Backward (one page back)
PF 8	-	Forward (one page forward)
PF 9	-	Left (go to left side)
PF10	-	Right (go to right side)
PF11	Save/Load (save or load parameters)	-
PF12	General files (modify name of general file)	-

The bottom lines of each panel show the available PF-keys. For some applications the meaning of PF-keys may change (e.g. PF4). PF1 (Help) and PF3 (Quit) are invariable.

### 5.1.3. On-line help information

Context-sensitive on-line help is available for each panel field. The user merely has to place the cursor on any displayed text and press the help key (PF1).

On-line help is also available for error messages currently displayed. After pressing the help key more detailed information is displayed in a display panel.

If the user has placed the cursor on the text "Table file" in the edit panel example given above and pressed PF1 the following screen will be displayed.

#### Example : Help information screen

```
Help informations
-Table file
  This file is used for output and must be of TABFILE format (SPEL work file)
  All data which fit the selection criterion (next panel) will
  stored as 'table' entries in this file
  To select a file from a list of existing files please use
  wildcard characters such as *** (e.g. ** TABFILE **)
  For further detailed information about SPEL data file formats
  see "Data Management" chapter in the Technical Documentation.
Enter= ok  3= Quit  4= Find
```

To search for keywords in a large help text a "Find" key (PF4) is defined. When the information does not fit on to one screen, paging facilities are offered (PF7 - PF8).

All help texts are collected in HELP files and are used by the on-line help system.

Two HELP files are accessed by each program :

- the local HELP file  
includes help information which are specific to the program
- the global HELP file  
includes help information used by several programs.

The names for the HELP files are defined in the parameter file.

The user/operator neither needs to know the structure of the HELP file type nor has to change any text in those files.

#### 5.1.4. Parameter file link panels

As already mentioned above, presets for all control parameters ever used in panels are loaded from general control files, which are structured in named parameter blocks. The DEFAULT PARAM file is used when the program is called. If the user wants to use an own PARAM file, he can press the SAVE/LOAD-key to do this. Every main program has its own parameter block, the name of which is similar to the program name. Inside the parameter block all program-specific control parameter names are in accordance with the input field description displayed on the screen.

General parameters, used for most SPEL program panels, like protocol file names, are collected into separate parameter blocks. Access to those blocks can be provided by the special parameter 'Additional parameter blocks' defining additional parameter blocks to be loaded. The parameters appearing in the parameter block corresponding to a main program will always have priority, if the parameter is also defined in an additional parameter block.

If a parameter is not found, the related panel input field preset is blank.

In the example in the chapter "General control files", part of a PARAM file is shown. The parameter block called ORIGIN is the one specifically used by the ORIGIN main program. The last parameter 'Additional parameter blocks' extends access to the '#SPEL' parameter block.

If the user has changed some parameter values while revising the control panels, he can store the new parameter settings by pressing the PF11=Save/load key. An edit panel will be displayed to define the name of the parameter file. All parameter values will be updated in the blocks they come from, with the exception of those coming from protected blocks with a name beginning with "#", like "#SPEL". If protected parameters have changed, they will be added to the program block. In the example the parameter "Protocol file" originally in the "#DEFAULT" block will be saved in the "ORIGIN" block, if the value has changed.

#### Example : Save/load panel

```

Import original data ----- SPEL ----- Save/load parameter
                                Please enter PARAM file name

Parameter file => DEFAULT PARAM F

                                1= Help  3= Quit  4= Exit  11= Save  12= Load

```

If the user has decided to load another parameter scenario while revising the control panels, he can press the PF11=Save/load key too. An edit panel will be displayed and the user can define the name of a new parameter file to be loaded. The program will start again at the first control panel.

## 5.2. Special control files

The programs documented in this volume often use a large amount of control information. It is quite clear that all these parameters should not be handled by panels. Special control files are allocated for retrieving them.

Four kinds of special control files are supported using the following file name extensions:

ASSIGN	files	for data selection used by the FILTER and ORIGIN programs
REGSTE	files	for trend estimation control used by the TREND program
STEER	files	for the EV evaluation utility.
SDATA	files	for special program control

### 5.2.1. ASSIGN files

The original data have to be selected for SPEL usage. General selections for regions, sub-regions, years, periodicity and types are done by control parameters coming from PARAM files revised by panels.

The ASSIGN format was created for selecting thousands of source data codes and assigning them to SPEL table element codes. The assignments are specified only once for each table structure and are implicitly used for all tables matching the general selections (regions, sub-regions ...).

ASSIGN files are sequential files with a fixed record length of 80 and fixed record structure in free format.

#### File structure :

There are two record types :

#### - Comment records

Beginning with an "\*" at the first character position. Comments may occur anywhere in the file

#### - Assignment records

Each record contains one assignment statement of the following statement structure :

```
i ccccrrrr = ttttdd...dd fff....fff;
```

where

<i>i</i>	Identifier for table structure <i>i</i> = B means Basic table structure <i>i</i> = C means Complementary table structure
<i>cccc</i>	SPEL table column code
<i>rrrr</i>	SPEL table row code
<i>tttt</i>	Type code (domain)
<i>dd...dd</i>	Data code, up to 9 characters (see SPELIMP files)
<i>fff....fff</i>	Scale factor for all data belonging to this code, when the data are assigned to SPEL table element codes. If no factor is specified, 1. is assumed
<i>;</i>	Semi-colon used as an assignment delimiter.

For ad hoc selections without any assignments to SPEL table element codes, a short form of assignment statements can be used as follows :

```
ttttdd...dd;
```

The remaining text following the delimiter is treated as a comment and ignored.  
The interpretation of the assignment is related to its use (see ORIGIN and FILTER programs).

#### Example of an ASSIGN file:

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
*Example
B SWHELEVL = ZPA13112020 ;
COSA0013406;
C PR01SWHE = PRAG3221120 10.;
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

### 5.2.2. REGSTE files

REGSTE format is used to steer trend estimations in the TREND program. The files are sequential files with a fixed record length of 72.

#### File structure:

There are three record types:

#### - Comment records

Beginning with an '\*' at the first character position. Comments may occur anywhere in the file.

#### - Definition block records.

This block is optional, but if used it must appear on top of the file. It specifies how to handle SPEL table elements not explicitly defined in the files regression steering records. The block consists of three records with identifiers beginning in the first column:

First record	DEFINE
Second record	INCLUDE or EXCLUDE
	INCLUDE All SPEL table elements are implicitly included in the regression and trend estimation
	EXCLUDE All SPEL table elements are implicitly excluded from the regression and trend estimation
Third record	END

If no definition block exists, INCLUDE is assumed.

**- Regression steering records**

All records have the following unique structure :

*sss xx cccrrrr dddiii ff:mm:ll*

where

pos. 01		must be blank
pos. 02-04	sss	Region code
pos. 05		<i>Blank</i>
pos. 06-07	xx	Sub-region code
pos. 08		<i>Blank</i>
pos. 09-16	cccrrrr	Table element code (col/row)

For all these positions wildcard characters such as '\*' may be used.

The next positions are optional

pos. 18-23	dddiii	Data transformation type ddd dependent variable (aligned left) iii independent variable (aligned left) <i>Blank</i> means automatic selection by best fit with the following data transformation types: LINLIN LININV LN INV LN LN LINLN LN LIN
pos. 24		<i>Blank</i>
pos. 25-32	ff:mm:ll	ff, mm and ll are limits of the reference interval The interpretation is application dependend.
pos. 33		<i>Blank</i>
pos. 34-72		Reserved for special extensions, e.g. applications in work sector SFSS of the SPEL/EC Model. All extensions are deccribed with the documentation of these applications.



The data transformation types for "ddd" and "iii" are as follows:

LIN	Linear (no transformation)	$x' = x$
LN	Natural logarithm	$x' = \text{LOG}_e(x)$
LOG	Logarithm base 10	$x' = \text{LOG}_{10}(x)$
SQR	Square root	$x' = \sqrt{x}$
REZ	Inverse	$x' = \frac{1}{x}$
EXP	Exponential	$x' = e^x$
POW	Power	$x' = 10^x$
QUA	Square	$x' = x^2$

### Example of a REGSTE file

```

-----1-----2-----3-----4-----5-----6-----7--
*
* define default modus
DEFINE
INCLUDE
END
* regression steering definitions
* (only format examples)
D 00 SWHELEVL LOGLIN 75:80:87
F 00 SWHELEVL LN LIN 76:80:87
NL 00 * * LINLIN
E 00 * * 80:85:90
*
-----1-----2-----3-----4-----5-----6-----7--

```

### 5.2.3. STEER files

These control files include the statements of the control language for utility EV. The files are sequential files with a fixed record length of 80.

A detailed description of the EV control language is given below.

### 5.2.4. SDATA files

SDATA files for special program control have the same structure and format as SDATA data files

The definitions for the tables in the file may be different, e.g. the tables may be not time series tables. The application dependent interpretation is described with the specific program documentation.

## 6. SPEL PROGRAMS

All SPEL programs are executed in a fixed order depending on the SPEL model. This volume only includes the description of multi-purpose programs and utilities which are used in all SPEL models.

The application sequence and all methodologically adjusted programs are described in the specific model documentation (e.g. SPEL/EC Model documentation).

### 6.1. Multi-Purpose Programs

There are two programs (the ORIGIN program and the TREND program) which have been specially designed for SPEL use and which have model based applications in different work steps.

#### 6.1.1. Importing standardized sequential files into TABFILE work files (ORIGIN)

Whenever data from sequential files are to be imported into tables in a TABFILE work file, the ORIGIN program is used.

Up to eight SPELIMP and/or SDATA input files are supported.

A selection of input data is presented for each differentiation criterion (e.g. region, year, etc.). The user may define selection ranges.

If all input files are of the SDATA format, it is not necessary to specify any ASSIGN file. If at least one of the import files is of the SPELIMP format, then an additional ASSIGN file must be allocated.

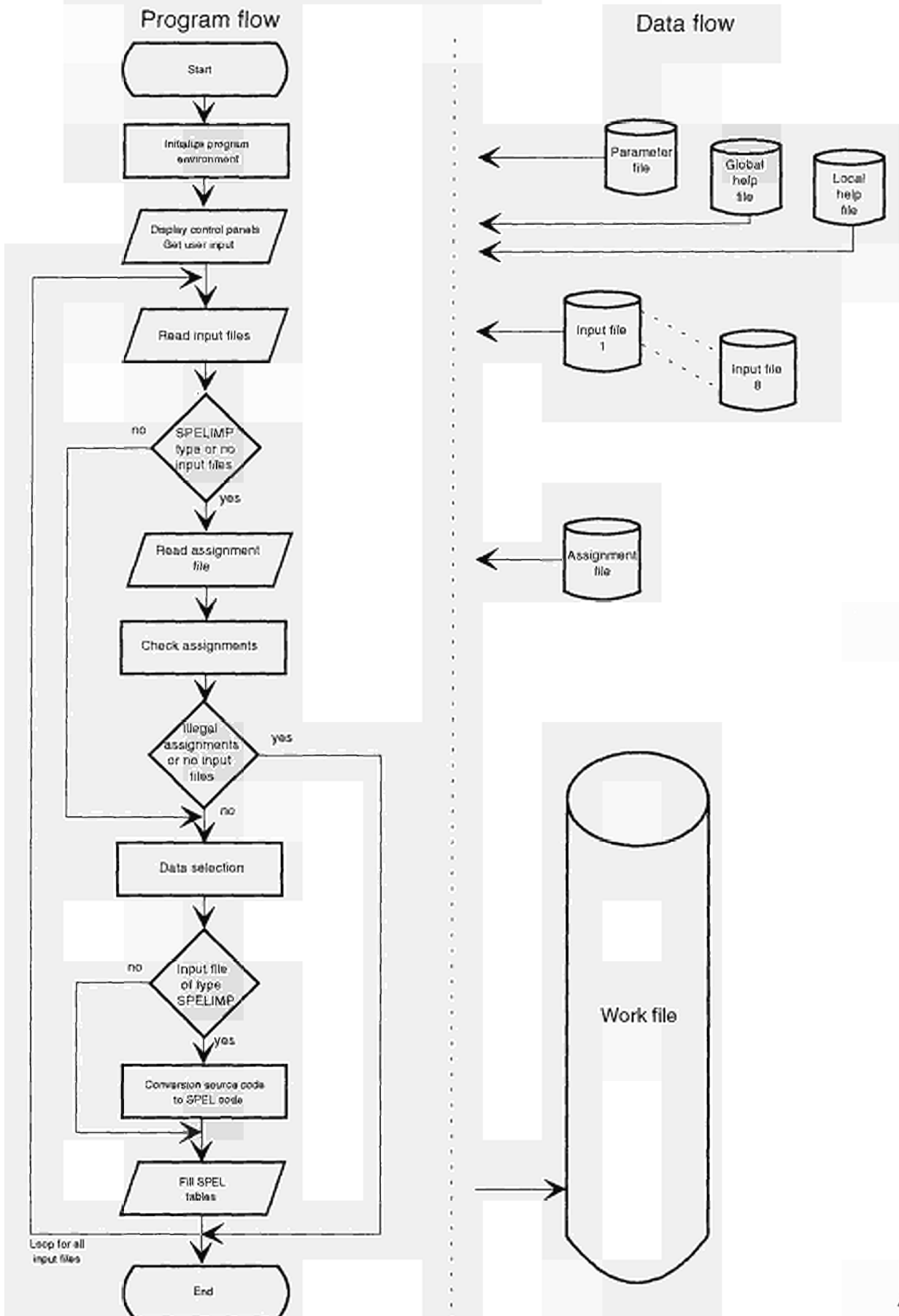
The ASSIGN file is used for :

- selection of import data by codes,
- compilation of the original source codes into SPEL table element codes,
- assignment of a table structure.

Only one assignment file can be defined for all SPELIMP input files. The set of all assignment statements in the ASSIGN file illustrates the unambiguous relationship between original source data and SPEL table elements.

ORIGIN can also be used to check an assignment file for legal SPEL table element codes. In this mode the program does not process any data.

Figure 4 : ORIGIN program flow chart



## Module startup screen

```
Import original data ----- SPEL ----- Module startup logo

      I M P O R T   D A T A   I N T O   S Y S T E M   D A T A   B A S E

Get original data from files of SDATA or SPELIMP format,
select data items with ASSIGN files (if original format is SPELIMP),
select table keys by dialog input,
store tables in a TABFILE.

      Current settings for general files:
      Run time protocol : SPEL LOG C
      Error messages    : SPEL ERROR C

      Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key, an edit panel will be displayed which will permit the changing of names of general files for the run time protocol (LOG-file) and error messages (ERROR-file).

In the LOG-file the user will find a copy of all messages displayed on screen during program execution. More detailed information about work steps will be included as well.

The ERROR-file contains the messages of any error conditions which may have occurred and which might have caused the program to stop prematurely.

It is always useful for the program user to check both general files after the program has ended.

### Parameters:

None.

**Work file selection screen**

```

Import original data ----- SPEL ----- Work file selection
                                Please enter file names

Table file   =>
Assign file  =>
Input file 1 =>
Input file 2 =>
Input file 3 =>
Input file 4 =>
Input file 5 =>
Input file 6 =>
Input file 7 =>
Input file 8 =>

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Table file**

This file is used for output and must be of TABFILE format (SPEL work file).

All data which fit the selection criteria (next panel) will be stored as 'table' entries in this file.

**Assign file**

This file contains a set of assignment statements for selecting source data and converting source data codes to SPEL table element codes.

The ASSIGN file must be used if at least one of the input files is of the SPELIMP format. Only one ASSIGN file can be defined for all SPELIMP input files.

**Input file 1 ... 8**

File of one of the following two formats:

- SPELIMP
- SDATA

The program identifies the file type automatically by analysing the file structure.

**Remark :**

If only the ASSIGN file name is specified, ORIGIN will just check the set of assignment statements for legal SPEL table element codes. The set to be tested has to be specified by user's selections in the 'Table key selection' panel (next panel).

**Table key selection screen**

```

Import original data ----- SPEL ----- Table key selection
      Please make your key selection for output table file
      Subkey for base year is protected

Region      ( 3 ch.) => B : UK
Sub-region  ( 2 ch.) => 00
Current year ( 2 ch.) => 85
Periodicity ( 2 ch.) => 00
Base year   ( 2 ch.) => NN
Type        ( 4 ch.) => ZPAB ZPAC EXPB EXPC
Model area  ( 1 ch.) => S

      Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumeric expansion)
- a range defined by region codes such as "D - UK"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

The maximum number of regions that may be selected is 400.

**Sub-region**

Sub-regions are currently not used.

The whole region has the sub-region code "00".

Always specify "00".

**Current year**

Current year selection. Specify one of the following:

- a sequence of years to be selected such as "70 75"

- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"

All codes and the range operand ":" must be separated by blanks.

The maximum number of years that may be selected is 50.

### Periodicity

Periodicity selection. Specify :

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
(alphanumerical expansion)
- a range defined by periodicity codes such as "Q1 - Q4"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "01 : 06 Q4"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

The whole year has the code "00". Other periodicity codes are currently not used.

When using periodicities the maximum number to be selected is 19.

### Base year

For putting original data into the SPEL work file the code for the subkey base year will be always "NN".

The subkey is protected for selection.

### Type

Type selection. Specify one of the following :

- a sequence of type codes to be selected such as "ZPAB COSC"
- a range defined by lower and upper bounds such as "AAAA : 9999"  
(alphanumerical expansion)
- a range defined by type codes such as "ZPAC - COSC"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "ZPAB : COSC EXPC".
- use '\*' to select all codes as defined in the CODES file.

All codes and the range operands ":" and "-" must be separated by blanks.

The maximum number of types that may be selected is 10.

### Model area

Select one of the following model area codes :

"S" for supply,  
 "D" for demand,  
 "T" for trade

### Check selection screen

```
Import original data ----- SPEL ----- Check selection
A T T E N T I O N: Make sure that you have specified all input files which are
necessary to fill the tables for the specified types. Otherwise there will be
data missing in the tables, at the worst they will contain only zeros, even if
those tables have been complete before. Please check the current selection.

Types to be created: ZPAC
Input files          : CORR      SDATA      A

Select an action by moving the cursor to any topic below

Continue doing the selection
Revise selections for types/input files

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

The user may once more reflect if all input files which are necessary to fill the tables for the selected types are specified. Otherwise there will be data missing in the tables, at the worst they will contain only zeros, even if those tables have been completed before.

#### Parameters:

Continue doing the selection

Program execution continues selecting the input data for specified types.

Revise selection for types/input files

Program returns to the "Table key selection"/"Work file selection" panel for new selections.



### 6.1.2. Trend estimations (TREND)

The TREND program does trend estimations by single regressions using ordinary least squares (OLS). It is designed to work with simple control for a large quantity of time series.

Input data comes from tables stored in a TABFILE work file. Estimation results are again stored in a work file.

Default parameters for regression control, the reference period and "best fit" criteria for automatic selection come from a PARAM file revised by panels.

On default for each time series, six OLS estimations are calculated with different data transformations for the dependent variable and the trend. The program automatically selects the results of the transformation which provides the best fit. Best fit criteria may be the coefficient of determination or Theil's coefficient <sup>7</sup>. The data transformations are:

	<b>dependent</b>	<b>trend</b>
1.	linear	linear
2.	linear	inverse
3.	logarithmic	inverse
4.	logarithmic	logarithmic
5.	linear	logarithmic
6.	logarithmic	linear

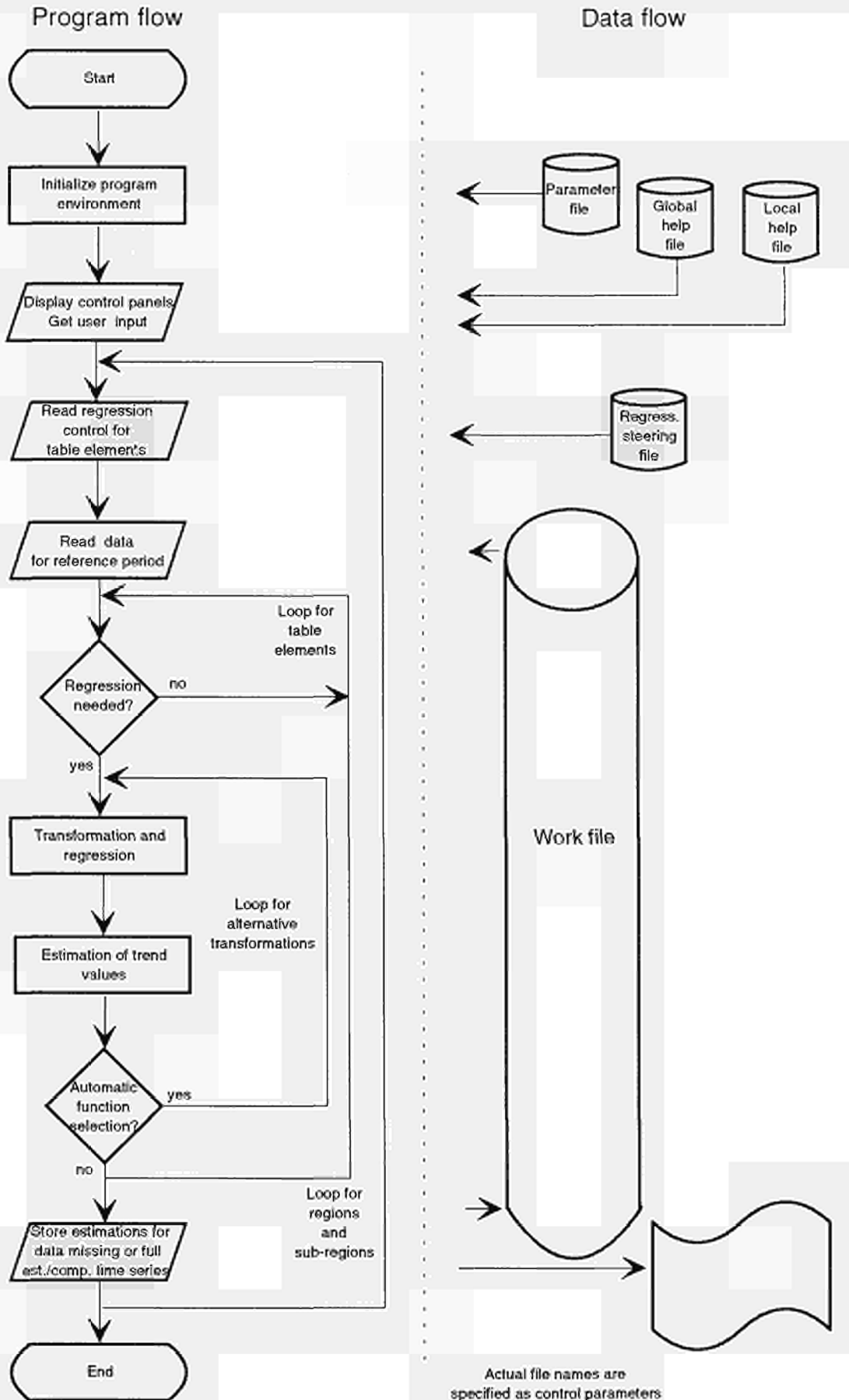
Special regression control parameters for specific time series may be selected in the REGSTE control file.

In the SPEL system, TREND is used for filling in missing data in time series as well as for doing trend estimations for simulation years.

---

<sup>7</sup> For further information see methodological documentation

Figure 5 : TREND program flow chart



**Module startup screen**

```
Trend estimation ----- SPEL ----- Module startup logo
  
T R E N D   E S T I M A T I O N S   B Y   S I N G L E   R E G R E S S I O N
  

  
Check and generation of data completeness in
time or projection for simulation years.
  

  
Current settings for general files:
Run time protocol : SPEL LOG C
Error messages    : SPEL ERROR C
Print file        : SPEL LISTING C
  

  
Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key a special panel will be displayed which will permit changes to names of general files for run time protocol (LOG-file), error messages (ERROR-file) and documentation file (LISTING-file).

In the LOG-file the user will find all messages ever displayed on screen during a program run. More detailed information about working steps will be included too.

The ERROR-file contains the list of errors which have occurred doing data transformations, regressions and trend estimations for some time series.

The detailed protocol for all estimation steps is stored in the LISTING-file

It is useful for the program user to check first the LOG- and ERROR-file after a program has ended to get a rapid overview of the success of estimations.

**Parameters:**

None.

### Work file selection screen

```
Trend estimation ----- SPEL ----- Work file selection
                               Please enter file names

Regression steering f. => FEED1 REGSTE F
Table file              => SPEL-USR TABFILE A
Add.table file (input) =>

                               Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

#### Parameters:

##### Regression steering f.

This REGSTE file contains explicit definitions for data transformation types and reference intervals for selected time series.  
The reference intervals are defined in the the panel "Regression steering" which follows.

##### Table file

This file is a direct access file containing SPEL tables for entries and is of TABFILE (TABFILE work file) format.

##### Add. table file (input)

Additional TABFILE input file only opened for read access. This file is to be accessed for tables not found in the first defined table file.

#### Remark:

If only the regression steering file name is specified, TREND will just check a set of trend estimation steering statements for legality. The set to be tested has to be specified by user's selection in the "Table key selection" panel (next panel)

**Table key selection screen**

```

Trend estimation ----- SPEL ----- Table key selection

                Please make your selection
                Format for table types:  outputtype= inputtype1 inputtype2  ....
                e.g.:  COMB              = ZPAB COSB
                or   :  COMC              = ZPAC COSC

Region           ( 3 ch.) => D F
Sub-region       ( 2 ch.) => 00
Current year     ( 2 ch.) => 72 : 89
Periodicity     ( 2 ch.) => 00
Table types      ( 4 ch.) => COMC = ZPAC EXPC RSTC
Model area       ( 1 ch.) => S

                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical expansion)
- a range defined by region codes such as "D - UK"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

The maximum number of regions that may be selected is 400.

**Sub-region**

Sub-regions are currently not used.

The whole region has the sub-region code "00".

Always specify "00".

### Current year

Current year selection. Specify one of the following:

- a sequence of years to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"

All codes and the range operand ":" must be separated by blanks.

The maximum number of years that may be selected is 30.

### Periodicity

Periodicity selection. Specify one of the following:

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
(alphanumerical expansion)
- a range defined by periodicity codes such as "Q1 - Q4"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "01 : 06 Q4"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

The whole year has the code "00". Other periodicity codes are currently not used.

When using periodicities the maximum number that may be selected is 19.

### Table types

Selection of table type codes for output and input tables

Syntax:

output type = inputtype1 inputtype 2 ....

For outputtype specify one legal table type code.

For inputtype1, inputtype2 ..... specify one table type code or sequence of codes to be selected such as "ZPAC EXPC".

The maximum number of input type codes that may be selected is 10.

### Model area

Select one of the following model area codes:

- "S" for supply
- "D" for demand
- "T" for trade

**Regression steering screen**

```

Trend estimation ----- SPEL ----- Regression steering

      Please select parameters for regression steering

Regression interval      (start : end1 : end2) => 72 : 85 : 89
Significance coefficient  (THEIL/DETERMINATION) => DETERMINATION
Documentation level      (FULL/ROUGH/MINIMUM/NONE) => MINIMUM
Output level            (COMPLETED/FULL TREND/MISSING/NONE) => COMPLETED

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Regression interval**

Define the years for short and long regression reference intervals.

**Syntax:**

start : end1 : end2

start : end1 will be the short interval

start : end2 will be the long interval

For the short interval you must specify at least 4 years

end1 and end2 may be the same year.

The start year has to precede the end years.

Example: 72 : 80 : 89 or 80 : 90 : 90

### Significance coefficient

Select one of the following for best fit criteria :

- THEIL - Theil's coefficient <sup>8</sup>
- DETERMINATION - coefficient of determination <sup>9</sup>.

### Documentation level

You can choose between the following documentation levels:

"FULL" provides, for each time series, documentation on

- steering parameters
- original values for the whole of reference and output period
- results of all data transformation alternatives
- the selected transformation type
- the completed time series.

"ROUGH" provides, for each time series, documentation

- on steering parameters
- original values for the whole of reference and output periods
- the completed time series.

"MINIMUM" provides, for each time series, documentation on

- original values for the whole of reference and output periods
- the completed time series.

"NONE" nothing will be documented.

### Output level

For output into the work file you can choose between four levels:

COMPLETED: time series including original values and estimates of missing data.

FULL TREND: full trend time series

MISSING : only the estimated values for missing data

NONE: nothing

---

<sup>8</sup> THEIL =  $\frac{\text{Sum of squares of deviation}}{\text{sum of squares of yearly changes}}$

Source (THEIL): Henri Theil, Applied Economic Forecasting, Amsterdam 1971, page 28

<sup>9</sup> DETERMINATION =  $\frac{\text{Sum of squares explained}}{\text{sum of squares total}}$



## 6.2. Utilities

Utilities are general programs not specially designed for SPEL. Using control parameters of SPEL they are applied to several work steps.

### 6.2.1. General estimation and evaluation program (EV)

#### 6.2.1.1. Application abstracts

The EV program offers general evaluations and single equation regression estimations based on the standard file formats TABFILE and SDATA supported by the SPEL system.

EV has its own command language for both batch and interactive operations. In interactive mode the commands are entered directly from the keyboard and executed immediately. For batch operations the commands are collected in a STEER control file.

The simplest use of the EV program is to obtain data, e.g. the production of wheat in a specific period for specified sequences of regions (e.g. Member States). Such data are provided in the form of tables for printing or listing on the screen. Growth rates and averages, etc. can be calculated.

The program also permits the analysis of data in the econometrical sense based on single equation regressions with one or more independent variables. Quality indicators of such regression estimates can be accessed on the basis of the coefficient of determination, correlation coefficient, T-value, F-value, standard of estimation error, Durbin-Watson coefficient, Theil's coefficient, etc.

In combination with large sets of commands collected in STEER files it is also used to carry out certain work steps of the SPEL system.

Most of the EV operations are defined at the "EV-tables" level.

These tables are not the same as SPEL tables in a SPEL work file. EV-tables normally have years as rows and regions as columns. The selection of years and regions is identical for all tables defined at the same time. Tables can be transposed by inverting the rows and columns. Each table is described by a code and, optionally, a descriptive text.

In SPEL the code of an EV-table is normally the code of an element of a SPEL table. In other words, all elements of one EV-table correspond to one (always the same) element of a set of SPEL tables. The number of SPEL tables in the set is equal to the number of elements in the EV-table. E.g. the data for wheat production for the years 80 to 85 and the regions D, F and NL form an EV-table, where the source data is obtained from 18 (= 6 years \* 3 regions) SPEL tables.

The selection of STEER control files may be done using a program selection shell. For further informations see documentation of the SPEL models (e.g. SPEL/EC Model).

### 6.2.1.2. Types of data files

The EV program supports TABFILE files and SDATA files. TABFILE files should be preferred because reading and writing is faster.

TABFILE files :

SPEL-Data to be processed are kept in TABFILE files containing various tables. They are distinguished by table types according to data source, scope and treatment.

SDATA files :

SDATA files may contain both SPEL-coded data not stored in a TABFILE file and non-SPEL coded data.

### 6.2.1.3. Control language structure

The available commands are divided into "main commands" and "subcommands". Subcommands can be used only when the corresponding main command was the last main command entered. Main commands have the form "\$.....", without any additional instructions.

The subcommands begin with a fixed key word followed by additional parameters separated by at least one blank.

Text parameters containing blanks must be enclosed in quotes.

**An EV-session is normally structured as follows:**

Definition of general control parameters:

General control parameters e.g. selection of regions, first and last year of time series, type of SPEL tables to be processed etc. are defined (see subcommands of \$CONTROL).

Definition of EV-variables (tables) :

The variables to be used are defined by specifying variable codes. The variables are handled as "EV-Tables" (see subcommands of \$NEW and \$DEFINE).

Data input :

SPEL-Data are read from the TABFILE work file (see subcommand of \$MATRIX). Other data are read from SDATA files or embedded in the EV commands (see \$DATA subcommands).

Data treatment :

Data treatment generally affects whole EV-tables. The tables are identified by a single number equal to its position in the sequence of table definition statements. Various arithmetical operations are offered to convert the data into the required form for preparing the printer output or the estimates (see \$ARITH subcommands). \$REGRESS subcommands are used for regression calculations. The variables to be entered in the regression may be transformed to fit several curve forms.

Data output :

The original data and results can be shown on screen or printer (\$TABOUT) or stored in either a TABFILE file or in a sequential SDATA file.

#### 6.2.1.4. Control language details

Main command for general parameter definition

\$CONTROL	Set and list general control parameters
-----------	---

Main commands for on-line help and program flow control

\$HELP	On-line help for main commands
\$SKIPON	Enable skipping of the following command statements (until the next \$SKIPOFF occurs)
\$SKIPOFF	Disable skipping of command statements
\$STOP	Exiting from EV-session

Main commands for defining variables

\$NEW	Definition of new tables (losing the old one)
\$DEFINE	Definition of additional tables
\$DEFOUT	Current list of table definitions

Main commands for treatment of data

\$MATRIX	Process TABFILE files
\$DATA	Process SDATA files and non-arithmetical data treatment
\$ARITH	Arithmetical operation with tables
SREGRESS	Multiple linear regression analysis with optional use of a forward stepwise algorithm
\$TABOUT	List or print tables
\$COMMENT	Print comments

## **\$CONTROL Set and list general control parameters**

\$CONTROL subcommands are used to define general parameters for the EV-session.

Syntax : \$CONTROL  
 e.g. \$CONTROL

### **\$CONTROL subcommands :**

INCLUDE	Define regions to be processed
FROMYEAR	Define first year to be processed
TOYEAR	Define last year to be processed
BASEYEAR	Define base year
TYPIN	Define type codes of tables for data input
TYPOUT	Define type codes of tables for data output
CALL	Call external command files
STATUS	List the status of definitions that may be done by \$CONTROL subcommands
ECHOON	Enable echoing of control statements in the protocol file
ECHOOFF	Disable echoing of control statements in the protocol file
ERRON	Enable writing protection for the TABFILE file
ERROFF	Disable writing protection for the TABFILE file
HELP	On line help for \$CONTROL subcommands

### **INCLUDE: Definition of region selection (\$CONTROL subcommand)**

Region selection (e.g. Member States) by regional codes.

Syntax: INCLUDE *state1 state2 .... state18*  
 e.g. INCLUDE D F I

The following regional codes can be used at the SPEL/EC level:

D	DEUTSCHLAND	L	LUXEMBOURG
F	FRANCE	UK	UNITED KINGDOM
I	ITALIA	IRL	IRELAND
E	ESPAÑA	DK	DANMARK
P	PORTUGAL	GR	ELLAS
NL	NEDERLAND	E09	EUROP.COMM.(EUR 09)
B	BELGIQUE/BELGIE	E10	EUROP.COMM.(EUR 10)
BL	BEBL/BLEU	E11	EUROP.COMM.(EUR 11)
		E12	EUROP.COMM.(EUR 12)

After changing the regional selection, all data in the program are lost, but the current table definitions are kept.

### **FROMYEAR: Definition of first year to be processed (\$CONTROL subcommand)**

First calendar year of calculation period.

syntax: FROMYEAR *year*  
 e.g. FROMYEAR 1967

**TOYEAR: Definition of last year to be processed (\$CONTROL subcommand)**

Last calendar year of calculation period.

syntax:     TOYEAR *year*  
e.g.        TOYEAR 1986

**BASEYEAR: Definition of base year (\$CONTROL subcommand)**

Base year, e.g. for calculating an index (see \$ARITH).

Syntax:     BASEYEAR *year*  
e.g.        BASEYEAR 1980

The year must be inside the calculation period FROMYEAR to TOYEAR.

**TYPIN: Definition of type code of table for data input (\$CONTROL subcommand)**

Type code for the SPEL tables from which the data are to be selected. This table type will be used as long as no other type definitions are given (see also \$MATRIX and \$DATA subcommands).

Syntax:     TYPIN *type*  
e.g.        TYPIN CONB

**TYPOUT: Definition of type code of table for data output (\$CONTROL subcommand)**

Type code for the SPEL tables to be stored. This table type will be used, as long as no other type definitions are given (see also \$MATRIX and \$DATA subcommands).

syntax:     TYPOUT *type*  
e.g.        TYPOUT TESB

**CALL: Execution of a STEER file (\$CONTROL subcommand)**

Often repeated command sequences can be stored as macros in additional STEER control files. The CALL command can be used to call up the files and execute the commands.

syntax:     CALL *controlfilename*  
e.g.        CALL MACRO1 STEER A

**STATUS: List the status of definitions (\$CONTROL subcommand)**

Listing of the current program status (e.g. general control parameters).

syntax:     STATUS  
e.g.        STATUS

**ECHOON: Enable echoing of control statements in the protocol file (\$CONTROL subcommand)**

All commands entered are logged. Used mainly when working with a control data file to check the command input.

syntax: ECHOON  
e.g. ECHOON

**ECHOOF: Disable echoing of control statements in the protocol file (\$CONTROL subcommand)**

Discontinues logging of commands.

syntax: ECHOOFF  
e.g. ECHOOFF

**ERRON: Enable writing protection for the TABFILE work file (\$CONTROL subcommand)**

Setting of an internal "error"-flag; storing of data in the work file is no longer allowed.

syntax: ERRON  
e.g. ERRON

Remark : Fatal errors occurring will automatically set the error flag.

**ERROFF: Disable writing protection for the TABFILE work file (\$CONTROL subcommand)**

See "ERRON". ERROFF clears the internal "error" flag.

syntax: ERROFF  
e.g. ERROFF

**HELP : On-line help for \$CONTROL subcommands (\$CONTROL subcommand)**

List \$CONTROL subcommands, for cursor selection.

Syntax: HELP  
e.g. HELP

**\$HELP : On-line help for main commands**

This command displays all main commands for cursor selections (see "List of main commands")

Syntax : \$HELP  
e.g. \$HELP

### **\$SKIPON : Enable skipping of the following command statements**

By inserting the main command \$SKIPON in a control file, the following command statements are skipped. Only the main commands \$STOP and \$SKIPOFF are interpreted. The command is normally used to de-activate a command sequence in a control file without removing the de-activated commands.

Syntax :    \$SKIPON  
e.g.        \$SKIPON

### **\$SKIPOFF : Disable skipping of the following command statements**

See \$SKIPON. The command \$SKIPOFF disables the skipping function.

Syntax :    \$SKIPOFF  
e.g.        \$SKIPOFF

### **\$STOP: Exit from EV-session**

Enter \$STOP to finish your EV-session. You will come back to the level from where you started the EV-session.

Syntax:     \$STOP  
e.g.        \$STOP

### **\$NEW: Definition of new tables losing the old ones**

To define EV-tables, the variable definition modus must first be called up by entering \$NEW or \$DEFINE. Then the variables are specified by giving their codes line by line. Later operations then refer to the variable (table) number rather than to the variable name. The variable number is equal to the position of the table in the sequence of table definitions.

A \$NEW command deletes all tables defined so far and starts a new definition sequence.

syntax:     \$NEW  
e.g.        \$NEW

Until a new \$-command occurs, the following lines are subcommands or table definitions. If a line is not a legal subcommand, it is interpreted as a table definition.

## **\$DEFINE: Definition of additional tables keeping the old ones**

A \$DEFINE command starts a definition sequence to append new tables to the current table definitions.

syntax:     \$DEFINE  
e.g.        \$DEFINE

Until a new \$-command occurs, the following lines are subcommands or table definitions. If a line is not a legal subcommand, it is interpreted as a table definition.

### **Table definition lines of \$NEW or \$DEFINE**

Syntax :     xxxxyyyy 0 *descriptiontext*  
e.g.        WHEAWHEA 0 'Yield coefficient wheat kg/ha'

Up to 8 characters are allowed for defining a table code xxxxyyyy. Names of legal subcommands (see below) are not allowed. Using SPEL codes, the first four characters xxxx specify a column of SPEL tables, the remaining four characters yyyy a row of SPEL tables. Non-SPEL codes are allowed too, but data for those codes cannot be read from SPEL TABFILE files.

Currently the "0" must be specified as a "dummy" for further extensions. A description text could be assigned otherwise the system gets the systems default description text (if available).

Each table definition line excepts an immediately following RENAME subcommand that increments the number of defined tables by one. After creating a new table, the current value of the table counter is implicitly assigned as an identifier to the table defined. This "table number" represents the position in the sequence of defined tables and is used in all EV operations to specify selected tables. Therefore the table code may not be unique.

### **Subcommands of \$NEW or \$DEFINE :**

TREND     Definition Trend  
RENAME    Rename a defined table  
HELP      On-line help for \$NEW or \$DEFINE subcommands

### **TREND: Definition of a trend table (\$NEW or \$DEFINE subcommand)**

This subcommand defines a trend table to be used for regression calculations. The table is filled with trend numbers which are automatically generated.

Syntax:     TREND *startnumber*  
e.g.        TREND 65

The argument *startnumber* gives the number to be used for the first year in the tables. Trend numbers for the following years are generated using the increment 1. If *startnumber* is missing, 1 is assumed.



**RENAME: Rename an already defined table (\$NEW or \$DEFINE subcommand)**

A RENAME command allows the code and description text of a table to be changed. The specified table is replaced by the subsequently defined table.

Syntax:     RENAME *table*  
e.g.         RENAME 2

The new code and descriptive text for the specified table must be defined in a "table definition line" (see above) which must be followed immediately by the RENAME command. This definition line does not increment the table counter.

By entering RENAME without specifying a *table* you will rename the table which was defined last.

**HELP: On-line help for \$NEW or \$DEFINE subcommands \$NEW or \$DEFINE subcommand)**

List of subcommands for cursor selection.

Syntax :     HELP  
e.g.         HELP

**\$DEFOUT: List of current table definitions**

List the definitions of tables currently defined. \$DEFOUT allows the codes, sequence numbers and descriptive text to be obtained and checked.

Syntax:     \$DEFOUT  
e.g.         \$DEFOUT

Remark :    There are no subcommands related to \$DEFOUT.

**\$MATRIX: Reading and writing data from / into TABFILE files**

This command defines all input and output related to TABFILE files.

Syntax:     \$MATRIX  
e.g.         \$MATRIX

**\$MATRIX subcommands**

TABFILE	Definition of work file to be used
SELECT	Definition of a selection of tables to be read and written in
TYPIN	Definition of a type overlay sequence for SPEL tables to be read
GEN	Definition of model area subkey for SPEL tables to be read or written in
MATIN	Reading data
MATOUT	Writing data
MATOUTR	Writing data
CLOSE	Closing access to a work file
HELP	On-line help for subcommands of \$MATRIX

**TABFILE: Definition of work file to be used (\$MATRIX subcommand)**

The TABFILE subcommand opens a TABFILE work file either for input only or for both output and input.

Syntax : TABFILE *iomode filename*  
e.g. TABFILE INPUT SPEL-USR TABFILE A

Legal values for *iomode* are "INPUT" or "OUTPUT". If "INPUT" is specified, no subsequent writing is possible in the file. If "OUTPUT" is specified, reading from and writing in the file is allowed.

Only one TABFILE file can be open at one time. A subsequent TABFILE command will automatically close the currently opened file first and then open the currently specified file.

**SELECT: Definition of a selection of tables (\$MATRIX subcommand)**

The subcommand SELECT defines a selection of EV-tables for subsequent reading and writing by MATIN and MATOUT.

syntax : SELECT *table1 .... table18*  
e.g. SELECT 3 6 8

The parameters *table1 .... table18* are EV-table numbers (see "Definition of Tables").

All table selections are lost when the next main command is entered. The default selection depends on the related subcommand (see MATIN, MATOUT).

**GEN: Definition of the model area subkey (\$MATRIX subcommand)**

The subcommand GEN defines the model area subkey of SPEL tables to be read or written in.

syntax : GEN *area*  
e.g. GEN S

**TYPIN: Definition of a type overlay sequence for SPEL tables to be read (\$MATRIX subcommand)**

The subcommand defines an overlay sequence of type subkeys of SPEL tables to read later by the MATIN command.

syntax : TYPIN *type1 type2 .... type18*  
e.g. TYPIN ZPAB PRAB

The arguments *type1 type2 .... type18* specify the type subkeys in descending priority from left to right. If no arguments are given, the current overlay definition is deleted. Otherwise the definition is kept for the whole EV session until it is redefined by a new TYPIN subcommand of \$MATRIX.

**MATIN: Reading of data (\$MATRIX subcommand)**

MATIN reads data respecting the current selections done by the \$CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPIN, the EV-table definitions of \$NEW or \$DEFINE and also respecting the selections by the \$MATRIX-subcommands SELECT, TYPIN and GEN. If no SELECT-definitions exist, all defined tables are implicitly selected (see \$DEFINE). The reading is done from the file specified in the last TABFILE subcommand of \$MATRIX.

To read data by MATIN, the codes of all selected tables must also be legal codes of SPEL table elements. EV-table codes which are not SPEL table element codes are not affected by MATIN.

Syntax :    **MATIN** *tabletype* *region1* *region2* .... *region17*  
 e.g.        **MATIN** CONB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be read. If it is missing, the current setting of the TYPIN (see above) is used.

Parameters *region1* ... *region17* may define a selection of regions to be read. These parameters specify sequence numbers rather than regional codes. The above example will read SPEL tables of type code CONB for two regions defined as the 2nd and 5th arguments of the INCLUDE subcommand of \$CONTROL. If no *region..* parameter is given, data for all defined regions are read.

**MATOUT: Writing of data (\$MATRIX subcommand)**

MATOUT writes data respecting the current selections done by the \$CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPOUT, the EV-table definitions of \$NEW or \$DEFINE and respecting also the selections done by the \$MATRIX-subcommands SELECT and GEN. If no SELECT-definitions exist, only the "depending" table of the last regression estimation is implicitly selected. All elements of SPEL tables not currently selected are kept. The file specified in the last TABFILE command is written in. The file must be opened for output.

To write data by MATOUT, the code of the selected tables must also be legal codes of SPEL table elements.

Syntax :    **MATOUT** *tabletype* *region1* *region2* .... *region17*  
 e.g.        **MATOUT** CONB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be written. If it is missing, the current setting of TYPOUT (see \$CONTROL) is used.

For parameters *region...* see description of subcommand MATIN.

**MATOUTR: Writing of data (\$MATRIX subcommand)**

Same as the subcommand MATOUT, but all elements of SPEL tables not currently selected are reset to zero (0.0).

**CLOSE: Close the work file currently open(\$MATRIX subcommand)**

The CLOSE subcommand may be used to update the internal file directory of a work file but does not really close the file access. Further read or write operations may be done without reopening the file.

syntax :    CLOSE  
e.g.        CLOSE

If CLOSE is not explicitly called, the file directory of open work files is updated when opening a new work file by using the TABFILE subcommand of \$MATRIX or by ending the EV session.

**HELP: On line help for \$MATRIX subcommands (subcommand of \$MATRIX)**

List of subcommands for cursor selection.

Syntax :    HELP  
e.g.        HELP

**\$DATA: Processing of SDATA files and non-arithmetical data treatment**

\$DATA subcommands allow processing of sequential SDATA standard files and general non-arithmetical treatment of data.

Syntax :    \$DATA  
e.g.        \$DATA

**\$DATA subcommands**

STDFILE	Opening of SDATA standard files
TYPIN	Definition of a type overlay sequence for SPEL tables to be read
YEARS	Definition of the time period for output
SELECT	Definition of a selection of tables for reading, writing in and "lagging".
DATIN	Reading of data
DATOUT	Writing of data
LAG	Definition of time lags
TRANSP	Transposing of current data
DELETE	Deletion of tables
EDIT	Editing of data

**STDFILE: Definition of SDATA files to be used (\$DATA subcommand)**

The subcommand STDFILE opens a SDATA standard file either for input only or for output only.

Syntax :    STDFILE *iomode filename*  
e.g.        STDFILE INPUT MYFILE SDATA A

Legal values for *iomode* are "INPUT" or "OUTPUT". The INPUT mode is directly related to the DATIN command, the OUTPUT mode is related to the DATOUT command.

Two different files may be open at one time, one for input and one for output. Any subsequent STDFILE command first will implicitly close the currently opened file with the same *iomode*.

**TYPIN: Definition of a type overlay sequence for SPEL tables to be read (\$DATA subcommand)**

This subcommand defines an overlay sequence of type subkeys of SPEL tables to be read later by the DATIN command.

syntax : TYPIN *type1 type2 .... type18*  
e.g. TYPIN EEEB FFFB

The arguments *type1 type2 .... type18* specify the type subkeys in descending priority from left to right. If no arguments are given, the current overlay definition is deleted. Otherwise the definition is kept for the whole EV session until it is redefined by a new TYPIN subcommand of \$DATA.

**SELECT: Definition of a selection of tables (\$DATA subcommand)**

Data input and output (DATIN and DATOUT) and definitions of time lags (LAG) can be done selectively for specific tables.

syntax : SELECT *table1 table2 .... table18*  
e.g. SELECT 3 5

The parameters *table1 .... table18* are table numbers (see "Definition of Tables").

All table selections are lost by entering the next main command. The default selection depends on the related subcommand (see MATIN, MATOUT, LAG).

**GEN: Definition of the model area subkey (\$DATA subcommand)**

The subcommand GEN defines the model area subkey of SPEL tables to be read or written in.

syntax : GEN *area*  
e.g. GEN S

**DATIN: Reading of data (\$DATA subcommand)**

DATIN reads data respecting the current selections done by the \$CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPIN, the EV-table definitions of \$NEW or \$DEFINE and respecting also the selections by the \$DATA-subcommands SELECT, TYPIN and GEN. If no SELECT-definitions exist, all defined tables (see \$DEFINE) are implicitly selected. The reading is done from the file opened in the last STDFILE command with INPUT mode specified.

To read data by DATIN, the table codes used in the file to be read must match the table codes defined in the EV session.

Syntax : DATIN *tabletype region1 region2 ... region17*  
e.g. DATIN XXXB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be read. If it is missing, the current setting of the TYPIN (see above) is used.

Parameters *region1 ... region17* may define a selection of regions to be read. These parameters specify sequence numbers rather than regional codes. The above example will read SPEL tables of

type code CONB for two regions defined as the 2nd and 5th argument of the INCLUDE subcommand of \$CONTROL. If no *region..* parameter is given, data for all defined regions are read.

### DATOUT: Writing of data (subcommand of \$DATA)

DATOUT writes data respecting the current selections done by the \$CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPLOT, the EV-table definitions of \$NEW or \$DEFINE and respecting also the selections done by the \$DATA-subcommands SELECT and GEN. If no SELECT-definitions exist, only the "depending" table of the last regression estimation is implicitly selected. Writing is done into the file which was opened for OUTPUT mode.

To write data by MATOUT, the codes (see \$DEFINE or \$NEW above) of the selected EV-tables must also be legal SPEL table element codes.

Syntax :     MATOUT *tabletype region1 region2 .... region17*  
e.g.         MATOUT YYYYB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be written. If it is missing, the current setting of TYPLOT (see \$CONTROL) is used.

For parameters *region...* see description of DATIN subcommand.

### YEARS: Definition of time period for output (\$DATA subcommand)

Using the subcommand YEARS the time series written by the next DATOUT command may be shorter than the calculation period defined by \$CONTROL-subcommands FROMYEAR and TOYEAR.

Syntax :     YEARS *firstyear lastyear*  
e.g.         YEARS 70 75

The parameters *firstyear* and *lastyear* represent the first and last year of data which will be output by subsequent DATOUT commands. The output period is kept for the whole EV session until it is redefined by a new YEARS subcommand of \$DATA. If the new defined output period exceeds the calculation period it is clipped.

### LAG: Definition of time lags (\$DATA subcommand)

The LAG subcommand offers "permanent lagging" of time series in the EV core memory. Subsequently all input, output and arithmetical commands will process the "lagged" data. For temporary "lagging" of dependent and independent regression analysis variables see the main command \$REGRESS.

The LAG subcommand must always be used in combination with the SELECT subcommand.

syntax :     SELECT *table1 table2 .... table18*  
              LAG     *timelag1 timelag2 .... timelag18*

e.g.         SELECT 2 3  
              LAG     1 2

The LAG arguments correspond to the SELECT arguments. In the example, *timelag2* defines the lag shift for the table specified by *table2*.

---

All elements  $t$  of time series  $X$  are shifted as:  $X_t = X_{(t+timelag)}$ , where *timelag* must be a positive or negative integer. Depending on this number, at one end of the time series data figures are lost and at the other end zero figures are inserted.

**TRANSPO: Transposing the current data (\$DATA subcommand)**

In the normal status the data in the EV-tables are organized as

Table lines = years  
Table columns = regions  
Table headers = variables

Transposing data organized normally will change the organization to

Table lines = years  
Table columns = variables  
Table headers = regions

Transposing again will re-establish the normal status.

Syntax :    TRANSPO  
e.g.        TRANSPO

**DELETE: Deletion of tables (\$DATA subcommand)**

The subcommand DELETE deletes EV-tables.

Syntax :    DELETE *table1 table2 .... table18*  
e.g.        DELETE 2 3 4

The parameters *table1 .... table18* are table numbers (see "Definition of Tables").

Remark :    When deleting tables which are not at the end of the definition sequence, the remaining table numbers will change. Assuming there were six originally defined tables, the deletion example will rename the remaining original table numbers 1, 5 and 6 as new table numbers 1, 2 and 3.

**EDIT: Editing of Data (\$DATA subcommand)**

The subcommand EDIT allows data in the EV-tables to be changed or new data to be inserted into the tables. Editing is always related to a column of an EV-table. To edit data in a table column, the subcommand EDIT first specifies the table number and the column number.

Syntax :    EDIT table column  
e.g.        EDIT 2 3

Immediately following the EDIT command, the data are entered in one or more "data lines".

Syntax:     *year value*  
e.g.        70 135.5

Editing of a table column must be terminated by entering a blank line. To edit another table or column, the complete edit sequence must be repeated.

**\$ARITH: Arithmetical calculations**

Various arithmetical and transformation operations for processing data are offered by the subcommands of \$ARITH. All operations affect complete EV-tables which are specified as command operands.

Syntax :    \$ARITH  
e.g.        \$ARITH

**Subcommands of \$ARITH**

ADD	Addition of table elements to table elements
SUB	Subtraction of table elements from table elements
MUL	Multiplication of table elements by table elements
DIV	Division of table elements by table elements
ADDC	Addition of constants to table elements
SUBC	Subtraction of constants from table elements
MULC	Multiplication of table elements by constants
DIVC	Division of table elements by constants
ROOTC	Root of table elements
POTC	Raising to a power of table elements
LN	Natural logarithm of table elements
LOG	Logarithm (base 10) of table elements
REZ	Reciprocal of table elements
EXP	Exponential operation for table elements
POW	Power operation for table elements
INDEXGR	Conversion of growth rate table elements to indices
GROWTH	Growth rates in percent for table elements
BASIS	Division of table elements by base year elements of the same tables
MULB	Multiplication of table elements by base year elements of other tables
DIVB	Division of table elements by base year elements of other tables
ADDB	Addition of base year elements of other tables to table elements
SUBB	Subtraction of base year elements of other tables from table elements
MAVC	Replacement of table elements by "Moving averages"



RESET	Resetting of table elements
CLEAN	Resetting of table elements depending on reference tables
OVERLAY	Replacement of zero table elements with elements of other tables
HELP	On-line help for subcommands of \$ARITH

Results of arithmetical operations change only the first operand or all operands depending on the command.

**ADD: Addition of table elements to table elements (\$ARITH subcommand)**

Syntax: ADD *table1 table2 .... table9*  
e.g. ADD 5 2

Operation:  $table1_{t,c} = table1_{t,c} + table2_{t,c} + \dots + table9_{t,c}$

**SUB: Subtraction of table elements from table elements (\$ARITH subcommand)**

Syntax: SUB *table1 table2 .... table9*  
e.g. SUB 5 2

Operation:  $table1_{t,c} = table1_{t,c} - table2_{t,c} - \dots - table9_{t,c}$

**MUL: Multiplication of table elements by table elements (\$ARITH subcommand)**

Syntax: MUL *table1 table2 .... table9*  
e.g. MUL 5 2

Operation:  $table1_{t,c} = table1_{t,c} * table2_{t,c} * \dots * table9_{t,c}$

**DIV: Division of table elements by table elements (\$ARITH subcommand)**

Syntax: DIV *table1 table2 .... table9*  
e.g. DIV 5 2

Operation:  $table1_{t,c} = \frac{table1_{t,c}}{table2_{t,c} * table3_{t,c} * \dots * table9_{t,c}}$

**ADDC: Addition of constants to table elements (\$ARITH subcommand)**

Syntax: ADDC *table1 constant1 .... constant8*  
e.g. ADDC 5 2.55

Operation:  $table1_{t,c} = table1_{t,c} + constant1 + constant2 + \dots + constant8$

**SUBC: Subtraction of constants from table elements (\$ARITH subcommand)**

Syntax: SUBC *table1 constant1 .... constant8*  
e.g. SUBC 5 2.55

Operation:  $table1_{t,c} = table1_{t,c} - constant1 - constant2 - \dots - constant8$

**MULC: Multiplication of table elements by constants (\$ARITH subcommand)**

Syntax: MULC *table1 constant1 .... constant8*  
 e.g. MULC 5 2.55

Operation:  $table1_{t,c} = table1_{t,c} * constant1 * constant2 * .... * constant8$

**DIVC: Division of table elements by constants (\$ARITH subcommand)**

Syntax: DIVC *table1 constant1 .... constant8*  
 e.g. DIVC 5 2.55

Operation:  $table1_{t,c} = \frac{table1_{t,c}}{constant1 * constant2 * .... * constant8}$

**ROOTC: Root of table elements (\$ARITH subcommand)**

Syntax: ROOT *table1 constant1*  
 e.g. ROOT 5 4.

Operation:  $table1_{t,c} = \sqrt[constant1]{table1_{t,c}}$

**POTC: Raising to a power of table elements (\$ARITH subcommand)**

Syntax: POTC *table1 constant1*  
 e.g. POTC 5 4.

Operation:  $table1_{t,c} = table1_{t,c}^{constant1}$

**LN: Natural logarithm of table elements (\$ARITH subcommand)**

Syntax: LN *table1 table2 .... table9*  
 e.g. LN 1 3 6

Operation:  $table_{i,t,c} = \text{LOG}_e(table_{i,t,c})$

**LOG: Logarithm (base 10) of table elements (\$ARITH subcommand)**

Syntax: LOG *table1 table2 .... table9*  
 e.g. LOG 1 3 6

Operation:  $table_{i,t,c} = \text{LOG}_{10}(table_{i,t,c})$

**REZ: Reciprocal of table elements (\$ARITH subcommand)**

Syntax: REZ *table1 table2 .... table9*  
 e.g. REZ 1 3 6

Operation:  $table_{i,t,c} = \frac{1}{table_{i,t,c}}$

**EXP: Exponential operation for table elements (\$ARITH subcommand)**

Syntax: EXP *table1 table2 .... table9*  
 e.g. EXP 1 3

Operation:  $table_{i,t,c} = e^{table_{i,t,c}}$

**POW: Power operation for table elements (subcommand of \$ARITH)**

Syntax: POW *table1 table2 .... table9*  
 e.g. POW 1 3 6

Operation:  $table_{i,t,c} = 10.^{table_{i,t,c}}$

**INDEXGR: Conversion of growth rate table elements to indices (\$ARITH subcommand)**

Syntax: INDEXGR *table1 table2 .... table9*  
 e.g. INDEXGR 1 3 6

Operation:  $table_{i,1,c} = 100.$   
 $table_{i,t,c} = \left(1 + \frac{table_{i,t-1,c}}{100.}\right)$  for  $t > 1$

**GROWTH: Growth rates in percent for table elements (\$ARITH subcommand)**

Syntax: GROWTH *table1 table2 .... table9*  
 e.g. GROWTH 1 3 6

Operation:  $table_{i,1,c} = 0.$   
 $table_{i,t,c} = \left(\frac{table_{i,t,c}}{table_{i,t-1,c}} - 1.\right) * 100.$  for  $t > 1$

**BASIS: Division of table elements by base year elements of the same table (\$ARITH subcommand)**

Syntax: BASIS *table1 table2 .... table9*  
 e.g. BASIS 1 3 6

Operation:  $table_{i,t,c} = \frac{table_{i,t,c}}{table_{i,tb,c}}$

where *tb* points to the base year defined by the BASEYEAR subcommand of \$CONTROL

Remark : To get an index-time series based on 100, use the subcommand MULC to multiply the results by 100.  
 e.g. MULC 1 100.  
 MULC 3 100.  
 MULC 6 100.

**MULB: Multiplication of table elements by base year elements of other tables (\$ARITH subcommand)**

Syntax: MULB *table1 table2 .... table9*  
 e.g. MULB 1 3 6

Operation:  $table1_{t,c} = table1_{t,c} * table2_{tb,c} * table3_{tb,c} * \dots * table9_{tb,c}$

where *tb* points to the base year defined by the BASEYEAR subcommand of \$CONTROL

**DIVB: Division of table elements by base year elements of other tables (\$ARITH subcommand)**

Syntax: DIVB *table1 table2 .... table9*  
 e.g. DIVB 1 3 6

Operation:  $table1_{t,c} = \frac{table1_{t,c}}{table2_{tb,c} * table3_{tb,c} * \dots * table9_{tb,c}}$

where *tb* points to the base year defined by the BASEYEAR subcommand of \$CONTROL

**MAVC: Replacement of table elements with "Moving averages" (\$ARITH subcommand)**

Syntax: MAVC *table n*  
 e.g. MAVC 3 3

Operation:  $table_{t,c} = \frac{\sum_{p=1}^n table_{t+p-h-1,c}}{n}$

where *h* is the integer part of  $\frac{n}{2}$

Remark : The formula implies time subscripts of less than 1 at the beginning of time series as well as time subscripts greater than the number of years defined at the end of time series. The related data elements are generated by duplicating the next available element.

**RESET: Resetting of table elements (\$ARITH subcommand)**

Syntax: RESET *table1 table2 .... table9*  
 e.g. RESET 1 3 6

Operation:  $table_{i,t,c} = 0.$

**CLEAN: Resetting of table elements depending on reference tables (\$ARITH subcommand)**

Syntax: CLEAN *table1 table2 .... table9*  
 e.g. CLEAN 1 3 6

Operation: Changes  $table1_{t,c}$  to zero if the corresponding element in at least one of  $table2 \dots table9$  is also zero.

**OVERLAY : Replacement of zero table elements with elements of other tables (\$ARITH subcommand)**

Syntax: OVERLAY  $table1 table2 \dots table9$   
e.g. OVERLAY 1 3 6

Operation: Replace zero elements of  $table1_{t,c}$  with corresponding elements of  $table2 \dots table9$  until  $table1_{t,c}$  is no longer zero.

**HELP: On-line help for \$ARITH subcommands (\$ARITH subcommand)**

List of subcommands for cursor selection.

Syntax : HELP  
e.g. HELP

**\$REGRESS : Multiple linear regression analysis with optional use of a forward stepwise algorithm**

\$REGRESS subcommands offer linear regression analysis with one or more independent variables. The subcommands affect complete EV-tables which are specified as command operands. All currently defined EV-tables may be defined as the regression variables. The variables to be entered into the regression are specified as table numbers (see \$DEFINE).

The variables can be transformed to fit several regression curves. Separate estimates are made for each table column. Optional step-by-step entering of independent variables may be selected.

EV calculates the usual statistical fit parameters (e.g. Durbin Watson coefficient, coefficient of determination, etc.). The "correlation coefficients" of regression equations and common "elasticities" are also available for each estimate.

Syntax : \$REGRESS  
e.g. \$REGRESS

**Subcommands:**

SELECT	Definition of variables to be entered in the regression
ESTIMATE	Regression estimation with optional variable transformations
PCTNEW	Definition of variance threshold for step-by-step entering of variables
LISTON	Enable listing of correlation coefficients and elasticities
LISTOFF	Disable listing of correlation coefficients and elasticities
PRINTON	Enable printing of correlation coefficients and elasticities
PRINTOFF	Disable printing of correlation coefficients and elasticities

**SELECT: Definition of variables to be entered in the regression (\$REGRESS subcommand)**

Syntax: SELECT  $table1 table2 \dots table18$   
e.g. SELECT 1 3 6

The first operand *table1* is defined as the dependent variable and the remaining operands *table2* .... *table18* are defined as independent variables. The variable selection is valid only for the next ESTIMATE subcommand. ESTIMATE will re-establish the default selection which is equal to the general variable definition (see \$NEW and \$DEFINE).

**ESTIMATE: Regression estimation with optional variable transformations (\$REGRESS subcommand)**

Syntax: ESTIMATE *trans1 trans2 .... trans18*  
 e.g. ESTIMATE LIN LOG LOG

The operands *trans1 trans2 .... trans18* define transformation codes for the corresponding selected regression variables (e.g. *trans1* is the transformation to be done for *table1*). New estimated values for the dependent variable are put into an EV-table which is identified by table number 999. Correlation coefficients and elasticities are listed and/or printed only if listing and/or printing is enabled. The usual statistical fit parameters are prepared to permit access by the \$TABOUT-subcommands LISTR and PRINTR.

The following transformations are possible for both dependent and independent variables:

LIN	Linear (no transformation)	$x' = x$
LN	Natural logarithm	$x' = \text{LOG}_e(x)$
LOG	Logarithm base 10	$x' = \text{LOG}_{10}(x)$
SQR	Square root	$x' = \sqrt{x}$
REZ	Reciprocal	$x' = \frac{1}{x}$
EXP	Exponential	$x' = e^x$
POW	Power	$x' = 10^x$
QUA	Square	$x' = x^2$

where x is an element of tables to be transformed.

**PCTNEW: Definition of variance threshold for stepwise entering of variables (\$REGRESS subcommand)**

Syntax: PCTNEW *pct*  
 e.g. PCTNEW 100.

"Stepwise" regression is allowed only when *pct* is not equal to zero. On default, the "stepwise" option is disabled (*pct* = zero). The definition is kept for the whole EV-session until it is redefined by a new PCTNEW command.

The parameter *pct* defines a threshold for the ratio (in ‰) of explanation rate of the variance of each independent variable related to the total variance. A dependent variable is entered in the regression when the explanation rate is greater or equal *pct*.

**LISTON: Enable listing of correlation coefficients and elasticities (\$REGRESS subcommand)**

Syntax : LISTON  
 e.g. LISTON

Listings of coefficients are displayed by the following ESTIMATE commands.

**LISTOFF: Disable listing of correlation coefficients and elasticities (\$REGRESS subcommand)**

Syntax : LISTOFF  
 e.g. LISTOFF

See LISTON

**PRINTON: Enable printing of correlation coefficients and elasticities (\$REGRESS subcommand)**

Syntax : PRINTON  
 e.g. PRINTON

Listings of coefficients are printed by the following ESTIMATE commands.

**PRINTOFF: Disable printing of correlation coefficients and elasticities (\$REGRESS subcommand)**

Syntax : PRINTOFF  
 e.g. PRINTOFF

See PRINTON.

**\$TABOUT: Listing or printing of tables**

The figures in normal EV-tables defined by \$NEW or \$DEFINE and the figures in special tables constructed by \$REGRESS are displayed or printed.

Syntax : \$TABOUT  
 e.g. \$TABOUT

Like other SPEL programs, EV prints in a "printer file" (e.g. named EV LISTING) rather than sending directly to the printer. The "printer file" may be printed after the end of the EV-session.

**Subcommands:**

LISTD	List normal tables on the screen
PRINTD	Print normal tables
LISTR	List regression result tables on the screen
PRINTR	Print regression result tables
COMMENT	Print comments
PRINTON	Enable subsequent printing of tables
PRINTOFF	Disable subsequent printing of tables
PAGE	Define printer page height
RECORD	Define printer page width
CNUMON	Enable column numbering
CNUMOFF	Disable column numbering
ROWS	Select table rows to be printed

LEGON            Enable printing of table legends  
LEGOFF          Disable printing of table legends

**LISTD: List normal tables on the screen (\$TABOUT subcommand)**

Syntax:        LISTD *table1 table2 .... table18*  
e.g.            LISTD 1 3 6

The arguments *table1 table2 .... table18* specify the numbers of tables to be listed. If no parameters are specified, all tables are listed.

The meaning of headers and columns depends on the current transposition mode (see the TRANSP0-subcommand of \$DATA). Rows are always reserved for years.

The following additional rows of simple statistical figures are generated:

MEAN            Arithmetical mean of all years  
GROWTH%        Arithmetical mean of annual growth rates in percent  
G.GR.%          Geometrical mean of annual growth rates in percent  
C.OF VAR        Coefficient of variation in percent

Zero figures are excluded from the computation of the statistical figures.

**PRINTD: Print normal tables (\$TABOUT subcommand)**

Syntax:        PRINTD *table1 table2 .... table18*  
e.g.            PRINTD 1 3 6

(See LISTD)

**LISTR: List regression result tables on the screen (\$TABOUT subcommand)**

Syntax:        LISTR *rtable1 rtable2 .... rtable18*  
e.g.            LISTR 1 2 5

The LISTR command is only valid if regression results have been prepared (see the ESTIMATE subcommand of \$REGRESS). The arguments *table1 table2 .... table18* specify numbers which identify regression result tables to be listed. If no parameters are specified, all regression result tables are listed.

The following regression tables may be selected:

Table 1:        Estimated figures for the dependent variable  
Table 2:        Partial Regression Coefficients  
Table 3:        Standard error of regression coefficients  
Table 4:        T-values  
Table 5:        Error probabilities  
Table 6:        Other statistical figures



The meaning of table headers and columns depends on the current transposition mode (see the TRANSPO subcommand of \$DATA).

The layout of regression table 1 is as for normal tables (see LISTD). In regression tables 2 to 5 the rows are the variables which are entered into the regression.

The rows of regression table 6 have a special meaning:

CONSTANT	Intercept constant of the regression
DEGRFREE	Degrees of freedom
OBSERVAT	Number of observations
%.SUM.SQ	Percentage of sum of squares of deviations
M.CORREL	Multiple correlation coefficient
M.CORR.A	Multiple correlation coefficient adjusted
STD.ERR.	Standard estimation error
ST.ERR.A	Standard estimation error, adjusted
F-RATIO	F-ratio
DURBIN	Durbin-Watson statistic
THEIL	Theil's coefficient

#### **PRINTR: Print regression result tables (\$TABOUT subcommand)**

Syntax: PRINTR *rtable1 rtable2 ... rtable6*  
 e.g. PRINTR 1 2 5

(See LISTR.)

#### **COMMENT: Print comments (\$TABOUT subcommand)**

Syntax: COMMENT  
*text*  
 e.g. COMMENT  
 This is the text printed.

The subcommand allows one line of comment to be printed. The comment *text* is specified by a separate line which must immediately follow the COMMENT command line. Carriage control is supported by special characters entered at the beginning of the comment line (see \$COMMENT).

#### **PRINTON: Enable subsequent printing of tables (\$TABOUT subcommand)**

Syntax: PRINTON  
 e.g. PRINTON

The subcommands may be used to enable printing after it has been disabled by a previous PRINTOFF command.

#### **PRINTOFF: Disable subsequent printing of tables (\$TABOUT subcommand)**

Syntax : PRINTOFF  
 e.g. PRINTOFF

Subsequent subcommands PRINTD or PRINTR are ignored until printing is enabled again by PRINTON.

**PAGE: Definition of printer page height (\$TABOUT subcommand)**

Syntax : PAGE *lines*  
e.g. PAGE 67

The default page height is 57 lines.

**RECORD: Definition of printer page width (\$TABOUT subcommand)**

Syntax: RECORD *characters*  
e.g. RECORD 80

The default page width is 132 characters. Legal values are 72 to 132.

**CNUMON: Enable column numbering (\$TABOUT subcommand)**

Syntax: CNUMON  
e.g. CNUMON

The command enables printing of column sequence numbers in tables printed by subsequent PRINTD and PRINTR commands. On default only column codes are printed as column headers.

**CNUMOFF: Disable column numbering (\$TABOUT subcommand)**

Syntax : CNUMOFF  
e.g. CNUMOFF

(see CNUMON)

**ROWS: Select table rows to be printed (\$TABOUT subcommand)**

Syntax: ROWS *line1 line2 .... line18*  
e.g. ROWS 1 3 5

This subcommand defines a selection of table lines to be respected in subsequent subcommands LISTD and PRINTD. The arguments are relative line numbers related to range defined by the \$CONTROL-subcommands FROMYEAR and TOYEAR. If FROMYEAR is set to 1981, the above example will select the lines of years 1981, 1983 and 1985. A table line selection is kept for the whole EV session until it is redefined by a new ROWS command. On default all table lines are selected.

**LEGON: Enable printing of table legends (\$TABOUT subcommand)**

Syntax : LEGON  
e.g. LEGON

If legend printing is enabled, a column description is printed in addition to the column codes in the header of the tables. On default the legend printing is disabled.

**LEGOFF: Disable printing of table legends (\$TABOUT subcommand)**

Syntax: LEGOFF  
e.g. LEGOFF

(See LEGON)

### **\$COMMENT: Printing of comments**

This command allows comments to be placed in the printer output. After this command is entered the following lines in the command stream are interpreted as comments to be printed until a new \$...-command occurs.

If the first character of a comment line is not a blank character, it has been given a special treatment: "0" will insert one blank line before printing the comment. "1" will start a new output page. Each character other than a blank character will be repeated to fill a whole output line. If this occurs, the comment itself will be printed in the next output line.

### 6.2.1.5. A typical EV session

The structure of the control language reflects the work steps within EV applications. Reading control files of existing applications will help the user to learn the control language.

#### Example of a typical STEER file

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
$CONTROL
INCLUDE F I
FROMYEAR 1973
TOYEAR 1988
TYPOUT ESTB
BASEYEAR 1980
$TABOUT
RECORD 80
LEGON
$COMMENT

      --- TABLE WINE ---

$COMMENT

      ESTIMATION BY REGRESSION, PINDTWIN = F(PINDWINT)

$NEW
PINDTWIN 1
PINDWINT 2
$MATRIX
TABFILE OUTPUT SPEL-USR TABFILE A
GEN D
TYPIN ZPAB EXPB
MATIN
SELECT 2
TYPIN ZPAC EXPC ESTC
MATIN
$DATA
TRANSPO
$TABOUT
PRINTD
$DATA
TRANSPO
$REGRESS
SELECT 1 2
ESTIMATE LIN
$TABOUT
PRINTR
$MATRIX
SELECT
MATOUT ESTB 1
$STOP
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----

```

The control file results in the following printer output:

--- TABLE WINE ---

ESTIMATION BY REGRESSION, PINDTWIN = F(PINDWINT)

-----  
TABLE 1: PINDTWIN PRICE INDEX (1980=100);  
-----

	F	I
1 ! 1973 !	0.00	0.00
2 ! 1974 !	0.00	0.00
3 ! 1975 !	0.00	0.00
4 ! 1976 !	0.00	0.00
5 ! 1977 !	0.00	0.00
6 ! 1978 !	0.00	0.00
7 ! 1979 !	0.00	0.00
8 ! 1980 !	0.00	0.00
9 ! 1981 !	0.00	0.00
10 ! 1982 !	9280.00	10468.00
11 ! 1983 !	8614.00	22399.00
12 ! 1984 !	11213.00	15881.00
13 ! 1985 !	8891.00	10916.00
14 ! 1986 !	7649.00	22710.00
15 ! 1987 !	12062.00	20724.00
16 ! 1988 !	5800.00	14222.00
1 MEAN	9072.71	16760.00
2 GROWTH %	-0.98	20.26
3 G.GR.%	-7.53	5.24
4 C.OF VAR	21.46	28.87

-----  
DATE: 26.04.91 SOURCE: SPEL LUX., "BDS"-TYPE: ZPAB

TABLE 2: PINDWINT PRICE INDEX (1980=100);

	F	I
1 ! 1973 !	18642.00	3450.00
2 ! 1974 !	24887.00	9527.00
3 ! 1975 !	13575.00	3510.00
4 ! 1976 !	14970.00	1801.00
5 ! 1977 !	8557.00	1700.00
6 ! 1978 !	9276.00	2060.00
7 ! 1979 !	21485.00	9400.00
8 ! 1980 !	19748.00	14653.00
9 ! 1981 !	11871.00	10564.00
10 ! 1982 !	20069.00	10831.00
11 ! 1983 !	16856.00	23153.00
12 ! 1984 !	17800.00	17750.00
13 ! 1985 !	17626.00	12828.00
14 ! 1986 !	16777.00	24815.00
15 ! 1987 !	20652.00	23332.00
16 ! 1988 !	13736.00	15588.00
1 MEAN	16657.94	11560.12
2 GROWTH %	6.00	38.91
3 G.GR.%	-2.02	10.58
4 C.OF VAR	25.92	66.23

DATE: 26.04.91 SOURCE: SPEL LUX., "BDS"-TYPE: ZPAC

TABLE 3: PINDTWIN ESTIMATED VALUES  
DEP:LIN INDEP:LIN

	F	I
1 ! 1973 !	9836.71	2871.19
2 ! 1974 !	14622.91	8544.09
3 ! 1975 !	5953.34	2927.20
4 ! 1976 !	7022.47	1331.84
5 ! 1977 !	2107.52	1237.55
6 ! 1978 !	2658.57	1573.61
7 ! 1979 !	12015.60	8425.54
8 ! 1980 !	10684.36	13329.24
9 ! 1981 !	4647.39	9512.14
10 ! 1982 !	10930.37	9761.38
11 ! 1983 !	8467.91	21264.03
12 ! 1984 !	9191.40	16220.30
13 ! 1985 !	9058.04	11625.59
14 ! 1986 !	8407.37	22815.52
15 ! 1987 !	11377.18	21431.13
16 ! 1988 !	6076.73	14202.07
1 MEAN	8316.11	10442.02
2 GROWTH %	23.28	45.82
3 G.GR.%	-3.16	11.25
4 C.OF VAR	39.79	68.45

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

-----  
 TABLE 4: PINDTWIN PARTIAL REGRESSION COEFFICIENTS  
 ABH:LIN UNABH:LIN  
 -----

	F	I
1 PINDWINT	0.77	0.93

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

-----  
 TABLE 5: PINDTWIN STANDARD ERROR OF REGR.-COEFFICIENTS  
 ABH:LIN UNABH:LIN  
 -----

	F	I
1 PINDWINT	0.22	0.06

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

-----  
 TABLE 6: PINDTWIN T-VALUES  
 ABH:LIN UNABH:LIN  
 -----

	F	I
1 PINDWINT	3.41	16.61

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

-----  
 TABLE 7: PINDTWIN ERROR PROBABILITY, %  
 ABH:LIN UNABH:LIN  
 -----

	F	I
1 PINDWINT	1.43	0.00

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

-----  
 TABLE 8: PINDTWIN INTERCEPT CONSTANT AND TEST STATISTICS  
 DEP:LIN INDEP:LIN  
 -----

	F	I
1 CONSTANT	-4450.60	-349.41
2 DEGRFREE	6.00	6.00
3 OBSERVAT	7.00	7.00
4 % .SUM.SQ	69.93	98.22
5 M.CORREL	0.84	0.99
6 M.CORR.A	0.84	0.99
7 STD.ERR.	1263.37	763.91
8 ST.ERR.A	1263.37	763.91
9 F-RATIO	11.63	275.81
10 DURBIN	1.87	1.29
11 THEIL	0.27	0.08

DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

### 6.2.2. Data format conversion (DATCON)

Most exogenous data files are stored on tapes. The format conversion has to be done on a computer system which can access tape units (MVS batch system or VM batch).

The format standardization process, which includes a complete sorting step, is CPU time and I/O consuming. Therefore it is recommended that the format conversion should always be run in batch mode.

The format conversion program DATCON is steered by a parameter specifying the source format and a parameter for the type code to be created (FAO data only). The output format is always SPELIMP.

DATCON needs the following parameter block (PARAM file) :

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
DATCON
Input format = .....
Type code = .....
END
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----

```

Legal values for "Input format" are for example "FAO" and "CRONOS".

This parameter block may be generated by a program selection shell. For further informations see the documentation of the SPEL models (e.g. SPEL/EC Model).

### 6.2.3. Selection and conversion of sequential standard file formats (FILTER)

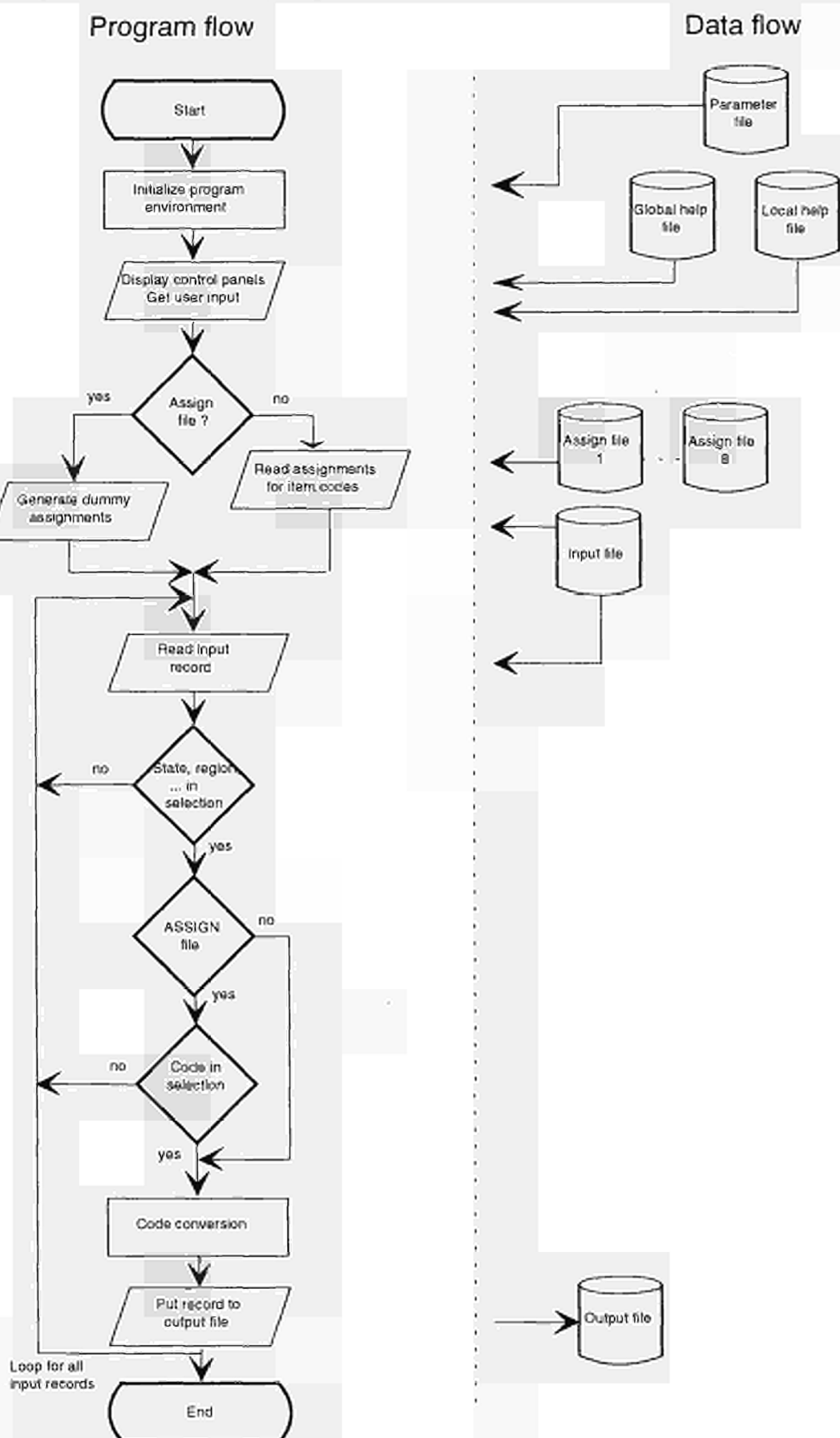
SDATA or SPELIMP data files can be selected and the format can be converted by the FILTER utility. The program works for one input file, the format of which is automatically recognized, and builds up a new SDATA or SPELIMP output file. Up to eight ASSIGN files can be used for selection by SPEL model codes and for code conversion.



The data selection is done first using a list of selection criteria, referring to the identifiers for region, sub-region, year, periodicity and type. A second selection step is added for all defined codes appearing in the assignment statements collected from all accessed ASSIGN files. If no ASSIGN files are accessed, no further selection is done.

When a format conversion is carried out, it is advisable in most cases to convert the code, because the SPELIMP file codes are source domain codes and those of SDATA file types are SPEL model codes. If no assignment file is accessed the input file codes will be copied into the output file without conversion.

Figure 6 : FILTER program flow chart



**Module startup screen**

```
Select and format data ----- SPEL ----- Module startup logo

.

DATA SELECTION AND FORMAT CONVERSION

Data selection and format conversion of SDATA or SPELIMP data files.
Build up a new output file of SDATA or SPELIMP format.

Current settings for general files:
Protocol file      : SPEL LOG C
Error file        : SPEL ERROR C

Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key an edit panel will be displayed which will permit changes to names of general files for run time protocol (LOG-file) and error messages (ERROR-file).

In the LOG-file the user will find a copy of all messages displayed on screen during the program flow. Further detailed information about work steps will be included too.

The ERROR-file contains the messages of any error conditions which may have occurred and which might have caused the program to stop prematurely.

It is always useful for the program user to check both general files after the program has ended.

**Parameters:**

None.

### Output modus screen

```
Select and format data ----- SPEL ----- Output modus
      Select an output modus by moving the cursor to any topic below
      then press ENTER

                               SPELIMP format
                               SDATA format

                               Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

### Parameters:

For output file format you can choose between SPELIMP format and SDATA format.

SPELIMP format is suitable to import source data into the SPEL system.  
It is optimized for fast item selection.

SDATA format is suitable for data to be revised by user using an editor program.

## Working file selection screen

```

Select and format data ----- SPEL ----- Work file selection
                                Please enter file names

Input file      => ZPA1 SPELIMP A
Output (SDATA) => OUTPUT SDATA C
Assign file 1   => SUPPLY-B ASSIGN F
Assign file 2   =>

                                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

### Parameters:

#### Input file

The input file has to be of one of the following formats:

- SPELIMP
- SDATA

The program FILTER works for one input file. FILTER identifies the format of the file automatically.

#### Output (SDATA) or Output (SPELIMP)

The output file will be of SDATA or SPELIMP format and will be newly formed, if you do not enter the name of an existing file.

#### Assign file 1...8

These files contain assignment statements for selecting source data and converting source data codes to SPEL table element codes. It must be of ASSIGN file format.

For FILTER the assignment file may be used for additional data selection. The data selection is done first using a list of selection criteria, referring to the identifiers for regions, sub-regions, current year, periodicity, base year, type and model area. A second selection step can be added for all defined codes appearing in the assignment statements.

### Selection criteria screen

```
Select and format data ----- SPEL ----- Selection
                Please make your selection for output file

Region          ( 3 ch.) => AAA : ZZZ
Sub-region      ( 2 ch.) => 00
Current year    ( 2 ch.) => 65 : 90
Periodicity    ( 2 ch.) => 00
Base year       ( 2 ch.) => NN
Type           ( 4 ch.) => AAAA : 9999
Model area      ( 1 ch.) => S

                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

#### Parameters:

##### Region

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

##### Sub-region

Sub-region selection. Specify one of the following:

- a sequence of sub-region codes to be selected such as "01 06 20"
- a range defined by lower and upper bounds such as "01 : 99"
- a combination of sequences and ranges such as "01 06 10 : 20"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

Sub-regions are currently not used.

The whole region has the sub-region code "00". Always specify "00".

### Current year

Current year selection. Specify one of the following:

- a sequence to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"
- use '\*' to select all years

All codes and the range operand ":" must be separated by blanks.

### Periodicity

Periodicity selection. Specify one of the following:

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2"
- a range defined by lower and upper bounds such as "Q1 : Q4"
- a combination of sequences and ranges such as "01 : 06 Q4"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

The whole year has the code "00". Other periodicity codes are currently not used.

### Base year

Base year selection. The selection is limited to one base year.

Select one base year such as "85" or "NN".

For input files of SDATA format:

The base year is a selection criteria on the input file.

For input files of SPELIMP format:

- The specified base year will be used to define the subkey "base year" in the table keys on the output file of SDATA format
- This subkey is protected for selection when output file is of SPELIMP format, too.

### Type

Type selection. The selection is limited to one type.

Select one type code such as "ZPA1" or "EXPB"

## Model area

Model area selection. The selection is limited to one model area.

Select one model area code such as "D".

For input files of SDATA format:

The model area is a selection criterion on the input file

For input files of SPELIMP format:

- The specified model area will be used to define the subkey "model area" in the table keys on the output file of SDATA format
- The subkey is protected for selection in case of output file is of SPELIMP format, too.

### **6.2.4. Maintenance of TABFILE files (DASERV)**

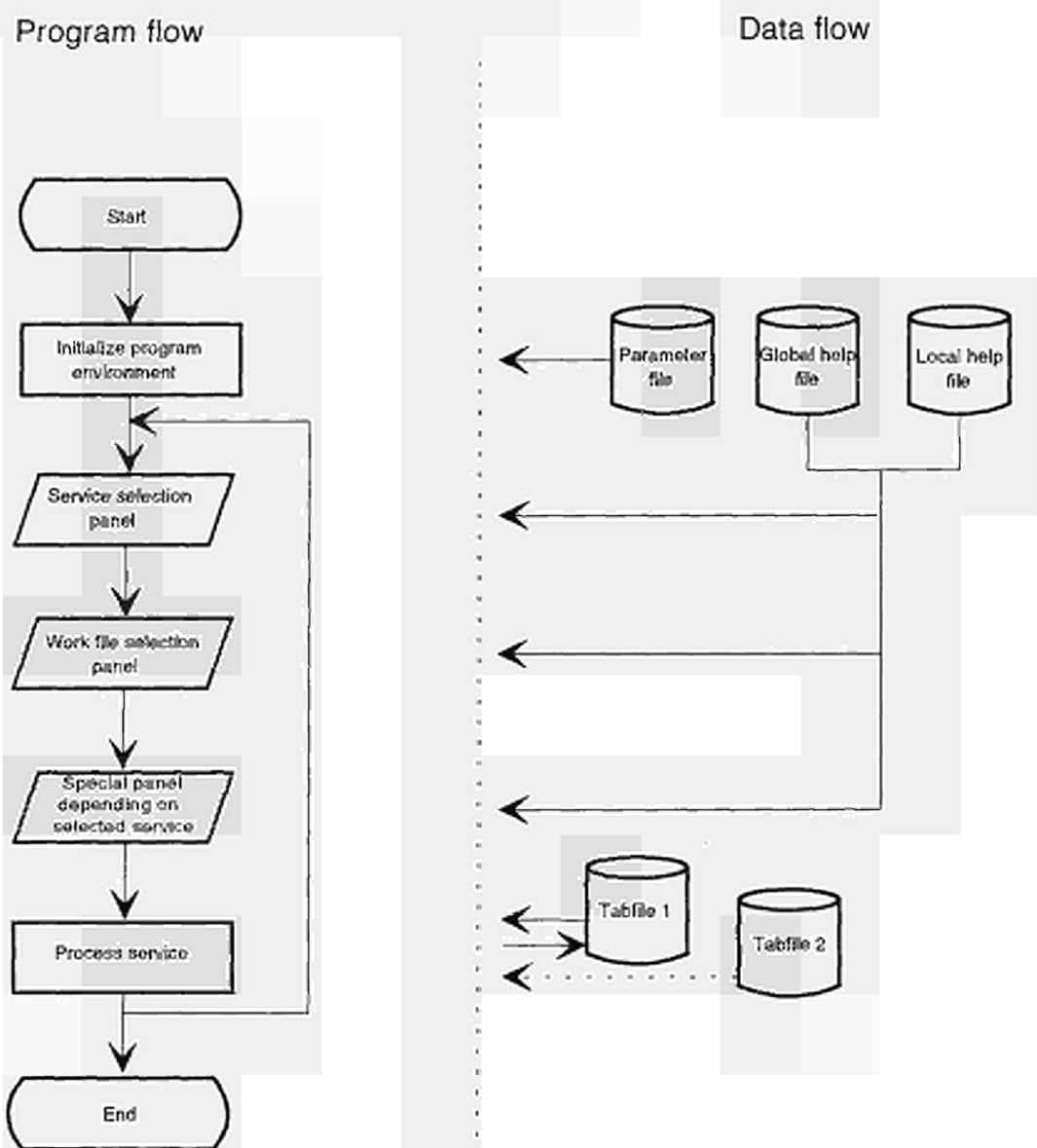
The most important file for the SPEL-Data is the TABFILE work file. This file format cannot be managed by editors or listed using view/browse commands.

The DASERV program offers, for TABFILE maintenance, many services:

- the organization and reorganization of the file
- the listing and checking of the file directory (table of contents of entries)
- the listing and printing of data.



Figure 7 : DASERV program flow chart



### Module startup screen

```
Table file services ----- SPEL ----- Module startup logo

          T A B L E   F I L E   S E R V I C E S

Maintenance services for work files of TABFILE format

Current settings for general files:
Protocol file      : DASERV PROTOCOL C
Error file        : DASERV ERROR C
Print file        : DASERV LISTING C
Scratch file      : DASERV SCRATCH C

Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key a special panel will be displayed which will permit changes to names of general files for run time protocol (LOG-file), error messages (ERROR-file), documentation file (LISTING-file) and temporary work file (SCRATCH file).

#### Parameters:

None.

## Service selection screen

```

Table file services ----- SPEL ----- Service selection
                Please select the general service

                Authorize access
                Compare entries
                Copy entries
                Copy key structure
                Create new file
                Define key structure
                Delete file
                Delete entries
                Edit entry data
                Free unused space
                List entry data
                List file directory
                List file status
                Print entry data
                Rename entries
                Test file consistency

                Select a service by moving the cursor to any of the above options

                Enter= ok  1= Help  3= Quit

```

### Selection items:

#### Authorize access

This service lets you define access restrictions for a specific file. You can define different groups of users for read and write access.

#### Compare entries

This service lets you compare the contents of entries which may be in the same or two different files.  
The program outputs a list of deviations found.

#### Copy entries

This service lets you copy a selection of entries from one file into another or into the same file.  
You may specify new subkey names for the destination entries.

#### Copy key structure

This service lets you copy the key structure information (name and length of subkeys) from one file into another.  
The length of the whole key in the source and destination files must be equal.

#### Create new file

This service lets you create a new TABFILE file for direct access to data and text entries.

#### Define key structure

This service lets you define a key structure which divides the table key into up to ten subkeys.

#### Delete entries

This service lets you delete entries in a file. It does not free the file space occupied by the deleted entries.

#### Delete file

This service lets you delete an existing TABFILE file.

#### Edit entry data

This service lets you edit the data area of existing TABFILE entries.

#### Free unused space

This service lets you free unused space, i.e. file directory space and file space occupied by 'deleted' entries.

#### List entry data

This service lets you list the data area of TABFILE entries.

#### List file directory

This service lets you list the entry names (keys) and entry text of a selection of entries in a file.

#### List file status

This service lets you list information about the file characteristics including capacity and usage information for the number of entries and number of records.

#### Print entry data

This service lets you print the data of TABFILE entries as "tables".

#### Rename entries

This service lets you rename entries in a TABFILE file. This means that the key will change without copying or moving any data.

#### Test file consistency

This service lets you test the file directory for inconsistencies.

## Authorize access :

This service lets you define access restrictions for a specific file. You can define different groups of users for read and write access.

You will be prompted for user ID's for read and write access for the specified file. You also have to define a password to control further access authorization modifications for the specified file.

## Work file selection screen

```
Table file services ----- SPEL ----- Authorize access
                                     Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be processed

The name of an existing file for which the access authorization is to be defined.

### Password screen

```
Table file services ----- SPEL ----- Authorize access
Please enter authorization password

Password =>

Enter= ok 1= Help 3= Quit
```

#### Parameters:

##### Password

The password allows changes to access authorization. The password is not displayed during typing.

**User names screen**

```

Table file services ----- SPEL ----- Authorize access
                Please define user access
Read only user  1 => USERID01 USERID02 USERID03 USERID04 USERID05 USERID06
                2 => USERID08 USERID09
                3 =>
                4 =>
                5 =>
                6 =>
                7 =>
                8 =>
                9 =>
Read/write user 1 => USERID11 USERID12
                2 =>
                3 =>
                4 =>
                5 =>
                6 =>
                7 =>
                8 =>
                9 =>

                Enter= ok  1= Help  3= Quit

```

**Parameters:****Read only user**

List of user ID's for which only read access is allowed.

The ID's have to be separated by blanks (e.g. USERID01 USERID02 USERID03... ).

You may use up to 9 lines from beginning.

**Read/write user**

List of user ID's for which read and write access is allowed.

The ID's have to be separated by blanks (e.g. USERID11 USERID12).

You may use up to 9 lines from beginning.

### New password screen

```
Table file services ----- SPEL ----- Authorize access
      Now you may modify the existing password

      New password or blank for no change =>
      If new password please repeat here  =>

      Enter= ok  1= Help  3= Quit
```

#### Parameters:

New password or blank for no change

The new password for further changes to access authorization. To keep the old password, do not enter anything in this panel

If new password please repeat here

If a new password was entered in the previous line, it has to be repeated here to prevent misspelling

#### Compare entries :

This service lets you compare the contents of entries which may be located in the same file or in two different files.

The program outputs a list of discrepancies found.

You will be prompted for the two file names and for an entry key selection.



**Work file selection screen**

```
Table file services ----- SPEL ----- Compare entries
                                     Please enter file names

Name of the reference file => SPEL-USR TABFILE A
Name of the compare file   => SPEL-USR TABFILE C

Enter= ok  1= Help  3= Quit
```

**Parameters:****Name of the reference file**

The name of an existing file which contains the reference entries.

**Name of the compare file**

The name of an existing file which contains the entries to be compared with the reference entries. This file may be the same as the reference file.

**Entry key selection screen**

```

Table file services ----- SPEL ----- Compare entries
                Please enter key selection

Region          ( 3 ch.) => D F
                compare with =>
Sub-region      ( 2 ch.) => 00
                compare with =>
Current year    ( 2 ch.) => 85 : 89
                compare with =>
Periodicity    ( 2 ch.) => 00
                compare with =>
Base year      ( 2 ch.) => NN
                compare with =>
Type           ( 4 ch.) => EXPC
                compare with =>
Model area     ( 1 ch.) => S
                compare with =>

                Enter= ok 1= Help 3= Quit

```

**Parameters:**

The parameter names "Region"....."Model area" are the subkey names defined in the service "Define key structure".

**Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical range, where "0" > "Z")
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

**Sub-region**

Sub-region selection. Specify one of the following:

- a sequence of sub-region codes to be selected such as "01 06 20"
- a range defined by lower and upper bounds such as "01 : 99"  
(alphanumerical range, where "0" > "Z")
- a combination of sequences and ranges such as "01 06 10 : 20"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

Sub-regions are currently not used.

---

The whole region has the sub-region code "00". Always specify "00".

#### Current year

Year selection. Specify one of the following : one of the following:

- a sequence to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

#### Periodicity

Periodicity selection. Specify one of the following :

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
(alphanumerical range, where "0" > "Z")
- a combination of sequences and ranges such as "01 : 06 Q4"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

Periodicities are currently not used

The whole year has the code "00". Always specify "00".

#### Base year

Base year area selection. Specify one of the following :

- a sequence of years to be selected such as "70 : 75"
- a range defined by lower and upper years such as "80 : 85"
- a combination of sequences and ranges such as "70 75 80 : 85"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

#### Type

Type selection. Specify one of the following :

- a sequence of type codes to be selected such as "ZPAB COSB"
- a range defined by lower and upper bounds such as "AAAA : 9999"  
(alphanumerical range, where "0" > "Z")
- a combination of sequences and ranges such as "AAAA : DDDD ZPAB"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

### Model area

Model area. Specify one of the following :

- a sequence of codes to be selected such as "S D"
- a range defined by lower and upper bounds such as "A : 9"  
(alphanumeric range, where "0" > "Z")
- a combination of sequences and ranges such as "A : D Z"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

### compare with

The subkeys to be compared with the subkeys specified in the previous line have to be specified in this line.

You may specify one code or a sequence of codes for each subkey referring to the compare file. When using sequences the n'th code in the sequence specifies the compare code for the n'th reference code. Therefore the number of codes of both sequences has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the reference file.

If there is no input in this line, the same subkeys as specified in the line above are used for comparison.

## Copy entries:

This service lets you copy a selection of entries from one file into another or into the same file. You may specify new subkey names for the destination entries. You will be prompted for the file names and for an entry key selection.

### Work file selection screen

```
Table file services ----- SPEL ----- Copy entries
                                     Please enter file names

Name of the source file   => SPEL-USR TABFILE A
Name of the destinat. file => SPEL-USR TABFILE C

Enter= ok  1= Help  3= Quit
```

### Parameters:

#### Name of the source file

The name of an existing file containing entries to be copied.

#### Name of the destinat. file

The name of an existing file into which entries are to be copied. This file may be the same as the source file.

**Entry key selection screen**

```

Table file services ----- SPEL ----- Copy entries
                                     Please enter key selection

Region      ( 3 ch.) => D F
              copy to =>
Sub-region   ( 2 ch.) => 00
              copy to =>
Current year ( 2 ch.) => 85 : 89
              copy to =>
Periodicity ( 2 ch.) => 00
              copy to =>
Base year    ( 2 ch.) => NN
              copy to =>
Type         ( 4 ch.) => EXPC
              copy to => XXXC
Model area   ( 1 ch.) => S
              copy to =>

                                     Enter= ok  1= Help  3= Quit

```

**Parameters:**

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area see '**Compare entries: - Entry key selection screen**'.

**copy to**

The subkeys to be copied to from the subkeys specified in the previous line have to be specified in this line.

You may specify one code or a sequence of codes for each subkey referring to the source file. When using sequences the n'th code in the sequence specifies the copy to code for the n'th reference code. Therefore the number of codes of both sequences has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the source file.

If there is no input in this line, the same subkeys as specified in the line above are used for copy.

## Copy key structure:

This service lets you copy the key structure information (name and length of subkeys) from one file to another. The length of the whole key in the source and destination files must be the same.

### Work file selection screen

```
Table file services ----- SPEL ----- Copy key structure
                                Please enter file names

Name of the source file   => SPEL-SYS TABFILE F
Name of the destinat. file => SPEL-USR TABFILE A

                                Enter= ok  1= Help  3= Quit
```

### Parameters:

#### Name of the source file

The name of an existing file containing key structure information.

#### Name of the destinat. file

The name of an existing file into which the key structure information is to be copied.

### Create new file:

This service lets you create a new TABFILE file for direct access to data and text entries. You will be prompted for the name of the file to be created and information about the intended use. This information to be entered is important for determining the file space to be reserved for the new TABFILE file. It is advisable to reserve sufficient space, but do not overestimate and waste disk space.

After the file has been created successfully, the entry key structure should either be defined by the "Define key structure" service or copied from an existing file by the "Copy key structure" service.

### Work file selection screen

```
Table file services ----- SPEL ----- Create new file
                                     Please enter file name

Name of file to be created => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be created

A legal file name in the current operating system for the file to be created. The file must not already exist.



**File parameter screen**

```

Table file services ----- SPEL ----- Create new file
      R e q u i r e d   p a r a m e t e r s   :

Length of entry key in characters           => 16
Maximum number of entries                   => 5000

Max. length of entry description text in characters   => 80
Minimum number of data elements in an entry         => 0
Maximum number of data elements in an entry         => 3000
Estimated average number of data elements in an entry => 500

      O p t i o n a l   p a r a m e t e r s   :
      (normally computed by the system)

Logical record length preferred             =>
Number of records to be reserved           =>

      Enter= ok  1= Help  3= Quit  5= Test

```

**Required parameters:****Length of entry key in characters**

The number of characters in the entry key. The key length is fixed for all entries in the file.

According to SPEL rules for the key structure the length of the entry key will be 16 characters.

**Maximum number of entries**

This number will be used internally for computing requirements to reserve space for data in the created TABFILE file and thus determines the size of the work file.

No more entries will fit in the created file. The number given is internally rounded up to optimize disk space usage.

**Example:**

The new file should have space for 20 years, 15 regions and 10 types, all in all  
 $20 * 15 * 10 = 3000$  entries

Additional entry space should be reserved for 'deleted' entries and to allow storage of more tables in future.

**Max. length of entry description text in characters**

Maximum number of characters in the entry description text associated with each key. The legal range is 2 to 255.

According to SPEL rules the comment text to be stored with each SPEL table is of length 80 characters.

#### Minimum number of data elements in an entry

The minimum number means the lowest number of data elements over all entries to be stored in the file

The number specified has to be lower or equal to the number actually occurring.

SPEL 'basic table structure' and 'complementary table structure' tables are to be stored in TABFILE files. In SPEL the storing mode is always 'packed' (zero data are not explicitly stored). Therefore the number of data elements in the entries differs.

In extreme cases the number of data elements explicitly to be stored may decrease to zero, e.g. no data are available for special countries in the current year.

As entries are always 'packed' in SPEL, "=" has to be specified for the minimum number of data elements in an entry.

#### Maximum number of data elements in an entry

The maximum number means the highest number of data elements over all entries to be stored in the file.

The number specified has to be greater or equal to the number actually occurring.

SPEL tables of basic table structure and complementary table structure are to be stored in TABFILE files. In SPEL the storing mode is always 'packed' (zero data are not explicitly stored). Therefore the number of data elements in the entries differs.

Example:

There may be 2000 data elements estimated for maximum. Entering "3000" will create no disadvantages either.

#### Estimated average number of data elements in an entry

See "Minimum number of data elements in an entry".

Enter your estimation of the average number of data elements explicitly stored taking into account the total number of entries to be stored in the file.

Example:

The rough estimate may be about 450 data elements. To be on the safe side enter e.g. "500".

#### Optional parameters:

##### Logical record length preferred

Length of each record in the TABFILE file in bytes.

Do not enter a value if you are not familiar with details of the TABFILE file manager.

If not specified, the program computes a suitable record length. You can display the system's choice by pressing PF5.

The logical record length entered must be a multiple of 4 in the range 44 to 2048 (normal entries) or 48 to 2048 ("packed" entries).

### Number of records to be reserved

The total number of records to be reserved for the TABFILE file.  
Do not enter a value if you are not familiar with details of the TABFILE file manager.  
If not specified, the program computes a number suited to the specified number of entries and the average entry size. You can display the system's choice by pressing PF5.

#### Note:

You must specify the parameters marked "**required**". You should not specify the parameters marked "**optional**" if you are not familiar with details of the table file system.

#### Example for creating a new file :

- In accordance with SPEL rules for the key structure the total length will always be 16 characters. Therefore, "16" must be entered in the panel field "**Length of entry key in characters**".
- The new files should have space for tables for 20 years, 15 regions and 10 table types, all in all  $20 * 15 * 10 = 3000$  tables. Additional entry space should be reserved for "deleted" entries (see description of TABFILE files) and to allow storage of more tables in future. So "5000" may be entered in the panel field "**Maximum number of entries**". Do not specify too few entries!
- The comment text to be stored with each entry may not exceed 80 characters (SPEL always reserves 80 characters for storing the date and time of the last modification). An "80" must be entered in the panel field "**Max. length of entry description in characters**".
- SPEL tables of basic table structure and complementary table structure" tables are to be stored. In SPEL the storing mode is always "packed". Therefore the number of data elements in the entries differ. In extreme cases the number of data elements explicitly to be stored (in SPEL the number of non-zero elements) will decrease to zero, e.g. no data are available for special countries in the current year. Entering a "0" in the panel field "**Minimum number of data elements in an entry**" will create no disadvantages.
- The estimate for the maximum number of data elements may be 2000. Entering a "3000" in the panel field "**Maximum number of data elements in an entry**" will create no disadvantages either.
- The rough estimate for the average number of data elements explicitly to be stored may be in the vicinity of 450. To be on the safe side, enter "500" in the panel field "**Estimated average number of data elements in an entry**".

The figures entered in the panel fields "Maximum number of entries" and "Estimated average number of data elements in an entry" are the most important ones for determining the file space to be reserved. Overestimates only waste disk space. There is no need to input one of the "Optimum parameters".

After the file has been successfully created, the table key structure should either be defined by the service "Define key structure" or copied from an existing TABFILE file by the service "Copy key structure".

### Define key structure:

This service lets you define a key structure which divides the table key into up to ten subkeys.

You will be prompted for a TABFILE file name and the subkey definitions. Up to ten subkeys may be defined specifying name and length of each subkey.

The total of subkey lengths must be equal to the length of the whole key displayed at the top of the subkey definition panel.

### Work file selection screen

```
Table file services ----- SPEL ----- Define key structure
                                     Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be processed

The name of an existing TABFILE file for which the key structure is to be defined.

**Key structure definition screen**

```

Table file services ----- SPEL ----- Define key structure
Please define the subkeys for a total key length of 16 characters

Subkey 1 length => 3 name => Region
Subkey 2 length => 2 name => Sub-region
Subkey 3 length => 2 name => Current year
Subkey 4 length => 2 name => Periodicity
Subkey 5 length => 2 name => Base year
Subkey 6 length => 4 name => Type
Subkey 7 length => 1 name => Model area
Subkey 8 length => name =>
Subkey 9 length => name =>
Subkey 10 length => name =>

Enter= ok 1= Help 3= Quit

```

**Parameters:**

Up to ten subkeys can be defined, specifying the name and length of each subkey. The total subkey length must be equal to the length of the whole key displayed at the top of the subkey definition panel.

According to SPEL rules the total key length has to be 16 characters. The codes for the subkey names are limited to a length of 20 characters. For SPEL the following definitions have to be used :

```

subkey 1 length => 3 name => Region
subkey 2 length => 2 name => Sub-region
subkey 3 length => 2 name => Current year
subkey 4 length => 2 name => Periodicity
subkey 5 length => 2 name => Base year
subkey 6 length => 4 name => Type
subkey 7 length => 1 name => Model area

```

To remove an existing key structure, please define one subkey with the length of the whole key.

### Delete entries:

This service lets you delete entries in a file.

You will be prompted for the TABFILE file name and for a key selection of entries to be deleted.

The service does not free the file space occupied by the deleted entries. You have to use the 'Free unused space' service to make the space reusable.

### Work file selection screen

```
Table file services ----- SPEL ----- Delete entries
                               Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

                               Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be processed

The name of an existing TABFILE file for which entries are to be deleted.

**Entry key selection screen**

```
Table file services ----- SPEL ----- Delete entries
                                     Please enter key selection

Region          ( 3 ch.) =>
Sub-region      ( 2 ch.) =>
Current year     ( 2 ch.) =>
Periodicity     ( 2 ch.) =>
Base year       ( 2 ch.) =>
Type            ( 4 ch.) =>
Model area      ( 1 ch.) =>

                                     Enter= ok  1= Help  3= Quit
```

**Parameters:**

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see '**Compare entries: - Entry key selection screen**'.

**Remark :**

For maximum security no presets are shown in the screen above.

## Delete file:

This service lets you delete an existing TABFILE file.  
You will be prompted for the name of the TABFILE file to be deleted.

## Work file selection screen

```
Table file services ----- SPEL ----- Delete file
                          Please enter file name

Name of file to be deleted => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

## Parameters:

Name of file to be deleted

The name of an existing TABFILE file to be deleted.



**Edit entry data:**

This service lets you edit the data area of existing TABFILE entries.

You will be prompted for the TABFILE file name and for a key selection of entries to be edited.

The edit screen may be moved horizontally and vertically over the data table by using PF-keys. Additional PF-keys let you page through the selected entries.

**Work file selection screen**

```
Table file services ----- SPEL ----- Edit entry data
                                Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

**Parameters:**

Name of file to be processed

The name of an existing TABFILE file to be edited.

### Entry key selection screen

```
Table file services ----- SPEL ----- Edit entry data
                                Please enter key selection

Region      ( 3 ch.) => D F
Sub-region  ( 2 ch.) => 00
Current year ( 2 ch.) => 85 : 89
Periodicity ( 2 ch.) => 00
Base year   ( 2 ch.) => NN
Type        ( 4 ch.) => EXPC
Model area  ( 1 ch.) => S

                                Enter= ok  1= Help  3= Quit
```

#### Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see '**Compare entries: - Entry key selection screen**'.

#### Note:

You cannot create new entries.

**Edit panel**

Table file services -----		SPEL -----				Edit tables	
TABLE: D 00 80 00 00 CONC S							
	13 ZP13	14 ZP14	15 ZP15	16 ZP16	17 PR01	18 PR02	
17	FLAX	3.00	8.00	15.00	14.00	0.00	0.0
18	TOBA	0.00	0.00	0.00	0.00	751.94	100.0
19	OIND	0.00	0.00	0.00	0.00	0.00	0.0
20	CAUL	194.42	52.77	219.41	0.00	73.28	100.0
21	TOMA	0.00	0.00	0.00	0.00	110.72	100.0
22	OVEG	0.00	0.00	0.00	0.00	0.00	0.0
23	APPL	0.00	0.00	0.00	0.00	0.00	0.0
24	OFRU	0.00	0.00	0.00	0.00	0.00	0.0
25	CITR	0.00	0.00	0.00	0.00	0.00	0.0
26	TABO	0.00	0.00	0.00	0.00	0.00	0.0
27	NURS	0.00	0.00	0.00	0.00	0.00	0.0
28	FLOW	0.00	0.00	0.00	0.00	0.00	0.0
29	OCRO	0.00	0.00	0.00	0.00	0.00	0.0
30	TWIN	0.00	0.00	0.00	0.00	0.00	0.0
31	OWIN	0.00	0.00	0.00	0.00	0.00	0.0
32	OROO	205.73	102.87	401.18	1.00	0.00	0.0

Enter= ok 1= Help 2= Refresh 3= Quit 5= Top 6= Bottom  
7= Backward 8= Forward 9= Left 10= Right 11= Prev.Table 12= Next Table

An edit panel is like a window which can be moved over the whole table. The second header line always shows the key of the current table. The next line displays the sequence numbers and codes for table columns in the current window. At the beginning of the following lines you will find the sequence numbers and codes for table rows.

The data area at the intersection points of row and column codes may be modified.

Special PF-keys:

- PF2 : Re-establish the data as it was immediately after the last ENTER or PF-key processing.
- PF5 to PF10 : Move the screen window over the whole table.
- PF11/PF12 : Move to the previous/next table in the entry selection.

### Free unused space:

This service lets you free unused space, i.e. file directory space and file space occupied by 'deleted' entries or entry parts.

You will be prompted for the name of the TABFILE file to be processed.

The checks of the 'Test file consistency' service are included in this service.. If inconsistencies are found, the message 'Inconsistencies found' is displayed. Inconsistencies may only occur as a result of program errors or abnormal program endings. All incorrect entries detected will be removed. The result is always a formally correct file, but there is no guarantee that the check algorithm can detect all entries affected by the abnormal program run.

### Work file selection screen

```
Table file services ----- SPEL ----- Edit entry data
                                     Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be processed

The name of an existing TABFILE file to be processed

**List entry data:**

This service lets you list the data area of TABFILE entries.  
The service works like the service "Edit entry data" without modification facilities.

**List file directory:**

This service lets you list the entry names (keys) and entry text of a selection of entries in a file.  
You will be prompted for the TABFILE file name and for a key selection.

**Work file selection screen**

```
Table file services ----- SPEL ----- List file directory
                                Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

**Parameters:**

Name of file to be processed

The name of an existing TABFILE file for which the file directory is to be listed.

### Entry key selection screen

```
Table file services ----- SPEL ----- List file directory
                                     Please enter key selection

Region      ( 3 ch.) => NL
Sub-region  ( 2 ch.) => 00
Current year ( 2 ch.) => 60 : 99
Periodicity ( 2 ch.) => 00
Base year   ( 2 ch.) => NN
Type        ( 4 ch.) => CONC
Model area  ( 1 ch.) => S

                                     Enter= ok  1= Help  3= Quit
```

#### Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see '**Compare entries: - Entry key selection screen**'.

**Directory list screen**

```

Table file services ----- SPEL ----- List file directory

-----
Key                size  stored  text
-----
NL 00 80 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 81 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 82 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 83 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 84 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 85 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 86 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 87 00 00 COMC S 11040 1302 09.02.91 17:10:39
NL 00 88 00 00 COMC S 11040 1303 09.02.91 17:10:39
NL 00 89 00 00 COMC S 11040 1301 09.02.91 17:10:39

Total      10 entries listed

Enter= ok  3= Quit  4= Find

```

**Parameters:**

None.

**Explanation:**

- The "size" column shows the number of elements in the table. In SPEL this number is equal to rows multiplied by columns.
- The "stored" column shows the number of explicitly stored data elements. In SPEL it is the number of 'non-zero' elements in the table.
- The "text" column shows comments stored with the entry. In SPEL it corresponds to the date and time of the last modification.
- If the file directory list does not fit one screen, you will have additional PF-keys to move through the whole list.

**Special PF-keys:**

PF4: Displays a special field to enter a search string to be searched.

### List file status:

This service lets you list information about file characteristics, including capacity and usage information for the number of entries and records.  
You will be prompted for the TABFILE file name.

### Work file selection screen

```
Table file services ----- SPEL ----- List file status
                                     Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

### Parameters:

Name of file to be processed

The name of an existing TABFILE file for which the status is to be listed.



**Status screen**

Table file services ----- SPEL -----		List file status	
S T A T U S   O F   T H E   D A O P E N   F I L E   S Y S T E M			
Dir. space max.:	600000	active:	18620
deleted:	0	free:	581380
		free %:	96.9
Files in access:	1		
File:	SPEL-USR TABFILE H		
Status, priority:	1	I/O:	READ
Length, record:	512	key:	16
Entries max.:	1024	active:	931
deleted:	0	free:	93
		free %:	9.1
Records max.:	5673	active:	5241
deleted:	0	free:	432
		free %:	7.6
Enter= ok   3= Quit   4= Find			

**Parameters:**

None.

**Explanation:**

Files in access:

Always "1" in the "List file status" service.

Directory space:

Program core memory in bytes reserved for the internal table file directory.

Status priority:

Priority in the list of open files (always 1).

Status I/O:

Input/output status (always "READ").

Length, record:

Record length in bytes.

Length, key:

Key length in bytes.

Entries:

Table keys in the internal file directory.

**Records:**

Records in the file, the smallest internally addressable unit in the file.  
An entry (= "table") uses one or more records.

**Directories Space / Entries / Record :**

... max.:

Total reserved space, can never be exceeded.

... active:

Actually in use by accessible entries.

... deleted:

De-activated by implicit or explicit deletion of entries, can be made reusable by using the "Free unused space" service.

... free

Immediately usable for new entries.

## Print entry data:

This service lets you print the data of TABFILE entries as 'tables'.  
You will be prompted for the TABFILE file name and for a key selection of entries to be printed.

## Work file selection screen

```
Table file services ----- SPEL ----- Print entry data
.
Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok 1= Help 3= Quit
```

## Parameters:

Name of file to be processed

The name of an existing TABFILE file for which entries are to be printed.

### Entry key selection screen

```
Table file services ----- SPEL ----- Print entry data
                                     Please enter key selection

Region      ( 3 ch.) => NL
Sub-region  ( 2 ch.) => 00
Current year ( 2 ch.) => 60 : 99
Periodicity ( 2 ch.) => 00
Base year   ( 2 ch.) => NN
Type        ( 4 ch.) => COMC
Model area  ( 1 ch.) => S

                                     Enter= ok  1= Help  3= Quit
```

#### Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see '**Compare entries: - Entry key selection screen**'.

**Rename entries:**

This service lets you rename entries in a TABFILE file. This means that the key will change without copying or moving any data.

You will be prompted for the TABFILE file name and for an entry selection with rename specifications.

**Work file selection screen**

```
Table file services ----- SPEL ----- Rename entries
                                     Please enter file name

Name of file to be processed => SPEL-USR TABFILE A

Enter= ok  1= Help  3= Quit
```

**Parameters:**

Name of file to be processed

The name of an existing TABFILE file for which entries are to be renamed.

**Entry key selection screen**

```

Table file services ----- SPEL ----- Rename entries
                                     Please enter key selection

Region      ( 3 ch.) => NL
             rename =>
Sub-region  ( 2 ch.) => 00
             rename =>
Current year ( 2 ch.) => 60 : 88
             rename =>
Periodicity ( 2 ch.) => 00
             rename =>
Base year   ( 2 ch.) => NN
             rename =>
Type        ( 4 ch.) => COMC
             rename => XXXX
Model area  ( 1 ch.) => S
             rename =>

                                     Enter= ok  1= Help  3= Quit

```

**Parameters:**

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see '**Compare entries: - Entry key selection screen**'.

**rename**

The subkeys to be renamed from the subkeys specified in the previous line have to be specified in this line.

You may specify one code or a sequence of codes for each subkey referring to the old subkeys. When using sequences the n'th code in the sequence specifies the new code for the n'th reference code. Therefore the number of codes of both sequences has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes.

If there is no input in this line, the same subkeys codes as specified in the line above are used for rename.

## Test file consistency:

This service lets you test the file directory for inconsistencies which may only occur as a result of program errors or abnormal program endings.

If inconsistencies are found the message 'Directory wrong' is displayed. You can remove inconsistencies by using the service 'Free unused space', but you will lose data.

You will be prompted for the name of the TABFILE file to be processed.

## Work file selection screen

```

Table file services ----- SPEL ----- Test file consistency
                                Please enter file name

Name of file to be tested => SPEL-USR TABFILE A

                                Enter= ok  1= Help  3= Quit

```

### Parameters:

Name of file to be tested

The name of an existing TABFILE file for which the consistency is to be checked.

## 6.2.5. Print and export data from TABFILE file (DAOUT)

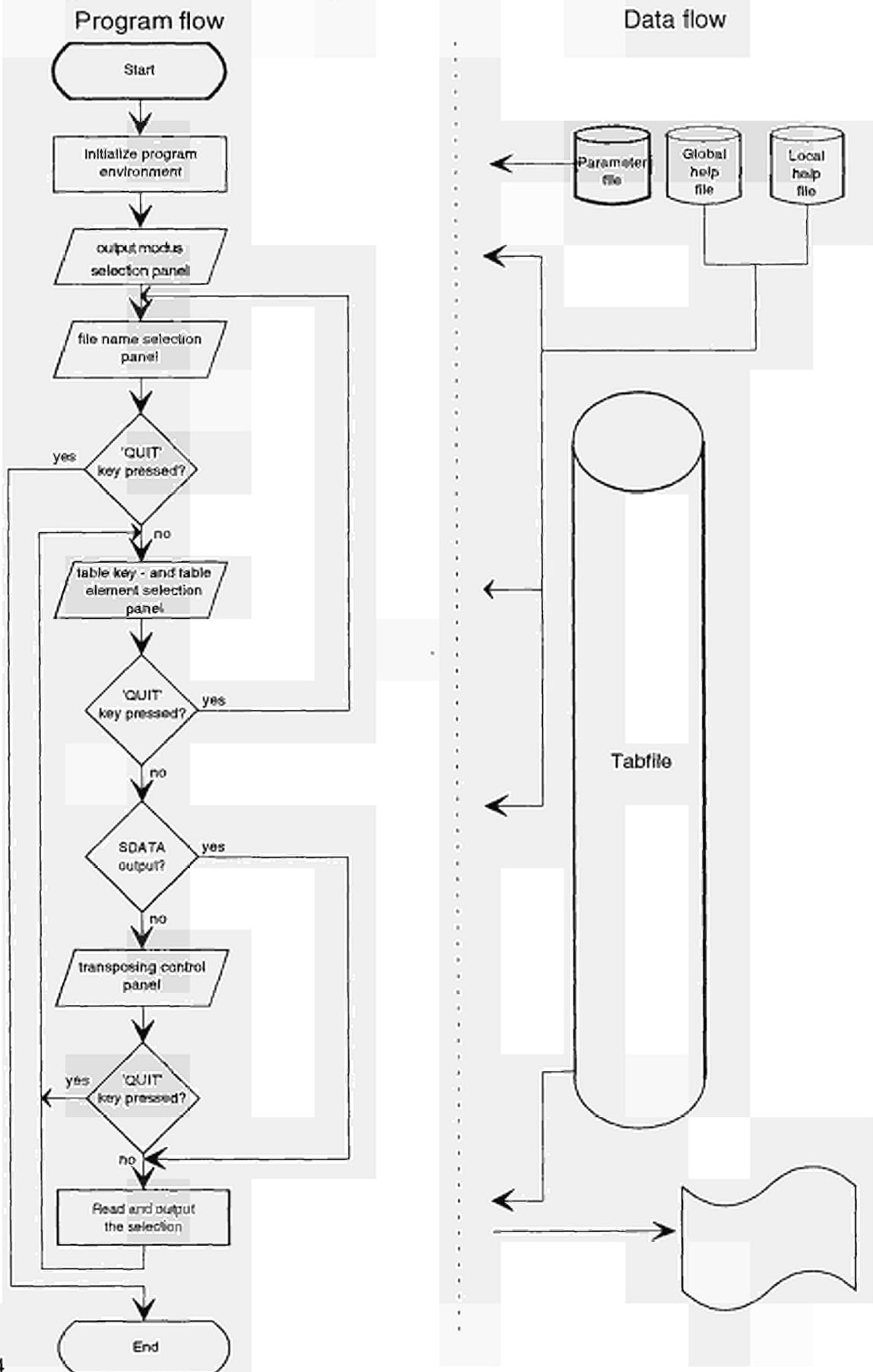
Selections of data from TABFILE files can be printed and exported by the DAOUT utility. The data selection is done by using a list of selection criteria, referring to the identifiers for the SPEL data dimensions (region, sub-region, year, periodicity, base year<sup>10</sup>, table type, table column and table row). There are four different output modi:

- printing file
- SDATA file
- file of CSV format for import to PC standard software, e.g. EXCEL or SYMPHONY
- file of German CSV format (including ";" as column separator)

The data dimensions can be transposed for output, except for SDATA output modus..

<sup>10</sup> Currently base year and table type are combined to the SPEL data dimension "status".  
See Figure 2: SPEL-Data dimensions

Figure 8 : DAOUT program flow chart





**Module startup screen**

```
Database printer ----- SPEL ----- Modul startup logo

          S P E L   D A T A B A S E   O U T P U T

Current settings for general files:
Protocol file   : SPEL LOG C
Print file     : DAOUT LISTING C

Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key, an edit panel will be displayed which will permit the changing of names of general files for the run time protocol (LOG-file) and documentation file (LISTING-file).

In the LOG-file the user will find a copy of all messages ever displayed on screen during program execution. More detailed information about work steps will be included as well.

The LISTING-file contains the "printed output".

**Parameters:**

None.

### Output modus selection screen

```
Database output ----- SPEL ----- Output modus
Select an output modus by moving the cursor to any topic below
then press ENTER

Print data
Export data in SDATA format
Export data in CSV format
Export data in German CSV format

Enter= ok 1= Help 3= Quit
```

#### Parameters:

For data output you can choose between four modi.

##### Print data

This modus provides printing of data in the LISTING file

##### Export data in SDATA format

The selected data from the TABFILE file are stored in a file of SDATA format.

##### Export data in CSV format

The selected data from the TABFILE file are stored in a file of CSV format.  
This format is used to transfer data to PC standard software like EXCEL or SYMPHONY.

##### Export data in German CSV format

The same format as CSV, but the ";" is used as column separation.

**Work file selection screen**

```
Database printer ----- SPEL ----- File name selection
                                     Please enter file name

. Table file    => SPEL-BAS TABFILE B

                                     Enter= ok  1= Help  3= Quit
```

**Parameters:****Table file**

TABFILE work file containing SPEL tables to print.

**Export file**

This parameter will not be displayed, if the output modulus "Print table" was selected.

Output file of SDATA or CSV format.

**Remark:** In the example above "Print table" was selected.

**Print selection screen**

```

Database printer ----- SPEL ----- Print selection

      Please select table keys, columns and rows

Region      ( 3 Ch.) => F IRL
Sub-region  ( 2 Ch.) => 00
Current year ( 2 Ch.) => 78 : 88
Periodicity ( 2 Ch.) => 00
Base year   ( 2 Ch.) => NN
Type        ( 4 Ch.) => BASB
Model area  ( 1 Ch.) => S
Table column ( 4 Ch.) -> POTA - RAPE
Table row   ( 4 Ch.) => LEVL

                        Enter= ok  1= Help  3= Quit

```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical range, where "0" > "Z")
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

**Sub-region**

Sub-regions are currently not used.

The whole region has the sub-region code "00".

Always specify "00".

**Current year**

Current year selection. Specify one of the following:

- a sequence of years to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"

All codes and the range operand ":" must be separated by blanks.

## Periodicity

Periodicity selection. Specify :

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
(alphanumerical expansion)
- a range defined by periodicity codes such as "Q1 - Q4"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "01 : 06 Q4"
- use '\*' to select all codes

All codes and the range operand ":" must be separated by blanks.

Periodicities are currently not used. The whole year has the code "00". Always specify "00".

## Base year

Current year selection. Specify one of the following:

- a sequence of years to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87"
- use "\*" to select all codes.

All codes and the range operand ":" must be separated by blanks.

For selecting SPEL tables from the SPEL work file the code for the subkey base year will be always "NN".

## Type

Type selection. Specify one of the following :

- a sequence of type codes to be selected such as "ZPAB COSC"
- a range defined by lower and upper bounds such as "AAAA : 9999"  
(alphanumerical expansion)
- a combination of sequences and ranges such as "ZPAB : COSC EXPC".
- use '\*' to select all codes.

All codes and the range operand ":" must be separated by blanks.

## Model area

Model area selection. Specify one of the following model area codes :

- "S" for supply,
- "D" for demand,
- "T" for trade

### Table column

Table column selection. Specify one of the following:

- a sequence of table column codes to be selected such as "PROP PRIC PRIN"
- a range defined by lower and upper bounds such as "BBBB : HHHH"  
(alphanumeric range)
- a range defined by row codes such as "PROP - PRIN"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "PROP - PRIN NAGG".
- use '\*' to select all codes.

All codes and the range operands ":" and "-" must be separated by blanks.

### Table row

Table row selection. Specify one of the following:

- a sequence of table row codes to be selected such as "PULS TOBA POTa"
- a range defined by lower and upper bounds such as "BBBB : HHHH"  
(alphanumeric range)
- a range defined by row codes such as "SWHE - POTa"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "SWHE - POTa OLIV OOIL".
- use '\*' to select all codes.

All codes and the range operands ":" and "-" must be separated by blanks.

**Remark:** The selection must not contain SPEL tables of different table structures.

### Transposing control screen

```
Database printer ----- SPEL ----- Transposing control
      Please enter transposing modes
C = Columns, L = Lines, 1 = lowest level tables .... 7 = highest level tables

      Transposing mode for: Region      => 1
      Transposing mode for: Sub-region  => 2
      Transposing mode for: Current year => L
      Transposing mode for: Periodicity => 4
      Transposing mode for: Base year   => 5
      Transposing mode for: Type        => 6
      Transposing mode for: Model area  => 7
      Transposing mode for: Table column => C
      Transposing mode for: Table row   => 3

      Enter= ok  1= Help  3= Quit
```

**Parameters:**

## Transposing mode for: Region

Transposing mode control for the "Region" dimension. Specify one of the following legal transposing mode codes:

- "C" for columns of output tables
- "L" for lines of output tables
- "1" for output table headers of lowest level
- ...
- "7" for output table headers of highest level.

The decimal codes "1", "2", "3", "4", "5", "6" and "7" control the sequence of the output tables where all table headers of level "1" (all regions in the example above) will follow immediately each another before the table header of level "2" will change.

The transposing mode codes must be unique.

## Transposing mode for: Sub-region

See "Transposing mode for: Region".

## Transposing mode for: Current year

See "Transposing mode for: Region".

## Transposing mode for: Periodicity

See "Transposing mode for: Region".

## Transposing mode for: Base year

See "Transposing mode for: Region".

## Transposing mode for: Type

See "Transposing mode for: Region".

## Transposing mode for: Model area

See "Transposing mode for: Region".

## Transposing mode for: Table column

See "Transposing mode for: Region".

## Transposing mode for: Table row

See "Transposing mode for: Region".

**Output example:**

The previous control panel input results in the following printer output:

```
-----
Table      1/001: Region IRL
              Sub-region 00
              Table row LEVL
              Periodicity 00
              Base year NN
              Type BASB
              Model area S
-----
```

	1	2	3
	POTA	SUGB	RAPE
1 78	53.40	35.20	0.00
2 79	41.20	36.50	0.00
3 80	40.60	34.90	2.20
4 81	41.56	32.98	0.00
5 82	36.00	34.90	0.00
6 83	37.40	35.50	1.42
7 84	33.50	36.40	3.03

Date: 28.11.91 14:09:22

```
-----
Table      1/002: Region IRL
              Sub-region 00
              Table row LEVL
              Periodicity 00
              Base year NN
              Type BASB
              Model area S
-----
```

	1	2	3
	POTA	SUGB	RAPE
8 85	35.70	34.90	4.45
9 86	33.00	33.90	4.45
10 87	30.50	37.00	2.02
11 88	30.30	37.10	5.66

Date: 28.11.91 14:09:22



## 7. INDEX

- active function key, → panel layout
- ASSIGN file, 32, 36, 37, 40, 43, 86, 87, 91
- ASSIGN format, → special control file format
- Authorize access, → DASERV program
- base year, → SPEL table subkey
- base year code, → code
- basic table structure, → table structure
- CMS dialog system, 14
- code
  - base year code, 24, 45, 139
  - compare code, 106
  - current year code, 23, 25
  - data code, 23, 36
  - model area code, 24, 45, 52, 94, 139
  - originator type code, 23
  - periodicity code, 23, 25, 45, 52, 93, 105, 139
  - reference code, 106
  - region code, 23, 24, 38, 44, 51, 92, 104, 138
  - source data code, 36, 43, 91
  - sub-region code, 23, 24, 38, 44, 51, 92, 104, 138
  - table column code, 25, 36, 140
  - table element code, 24, 36, 37, 38, 40, 43, 91
  - table row code, 25, 36, 140
  - type code, 24, 36, 45, 52, 59, 86, 105, 139
- code conversion, **86**
- CODES file, **21**, 44, 45, 51, 52, 139
- coefficient of determination, 54
- compare code, → code
- Compare entries, → DASERV program
- complementary table structure, → table structure
- control language, → EV program
- control level text, → panel layout
- Copy entries, → DASERV program
- Copy key structure, → DASERV program
- Create new file, → DASERV program
- CRONOS, 22, 86
- CSV format, 136, 137
- current year, 44, 52, 93, 105, 108, 117, 120, 124, 130, 132, 138
- current year code, → code
- DAOUT program, **133**
- DASERV program, 26, 27, **94**
  - Authorize access, 97, **99**
  - Compare entries, 97, **102**
  - Copy entries, 97, **107**
  - Copy key structure, 97, **109**, 110, 113
  - Create new file, 97, **110**
  - Define key structure, 98, 110, 113, **114**
  - Delete entries, 98, **116**
  - Delete file, 98, **118**
  - Edit entry data, 98, **119**
  - Free unused space, 98, 116, **122**, 128, 133
  - List entry data, 98, **123**
  - List file directory, 98, **123**
  - List file status, 98, **126**
  - Print entry data, 98, **129**
  - Rename entries, 98, **131**
  - Test file consistency, 98, 122, **133**
- data code, → code
- data element, 112, 113, 125
- data management, **22**
- data status flag, **23**
- DATCON program, 22, **86**
- DEFAULT PARAM file, → general program control
- default PF-keys, → PF-key
- Define key structure, → DASERV program
- Delete entries, → DASERV program
- Delete file, → DASERV program
- dependent variable, **38**, 76, 78

description text, 62, 63

destination entry, 97, 107

destination file, 97, 107, 109

display panel, → panel

documentation level, 54

Edit entry data, → DASERV program

edit panel, → panel

entry, → TABFILE

error messages, 31, 34, 42, 49, 89, 96

ERROR-file, 42, 49, 89, 96

EV program, 36, 39, **55**

- addition of constants, 71
- addition of table elements, 71
- Application abstracts, 55
- arithmetical calculations, 70
- close the work file currently open, 65
- column numbering
  - disable, 80
  - enable, 80
- command language, 55
- control language
  - details, 57
  - structure, 56
- conversion of growth rate to indices, 73
- data file types, 56
- define EV-tables
  - additional, 62
  - new, 61
- definition of variance, 76
- deletion of tables, 69
- disable echoing, 60
- division by base year elements, 73, 74
- division by constants, 72
- division of table elements, 71
- editing of Data, 70
- enable echoing, 60
- EV session (example), 82
- EV-table, 55
  - listing or printing, 77
- exit from session, 61
- exponential operation for table elements, 73
- general control parameters, → EV program \$CONTROL
- Growth rates in percent for table elements, 73
- list normal tables on the screen, 78

- list regression result tables on the screen, 78
- listing of correlation coefficients and elasticities
  - disable, 77
  - enable, 76
- logarithm (base 10) of table elements, 72
- logarithm (natural) of table elements, 72
- macros, 59
- main commands, 57
  - \$ARITH, 70
    - ADD, 71
    - ADDC, 71
    - BASIS, 73
    - CLEAN, 74
    - DIV, 71
    - DIVB, 74
    - DIVC, 72
    - EXP, 73
    - GROWTH, 73
    - HELP, 75
    - INDEXGR, 73
    - LN, 72
    - LOG, 72
    - MAVC, 74
    - MUL, 71
    - MULB, 74
    - MULC, 72
    - OVERLAY, 75
    - POTC, 72
    - POW, 73
    - RESET, 74
    - REZ, 72
    - ROOTC, 72
    - SUB, 71
    - SUBC, 71
  - \$COMMENT, 81
  - \$CONTROL, 58
    - BASEYEAR, 59
    - CALL, 59
    - ECHOOF, 60
    - ECHOON, 60
    - ERROFF, 60
    - ERRON, 60
    - FROMYEAR, 58
    - HELP, 60
    - INCLUDE, 58
    - STATUS, 59
    - TOYEAR, 59
    - TYPIN, 59
    - TYPOUT, 59
  - \$DATA, 66
    - DATIN, 67

- DATOUT, 68
- DELETE, 69
- EDIT, 70
- GEN, 67
- LAG, 68
- SELECT, 67
- STDFILE, 66
- TRANSP, 69
- TYPIN, 67
- YEARS, 68
- \$DEFINE, 62
  - HELP, 63
  - RENAME, 63
  - TREND, 62
- \$DEFOUT, 63
- \$HELP, 60
- \$MATRIX, 63
  - CLOSE, 65
  - GEN, 64
  - HELP, 66
  - MATIN, 65
  - MATOUT, 65
  - MATOUTR, 65
  - SELECT, 64
  - TABFILE, 64
  - TYPIN, 64
- \$NEW, 61
  - HELP, 63
  - RENAME, 63
  - TREND, 62
- \$REGRESS, 75
  - ESTIMATE, 76
  - LISTOFF, 77
  - LISTON, 76
  - PCTNEW, 76
  - PRINTOFF, 77
  - PRINTON, 77
  - SELECT, 75
- \$SKIPOFF, 61
- \$SKIPON, 61
- \$STOP, 61
- \$TABOUT, 77
  - CNUMOFF, 80
  - CNUMON, 80
  - COMMENT, 79
  - LEGOFF, 80
  - LEGON, 80
  - LISTD, 78
  - LISTR, 78
  - PAGE, 80
  - PRINTD, 78
  - PRINTOFF, 79
  - PRINTON, 79
- PRINTR, 79
- RECORD, 80
- ROWS, 80
- model area subkey, 64, 67
- moving averages, 74
- multiple linear regression analysis, 75
- multiplication by base year elements, 74
- multiplication by constants, 72
- multiplication of table elements, 71
- non-arithmetical data treatment, 66
- obtain data, 55
- on-line help, 60, 63, 66, 75
- operations, 55
- power of table elements, 72
- power operation for table elements, 73
- print comments, 79
- print normal tables, 78
- print regression result tables, 79
- printer page height, 80
- printer page width, 80
- printing of comments, 81
- printing of correlation coefficients and elasticities
  - disable, 77
  - enable, 77
- printing of table legends
  - disable, 80
  - enable, 80
- processing of SDATA files, 66
- program, 55
- reading data from TABFILE file, 63
- reading of data, 65, 67
- reciprocal of table elements, 72
- regression estimation, 55, 76
- rename table, 63
- replacement of table elements, 74
- replacement of zero table elements, 75
- resetting of table elements, 74
- root of table elements, 72
- SDATA files to be used, 66
- select table rows to be printed, 80
- selection of tables, 64, 67
- session, 56
- skipping
  - disable, 61
  - enable, 61
- STEER file, 55
- subsequent printing of tables
  - disable, 79
  - enable, 79
- subtraction of constants, 71
- subtraction of table elements, 71
- table definition lines, → \$DEFINE, \$NEW

- table definitions, list of, 63
  - tables
    - listing or printing, 77
  - time lags, 68
  - time period for output, 68
  - transposing the current data, 69
  - trend table, 62
  - type overlay sequence, 67
  - type overlay sequence for SPEL tables, 64
  - variables, 56
  - variables to be entered in the regression, 75
  - work file, 64
  - writing data into TABFILE file, 63
  - writing of data, 65, 68
  - writing protection
    - disable, 60
    - enable, 60
- evaluation program, → EV program
- export data, → DAOUT program
- FAO, 22, 86
- file consistency, → DASERV program
- file directory, → TABFILE
- file status, → DASERV program
- FILTER program, 22, 36, 37, **86**, 91
- format conversion, 22, 86, 87
- FORTTRAN 77, 14, 21
- Free unused space, → DASERV program
- general program control, **29**
  - general control file, 29, 35
    - DEFAULT PARAM file, **29**, 35
    - PARAM file, **29**, 30, 31, 35, 36, 47, 86
    - parameter block, 29, 35, 86
  - parameter file, 35
  - program control parameter, 29, 31
- help, **34**
  - HELP file, **34**
    - local HELP file, **34**
  - help key, 32, 34
  - help text, 34
  - on-line help, 34
  - on-line help (EV), 60, 63, 66, 75
- independent variable, **38**, 55, 75, 76
- key, **17**
  - key length, **111**, 115, 127
  - key structure, 27, 97, 98, 109, 110, 111, 113, **114**, 115
- List entry data, → DASERV program
- List file directory, → DASERV program
- List file status, → DASERV program
- LISTING-file, 49, 96, 135
- local HELP file, → help
- LOG-file, 42, 49, 89, 96, 135
- Logical record length, 112
- methodological documentation, 13
- model area, → SPEL table subkey
- model area code, → code
- model work file, 23
- multi-purpose program, 40
- non SPEL data file format, 22
- obtain data, → EV program
- on-line help, → help
- operations, → EV program
- operator, **14**
- operator level, 14
- ORIGIN program, 23, 31, 32, 36, 37, 40, 43
- originator type code, → code
- Output level, 54
- panel, 29, 30, 35
  - display panel, 30, 32, 34
  - edit panel, 30, **31**, 32, 35, 42, 89, 135
  - file selection panel, 31
  - selection panel, 30, 32
- panel layout, 31
  - active function key, 31
  - control level text, 31
  - parameter description, 31
  - parameter field, 31
  - parameter value, 31
  - PF-key description line, 31
  - top line, **31**
- PARAM file, → general program control
- parameter block, → general program control
- parameter description, → panel layout

- parameter field, → panel layout
- parameter file, 35
- parameter value, → panel layout
- periodicity, → SPEL table subkey
- periodicity code, → code
- PF-key, **31**, 33, 119, 121, 125
  - default PF-key, 33
- PF-key description line, → panel layout
- Print entry data, → DASERV program
- print tables, → DAOUT program
- printed output, 135
- program control parameter, → general program control
- program selection shell, 16, 55, 86
- projection year, 19
- read only user, 101
- read/write user, 101
- reference code, → code
- reference file, 103, 106
- region, → SPEL table subkey
- region code, → code
- region selection, 58
- REGSTE file, 36, 37, 47, 50
- REGSTE format, → special control file format
- Rename entries, → DASERV program
- SCRATCH file, 96
- SDATA file, 22, **24**, 26, 36, 39, 40, 56, 57, 86, 87
- SDATA format, → SPEL standard data file format
- selection panel, → panel
- Significance coefficient, → trend estimation
- simulation
  - simulation result, 19, 20
  - simulation year, 47
- source data code, → code
- source file, 107, 108, 109
- special control file, 29, **36**
- special control file format
  - ASSIGN format, 36
  - REGSTE format, 37
- SPEL model, 16, 17, 21, 40
- SPEL program, 16, 26, 27, 29, 35, **40**, 77
- SPEL software, 14, 16
- SPEL standard data file, 22
- SPEL standard data file format, 22
  - SDATA format, 22, 40, 43, 90, 91, 93, 94
  - SPELIMP format, 22, 40, 43, 86, 90, 91, 93, 94
  - TABFILE format, 22, 43, 50
  - SDATA format, 136, 137
- SPEL system, 13, 22
- SPEL table, 26, 111, 137
  - SPEL table column, 140
  - SPEL table dimension, **18**
  - SPEL table element, 37, 40
  - SPEL table row, 140
  - SPEL table subkey, **17**, 24
    - base year, **20**, 45, 93, 105, 108, 117, 120, 124, 130, 132, 139
    - model area, **20**, 45, 52, 94, 106, 108, 117, 120, 124, 130, 132, 139
    - periodicity, **20**, 45, 52, 93, 105, 108, 117, 120, 124, 130, 132, 139
    - region, **19**, 44, 51, 92, 104, 108, 117, 120, 124, 130, 132, 138
    - sub-region, **19**, 44, 51, 92, 104, 108, 117, 120, 124, 130, 132, 138
    - type, 45, 46, 93, 105, 108, 117, 120, 124, 130, 132, 139
    - year, **19**, 44, 52, 93, 105, 108, 117, 120, 124, 130, 132, 138
  - table element code, → code
- SPEL table subkey, → SPEL table
- SPEL work file, 43, 45, 139
- SPEL-Data, 16, 17, 22, 56, 94
  - SPEL-Data dimension, **17**
- SPEL/EC Model, 17, 19, 20
- SPEL/EC model, 13
- SPEL/EC-ROW model, 17, 19, 20
- SPELIMP file, 22, 23, 36, 40, 86, 87

SPELIMP format, → SPEL standard data file format

status, 18, **20**

STEER file, 36, **39**, 55, 59

Sub-region, → SPEL table subkey

sub-region code, → code

subkey, 17, 106, 108, 109, 114, 115, 132

- subkey code, 24, 106, 108, 132

- subkey length, 97, 114, 115

- subkey name, 97, 104, 107, 115

#### TABFILE

- entry, 98, 119, 123, 129

- entry data, 129

- entry data, 98, 119, 123

- entry description text, 111, 113

- entry key, 27, 111, 113

- entry key selection, 102, 107

- TABFILE file, 22, 23, **26**, 27, 50, 56, 57, 58, 63, 97, 98, 110, 111, 112, 113, 114, 116, 118, 119, 122, 123, 126, 129, 131, 133, 136

- file directory, 27, 65, 66, 94, 98, 122, 123, 125, 127, 133

- TABFILE file manager, 26, 112, 113

- TABFILE format, → SPEL standard data file format

- TABFILE maintenance, → DASERV

- TABFILE work file, **40**, 47, 50, 56, 60, 64, 94, 137

#### table

- table column, 18, **20**

- table column code, → code

- table element, **17**, 27

- table key, **26**, 27, 93, 94, 98, 114, 127

- table key structure, 113

- table row, 18, **21**

- table row code, → code

- table structure, **17**, 36, 40

- basic table structure, **17**, 20, 36, 112, 113

- complementary table structure, **17**, 20, 36, 112, 113

- table structure type, 20

- table subkey, 98

Test file consistency, → DASERV program

Theil's coefficient, 54, 55

time series, 24, 47, 49, 50, 54

time series table, 24, 39

top line, → panel layout

transposing mode, 141

trend estimation, 36, 37, 47, 49

- regression, 37, 49

- best fit, 47, 54

- coefficient of determination, 47

- data transformation, 47, 49, 54

- data transformation type, 38, 39, 50

- dependent variable, 47

- reference intervals, 50

- reference period, 47

- regression control, 47

- regression control parameter, 47

- regression reference interval, 53

- regression steering file, 50

- regression steering record, 38

- significance coefficient, 54

- single regression, 47

- Theil's coefficient, 47

TREND program, 36, 37, **47**, 50

type code, → code

user, **14**

user interface, 13, **29**

user level, 14

utility, 26, 40, **55**

warnings, 31

wildcard character, 31, 38

work file, **26**, 47, 54, 96, 111

year, → SPEL table subkey

years to be processed

- base, 59

- first, 58

- last, 59







# **PART 2**

Operator/User's Guide, SPEL/EC Model



# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>155</b>
<b>2. MODEL COMPONENTS AND IMPLEMENTATION .....</b>	<b>156</b>
2.1. Components of the SPEL/EC Model .....	156
2.2. SPEL/EC shell and environment .....	160
2.2.1. SPEL/EC shell .....	160
2.2.2. Initializing environment.....	160
<b>3. DATA PREPARATION .....</b>	<b>162</b>
3.1. Data extraction and compilation .....	164
3.1.1. Data format conversion (DATCON).....	165
3.1.2. Optional data selection (FILTER) .....	167
3.1.3. Data compilation (ORIGIN) .....	167
3.2. Data set completion .....	168
3.2.1. Completion by subjective estimation (EV).....	168
3.2.2. Completion by trend estimation (TREND).....	169
<b>4. MODEL CALCULATIONS.....</b>	<b>172</b>
4.1. Base Model (BM) .....	174
4.1.1. Consistency at Member State level and EC aggregation level (BASEMOD) .....	174
4.1.2. Optional aggregation of user defined regions (AGGREG).....	180
4.2. Short term forecast and simulation system (SFSS) .....	186
4.2.1. System proposals for exogenous variables (SFPROP).....	188
4.2.2. Forecast simulation (SFS).....	197
4.2.3. Optional aggregation of user defined regions (AGGREG ).....	202
4.3. Extensions to steering file formats for simulation systems .....	202
<b>5. UTILITIES.....</b>	<b>204</b>
5.1. Maintenance of TABFILE files (DASERV) .....	204
5.2. Selection and conversion of SPELIMP and SDATA files (FILTER) .....	205
5.3. General estimation and evaluation program (EV) .....	205
5.4. SPEL data base output (DAOUT).....	206
5.5. Link of the SPEL disks .....	207
5.6. File list of the output disk.....	208
5.7. CMS command level .....	208
<b>6. EXPLOITATION .....</b>	<b>209</b>
<b>7. ANNEX A: ERROR MESSAGES AND WARNINGS .....</b>	<b>211</b>
<b>8. INDEX.....</b>	<b>240</b>

## LIST OF FIGURES

<b>Figure 1 :</b>	<b>Components of the SPEL/EC Model .....</b>	<b>157</b>
<b>Figure 2 :</b>	<b>Work steps of the SPEL/EC Model .....</b>	<b>158/159</b>
<b>Figure 3 :</b>	<b>Work sectors of Data preparation .....</b>	<b>163</b>
<b>Figure 4 :</b>	<b>Work steps of Data preparation without additional data reduction step .....</b>	<b>170</b>
<b>Figure 5 :</b>	<b>Work steps of Data preparation with additional data reduction step .....</b>	<b>171</b>
<b>Figure 6 :</b>	<b>Work sectors of Model calculations.....</b>	<b>173</b>
<b>Figure 7 :</b>	<b>BASEMOD program flow chart .....</b>	<b>175</b>
<b>Figure 8 :</b>	<b>AGGREG program flow chart .....</b>	<b>181</b>
<b>Figure 9 :</b>	<b>SFPROP program flow chart .....</b>	<b>189</b>
<b>Figure 10 :</b>	<b>SFS program flow chart .....</b>	<b>198</b>

# 1. INTRODUCTION

The SPEL approach is characterized by mutual interaction between model-builders and statisticians or policy-makers. There are different models to deal with different questions, e.g. the SPEL/EC Model for the agricultural sector of the EC Member States and the EC as a whole.

In general the SPEL system offers a systematically structured data system to be used for

- checking the consistency of agricultural statistics,
- monitoring the present situation in agricultural sector,
- ex-post analyses of sectoral developments, and
- forecasts and policy simulations of the effects of alternative policies from short-term and medium-term viewpoints.

The set of statistical data is not taken as final but subjected to consistency checks and critical investigations, which may lead to the jointly agreed revision of existing statistics and proposals for amendments and conceptual changes for new statistics. It is not assumed either that policy-makers are able to specify target variables before taking a closer look at the problems involved. Instead, the fundamental idea is that target priorities emerge during the process of policy dialog on trade-offs between target variables.

The principle technical approach for the whole SPEL system is transparency and flexible user interface. Transparency means that each data element can be traced back to the basic data sources and the underlying assumptions. The flexible user interface facilitates dialogue between policy-makers and the model.

Some general features of the SPEL/EC Model can be characterized by the items:

- activity-based concept: division of the agricultural sector into activities, in order to trace productions interactions;
- consistency accounting framework: to balance physical and monetary flows.

The technical documentation of the SPEL system consists of the following parts :

Basic information

- Part 1: Operator/ User's Guide : Basics

Specific information for the SPEL/EC Model

- Part 2: Operator/ User's Guide : SPEL/EC Model

The technical documentation is designed to be a guide for persons familiar with the VM/CMS system on an IBM or compatible main frame.

A knowledge of the basic information as described in "SPEL System: Technical Documentation Part 1", will make it easier to follow and to appraise the details of this volume.

For the methodological aspects see the methodological documentation.

## **2. MODEL COMPONENTS AND IMPLEMENTATION**

To ensure the transparency of the SPEL/EC Model the technical program flow is structured.

This structure is implemented by flexible user interfaces realized by panels leading the operator/user through the SPEL/EC Model.

### **2.1. Components of the SPEL/EC Model**

The SPEL/EC Model is divided into two model components :

- Data preparation
- Model calculations

The Model calculations component works on the results of the Data preparation component.

The components are divided into several work sectors. The work sectors reflect the main methodological aspects.

Each work sector is technically implemented by a set of work steps, which are applications of separate computer programs.

Figure 1 gives an overview of the SPEL/EC Model components, their work sectors and work steps.

**Figure 1 : Components of the SPEL/EC Model**

Model components	Work sector	Work steps
Data preparation	Data extraction and compilation  Data set completion	Data format conversion (DATCON program) Optional data selection (FILTER program) Data compilation (ORIGIN program)  Completion by subjective estimation (EV program) Completion by trend estimation (TREND program)
Model calculations	Base Model (BM)  Short term forecast and simulation system (SFSS)	Consistency at Member State level and EC aggregation level (BASEMOD program) Optional aggregation of user defined regions (AGGREG program)  System proposals for exogenous variables (SFPROP program) Forecast simulation (SFS program) Optional aggregation of user defined regions (AGGREG program)

The programs BASEMOD, AGGREG, SFPROP and SFS are methodologically adjusted programs for the SPEL/EC Model and the descriptions are included in this volume. The others are utilities or multi-purpose programs described in "SPEL System: Technical Documentation Part1". Additional reports for the results of work steps can be generated by the EV program and the DASERV program.

Figure 2 shows the complete sequence of work steps for the SPEL/EC Model. Each step works on the results of the previous one. Therefore the control parameters have to be checked for consistency; e.g. if the previous work step has run for the region UK only, the next step must not access other regions. If a work step has to be repeated with different input data or control parameters the sequence can be restarted at the points marked in figure 2.

Figure 2 : Work steps of the SPEL/EC model

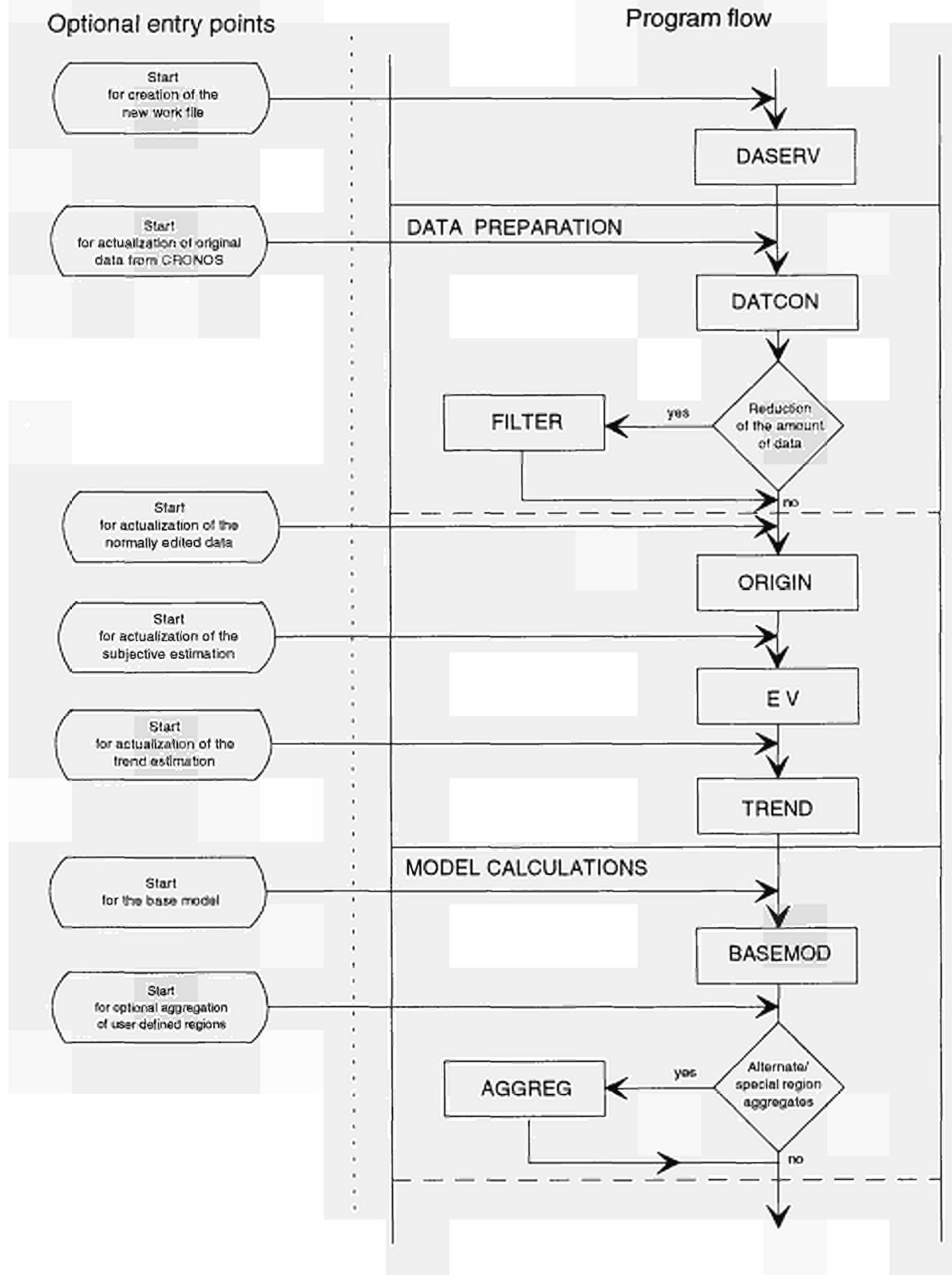
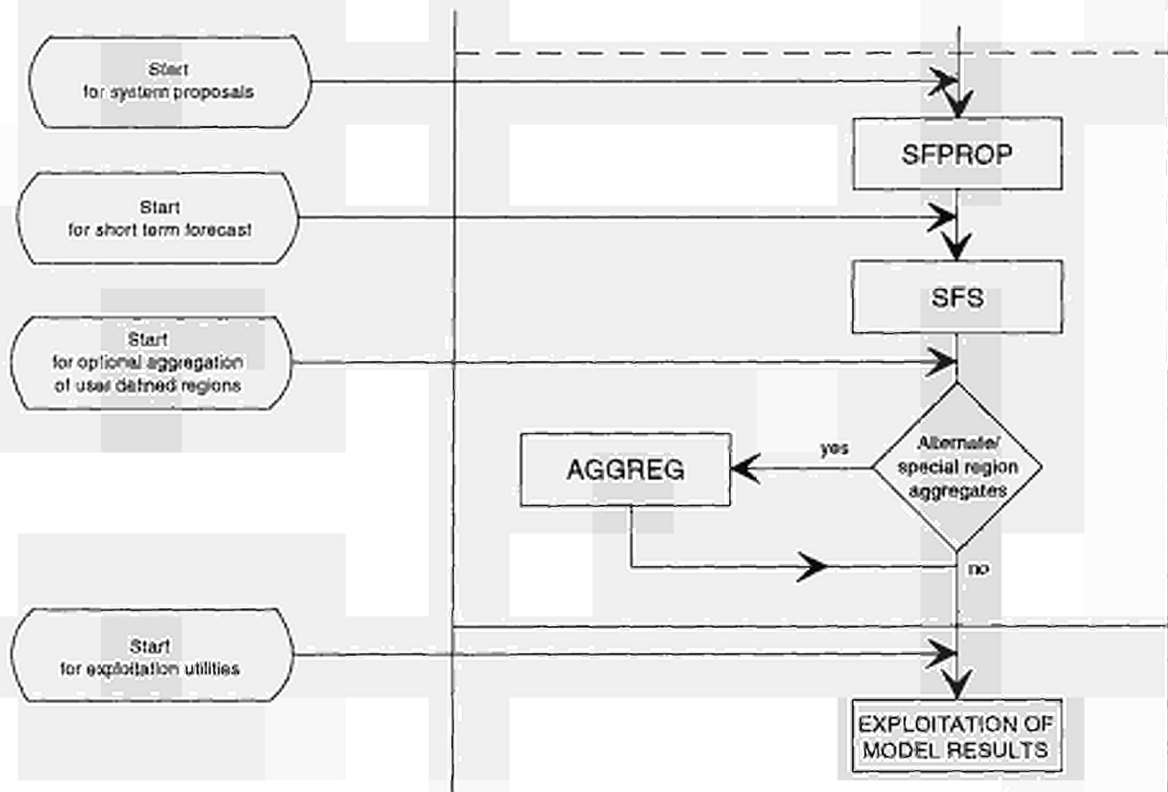




Figure 2 (continued) : Work steps of the SPEL/EC model



## 2.2. SPEL/EC shell and environment

### 2.2.1. SPEL/EC shell

The SPEL/EC shell is a full screen interface provided to work on the SPEL/EC Model. The sequence of the panels reflects the model components, work sectors and work steps as described in figure 1.

The additional utilities for free disposal are offered to the operator/user.

The SPEL/EC shell is composed of: selection ; edit ; and display panels. The SPEL/EC shell panel layout and usage are identical to the work step panels in SPEL, but are installation dependent and managed by the SPEL team, Luxembourg. (see "SPEL System: Technical Documentation Part1")

Before using the SPEL/EC shell make sure that the SPEL EXEC program is on your A disk. This program is distributed by the SPEL team, Luxembourg.

To start the SPEL/EC shell, type SPEL in the CMS command level and press ENTER. This command may be added to the PROFILE EXEC file in order to start the SPEL shell automatically at the beginning of the user's session. If the user intends to work without the SPEL/EC shell he may contact his SPEL team.

### 2.2.2. Initializing environment

The SPEL environment is initialized by the execution of the SPEL EXEC command.

Initializing the user environment means that the SPEL programs and utilities are made available by linking the following disks :

C : temporary disk linked with R/W access (only if the C disk mode is free for use)

F : SPEL/EC system disk providing the executable SPEL programs, linked in R/O

I : SPEL system disk providing general programs, linked in R/O.

Initializing the operator environment four additional disks may be linked:

E : SPEL operator system disk linked in R/O

H : SPEL/EC disk for model simulation results, linked in R/O

J : SPEL/EC disk for Base Model data (SPEL/EC-Data), linked in R/O

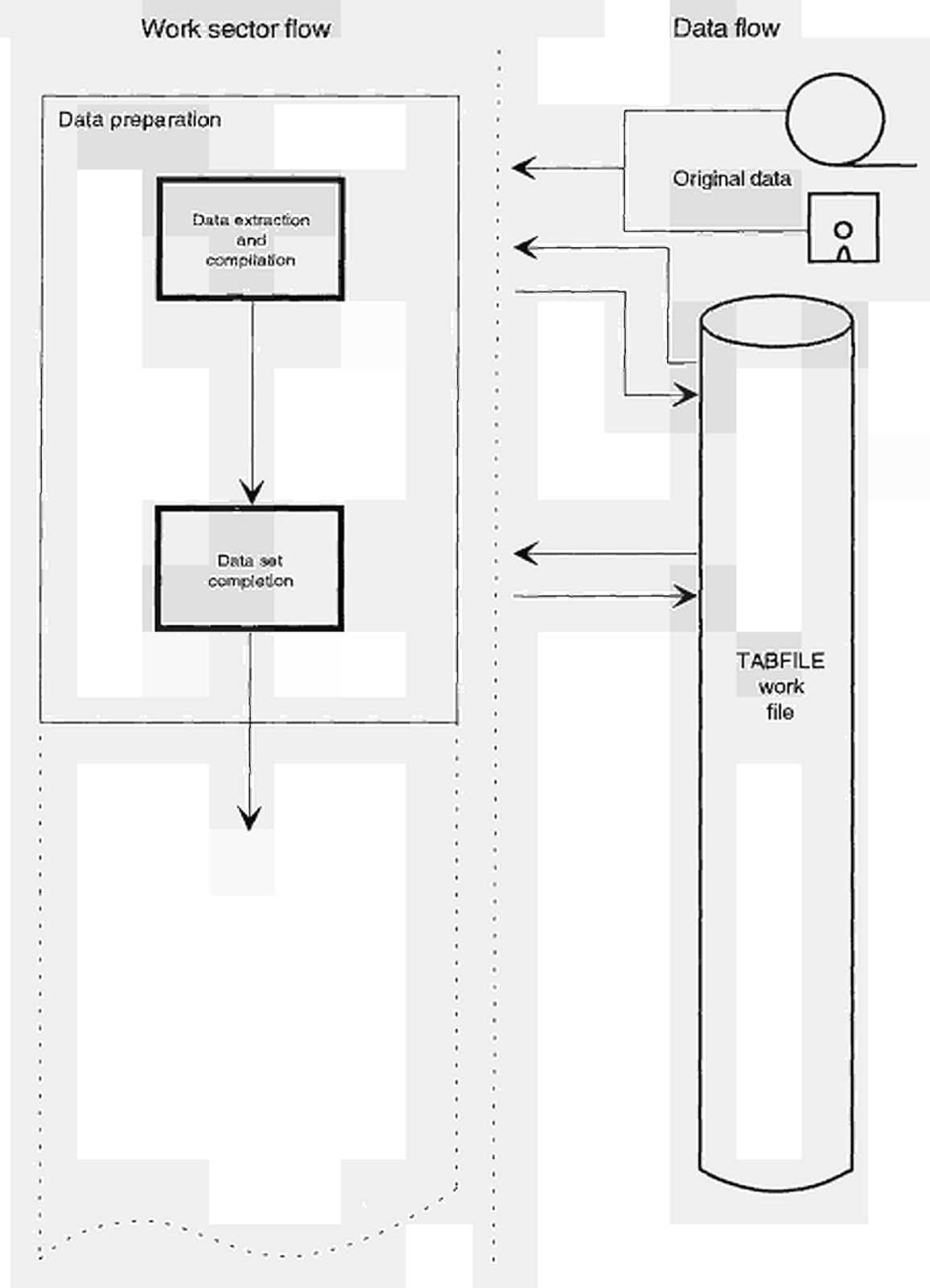
L : SPEL/EC disk containing exogenous data mostly stored in SPELIMP files, linked in R/O

The default A disk is still accessible for users and operators in current mode.





Figure 3 : Work sectors of Data preparation



### 3.1. Data extraction and compilation

The work sector Data extraction and compilation is mainly done in two work steps (DATCON and ORIGIN). A third work step (FILTER) for reducing the amount of source data can be added.

The following panel of the SPEL/EC shell lets the operator select a work step.

#### Work step selection screen

```
Program starter ----- SPEL/EC ----- Work step selection

                                     DATCON: Data format conversion
                                     FILTER: Optional data selection
                                     ORIGIN: Data compilation

                                     Select an item by moving the cursor to any topic above
                                     Enter= ok  1= Help  3= Quit
```

In the DATCON work step the different external data source formats are converted to SPELIMP format.

Optionally an additional data selection step, FILTER, can be inserted beforehand ORIGIN in order to reduce the amount of external source data (figure 5).

The ORIGIN work step imports data into the model work file (TABFILE work file).

Whenever a program is started from the SPEL/EC shell a file selection panel is displayed to define the parameter file name:

**PARAM file selection screen**

```
Program starter----- SPEL/EC ----- PARAM file selection
                        Please enter STEER file name

PARAM file      =>

Enter= ok  1= Help  3= Quit
```

**Parameters:****PARAM file**

Name of the parameter file containing presets for program parameters.  
If no file name is specified, the DEFAULT PARAM file is used.

**3.1.1. Data format conversion (DATCON)**

Most exogenous data files are stored on tapes. The format conversion has to be done on a computer system which can access tape units (batch component of the VM/CMS operating system).

The format standardization process, which includes a complete sorting step, is CPU time and I/O consuming. Therefore, it is recommended that the format conversion should always be run in batch mode.

The format conversion DATCON program is steered by a parameter specifying the external source format. The output format is always SPELIMP.

Format standardization should always be done by the operator. Users should contact the SPEL team. For the operator the SPEL/EC shell offers a panel to specify the domain type and the tape-id.

### Domain/tape selection screen

```
Data format conversion ----- SPEL/EC ----- Domain/tape selection
                                Extract/sort data from tape to reader
                                (in VM/BATCH)

Domain          =>
Tape-id         =>

Enter= ok  1= Help  3= Quit  9= Submit
```

#### Parameters:

##### Domain

Select one domain type code.

The following domain types are currently supported:

COSA, ICG\_, PRAG, SEC1, ZPA1

##### Tape-id

Enter the tape-id of that tape where the data of the selected domain are stored.

For all domains default tape-id's are available.

If no tape-id is entered, the default tape-id related to the selected domain is proposed.

#### Note:

The output file will be sent to the virtual reader and must be transferred to a CMS disk by the CMS RECEIVE command.

The program control parameters for the DATCON program parameter block will be generated by the SPEL/EC shell.

The DATCON program is an utility program described in "SPEL System: Technical Documentation Part1".



### 3.1.2. Optional data selection (FILTER)

Selection is done by the FILTER program.

The program panels allow control of the file names, data selection by regions, sub-regions, years and periodicity as well as the output file format. The output file format SDATA is recommended for optimum space economy. When SDATA format is selected an ASSIGN input file must have been created to define the assignments of original item codes to SPEL table element codes.

Up to eight ASSIGN files, may be used for selection and conversion of external data source codes to SPEL table element codes.

This program is a utility program described in "SPEL System: Technical Documentation Part1".

### 3.1.3. Data compilation (ORIGIN)

To import the standardized files of SPELIMP format (result of DATCON program) and/or SDATA format (e.g. results of FILTER program or manually edited) into the TABFILE work file, the ORIGIN program is used.

- For SPELIMP input files the original item codes are compiled into SPEL table element codes. An ASSIGN input file must have been created to define the assignments of original item codes to SPEL table element codes.
- For SDATA input files no code assignments are necessary because the data are already given in SPEL table element codes.
- A selection of input data is defined for each differentiation criterion (e.g. region, year).

The ORIGIN program can also be used to check the ASSIGN file for legal SPEL table element codes. In this mode the program does not access any data.

This program is a multi-purpose program and is described in "SPEL System: Technical Documentation Part 1".

## 3.2. Data set completion

The original data sources are incomplete. The work sector Data set completion fills in the missing data. It is divided into two work steps.

### Work step selection screen

```
Program starter ----- SPEL/EC ----- Work step selection

      EV   : Completion by subjective estimation
      TREND: Completion by trend estimation

Select an item by moving the cursor to any topic above

Enter= ok  1= Help  3= Quit
```

To fill in the missing data, first a subjective estimation is done by the EV program. Any further missing data are then filled in with values from the TREND trend estimation program.

Currently the control file of the EV program and the documentation listings of the TREND program outline the data completion rules and steps that have been followed in order to fill in missing data..

### 3.2.1. Completion by subjective estimation (EV)

The EV program allows data to be analysed in the econometrical sense based on regression equations with one or more independent variables. Simple arithmetical operations with time series (e.g. multiplication) are supported. The program is completely steered by a STEER control file.

Whenever regressions with more than one independent variable or arithmetical transformation are necessary to fill in missing data, EV is used. For each time series to be estimated, the control parameters (e.g. independent variables and data transformations) are subjectively determined by experts. They are collected as control statements in STEER control files.

The SPEL/EC shell offers a panel to specify the STEER control file.

**STEER file selection screen**

```
Evaluation program ----- SPEL/EC ----- STEER file selection
                                     Please enter STEER file name

STEER file      =>

Enter= ok  1= Help  3= Quit
```

**Parameters:****STEER file**

This STEER file contains the control statements for the EV program.  
It must be of STEER format.

The EV program is a utility program and is described in "SPEL System: Technical Documentation Part 1".

**3.2.2. Completion by trend estimation (TREND)**

Trend estimations for all remaining missing data are carried out by the TREND program by simple regressions using ordinary least squares (OLS). For each time series containing missing data, normally estimations with six different data transformations are calculated. The best fitting function is automatically selected. For this application of TREND, the coefficient of determination is used as the best fit criteria.

The original value time series, the estimated time series, the statistical figures, the type of selected function and the completed time series are printed. If the automatically selected function type is not acceptable for the user, he can explicitly define the transformation type and reference interval for each time series by creating a REGSTE steering file. This file may be input for the next run of TREND.

This program is a multi-purpose program and is described in "SPEL System: Technical Documentation Part 1".

Figure 4 : Work steps of Data preparation without additional data reduction step

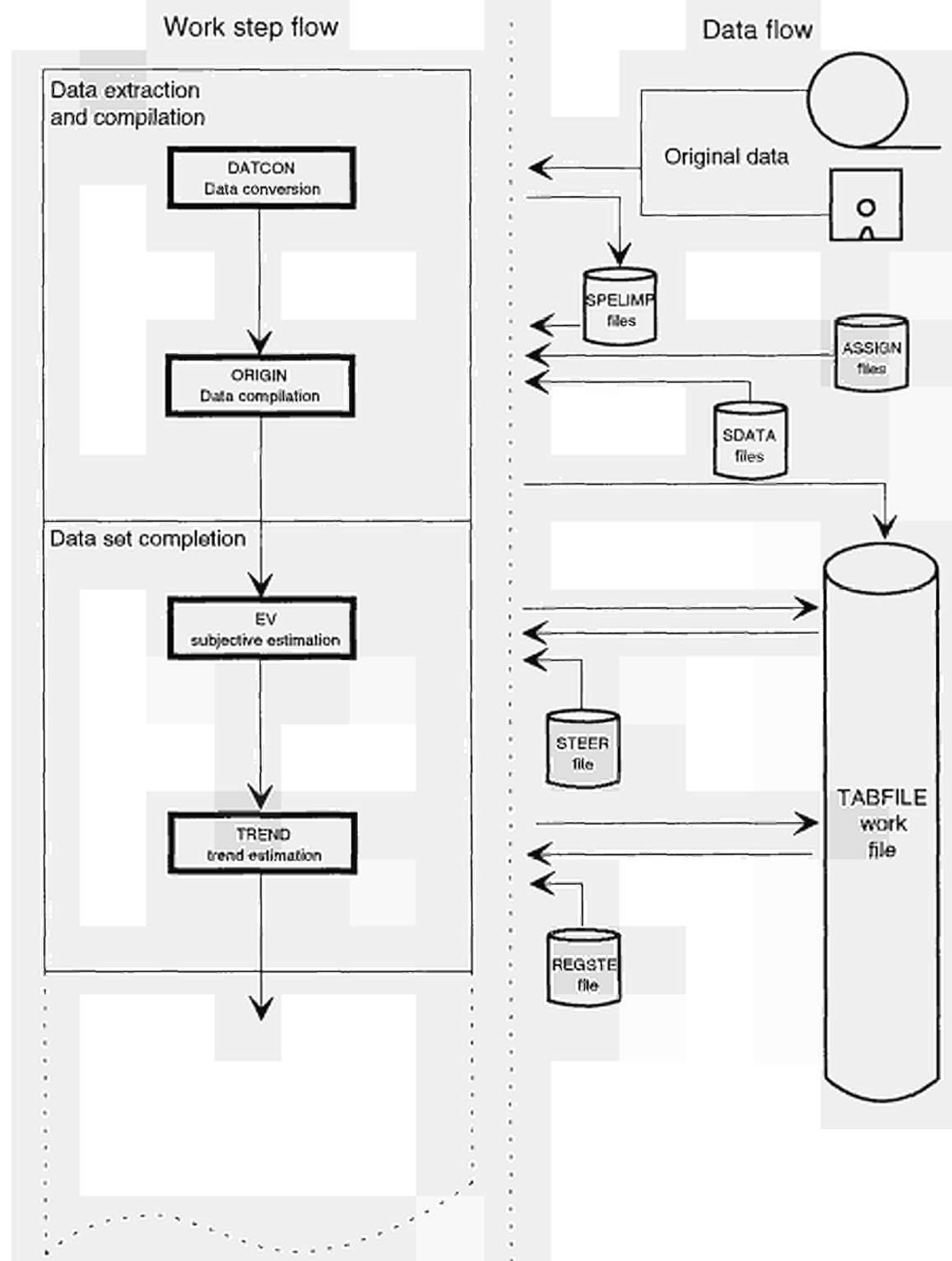
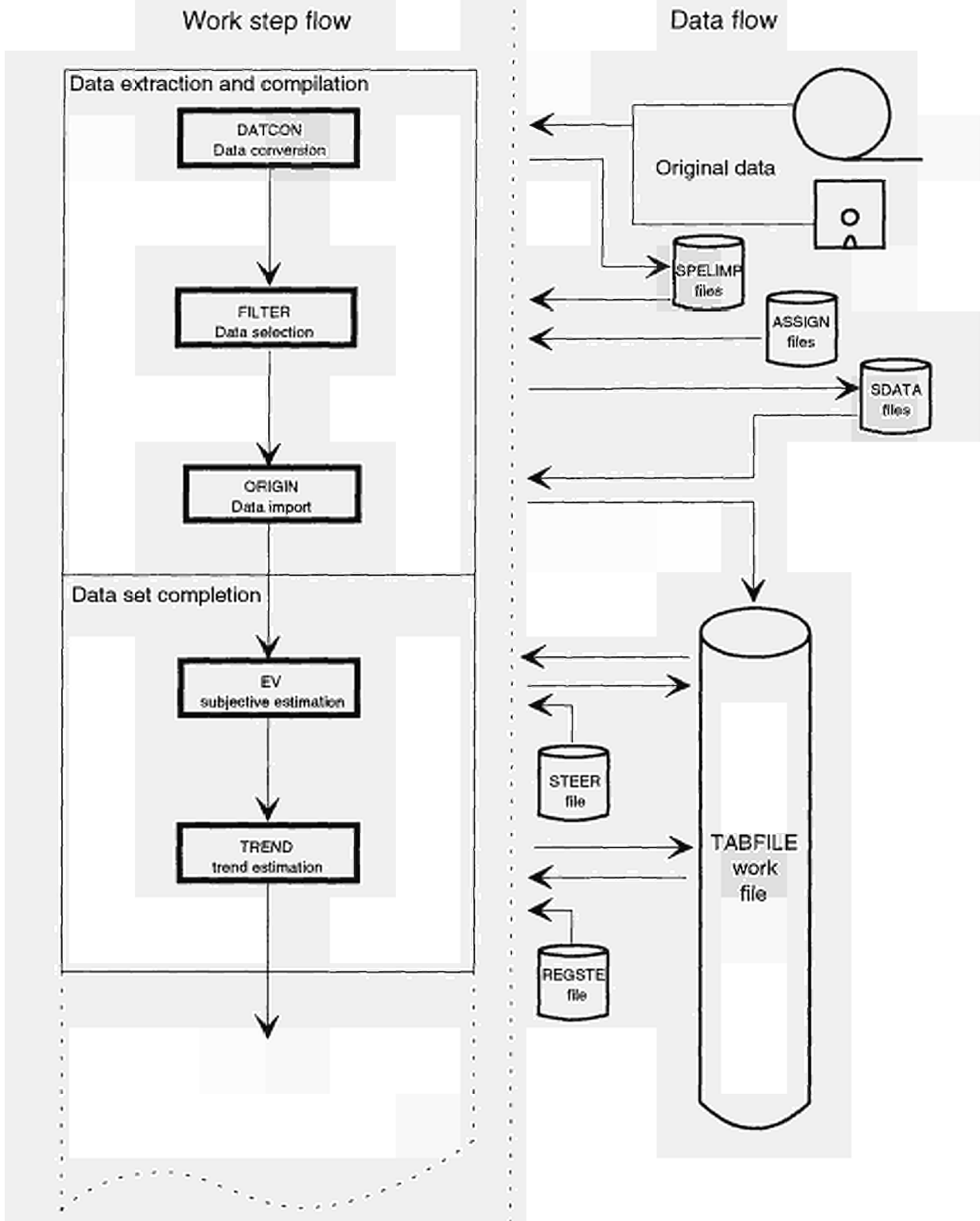


Figure 5 : Work steps of Data preparation with additional data reduction step



## 4. MODEL CALCULATIONS

The model calculations are divided into two work sectors:

- Base Model
- Short term forecast and simulation system

The Base Model establishes a consistent data base for the ex-post period from 1973 to the current year.

The result of the Short term forecast and simulation system are consistent data for one to three projection years based on the data of the ex-post period.

### Work sector selection screen

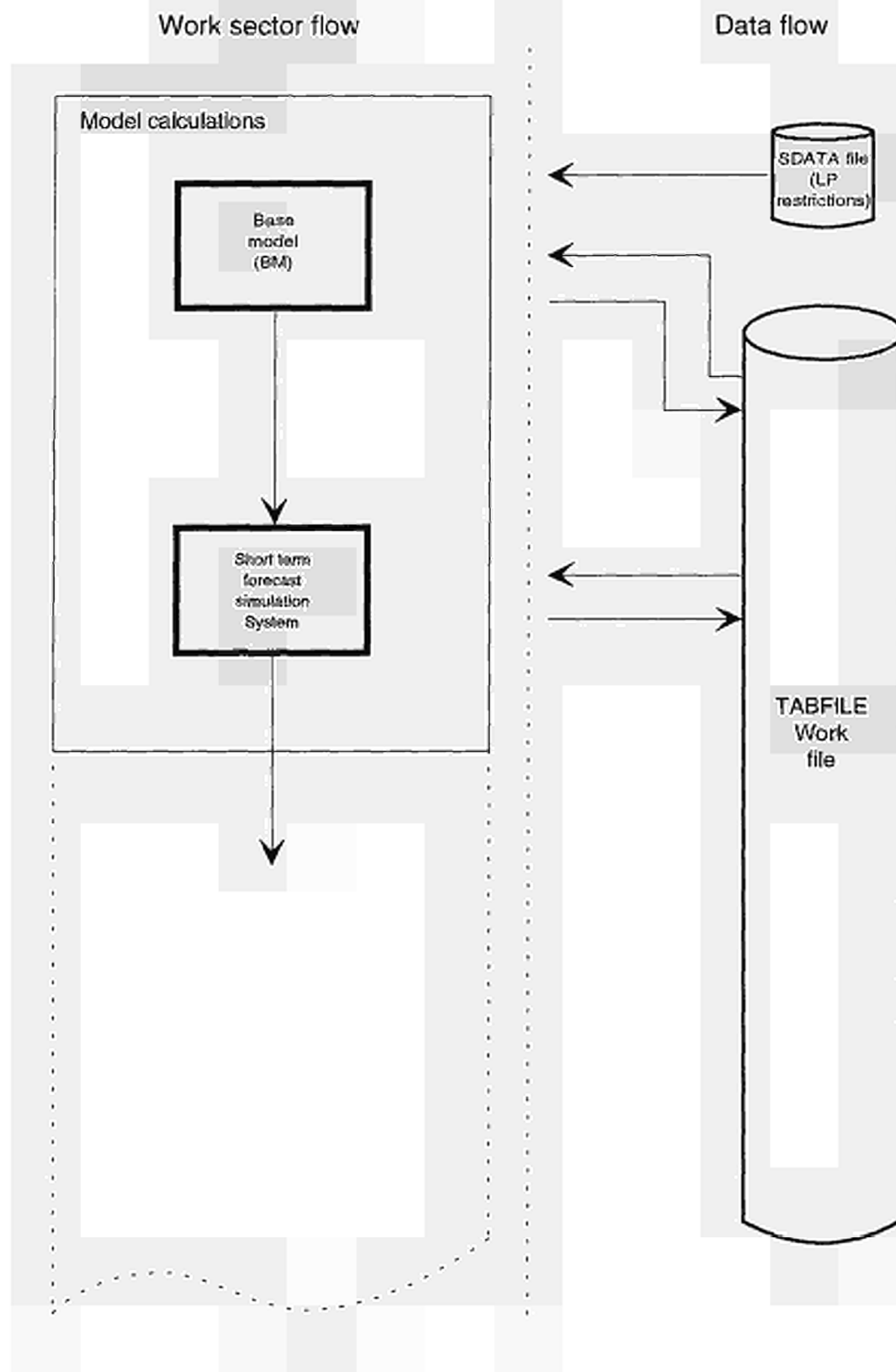
```
Program starter ----- SPEL/EC ----- Work sector selection

Base Model (BM)
Short term forecast simulation system (SFSS)

Select an item by moving the cursor to any topic above

Enter= ok 1= Help 3= Quit
```

Figure 6 : Work sectors of Model calculations



## 4.1. Base Model (BM)

After completing all the stages of the Data preparation component, a complete set of time series is stored in SPEL tables in the TABFILE work file. These tables contain a data set which represents the basis for establishing the consistent data base, called SPEL/EC-Data, for the ex-post period from 1973 to the current year. The SPEL/EC-Data consistency is calculated for each Member State and the EC as a whole by the Base Model. The methodological background is described in the methodological documentation "Introduction and Methodological Background (SPEL/EC Model)" in general terms and for the model in the methodological documentation "Base Model (SPEL/EC-BM)" for special numerical algorithms.

For further application select a work step.

### Work step selection screen

```
Program starter ----- SPEL/EC ----- Work steps selection

BASEMOD: Consistency at Member State level and EC aggregation level
AGGREG : Optional aggregation of user defined regions

Select an item by moving the cursor to any topic above

Enter= ok 1= Help 3= Quit
```

#### 4.1.1. Consistency at Member State level and EC aggregation level (BASEMOD)

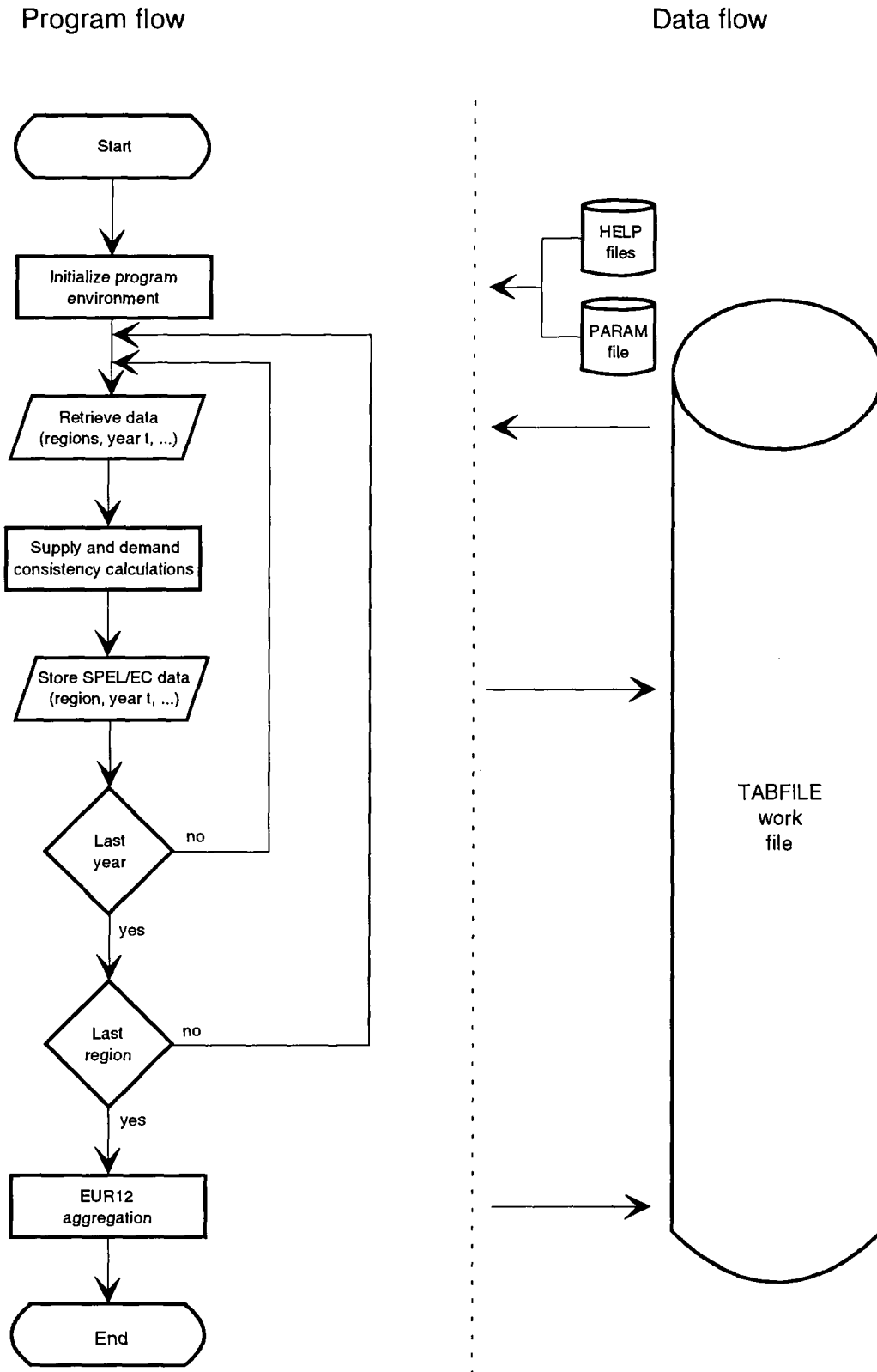
Consistency (model) calculations are carried out for SPEL/EC-Data both at Member State level and for the EC as a whole EUR12 (defined within SPEL as E12).

The BASEMOD program is divided into two main steps, which are :

- consistency at regional level
- aggregation to EC level.



Figure 7 : BASEMOD program flow chart



### Module startup screen

```
Base Model ----- SPEL/EC ----- Modul startup lcg0

      B A S E   M O D E L   S P E L / E C

Generation of the SPEL/EC-Data of table type BASB.
Base Model calculations are done for regions (Member States),
years and aggregates of regions.
For further informations see the methodological and technical
documentation.

Current settings for general files:
Protocol file      : BASEMOD   LOG      C
Print file        : BASEMOD   LISTING  C

Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF 12 key a special panel will be displayed which will allow changes to the names of general files for run time protocol (LOG-file) and documentation file (LISTING-file).

In the LOG-file the user will find all messages displayed on screen during a program run, including all error messages. More detailed information about working steps will be included too.

The LISTING-file contains more detailed documentation.

It is useful for the program user to check first the LOG-file after the program has ended, to see quickly if the calculations have been completed successfully.

#### Parameters:

None.

**Table file selection screen**

```
Base Model ----- SPEL/EC ----- Work file selection
                                     Please enter file names

Table file           => SPEL-BAS TABFILE F
Add.table file (input) =>
LP restriction file   => FEED-REL SDATA F

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

**Parameters:****Table file**

This file is a direct access file containing SPEL tables for entries and is of TABFILE (TABFILE work file) format.  
It is used for table input and table output and is opened for read/write access.

**Add.table file (input)**

Additional TABFILE input file only opened for read access. If specified, this file is then to be accessed for tables not found in the first defined table file.

**LP restriction file**

Input file of SDATA format containing restriction data for the feed component of the BASEMOD program.

**Table file key selection screen**

```

Base Model ----- SPEL/EC ----- Table file key selection

      Please make your key selection for output table file
      ATTENTION: - Subkeys for base year, type and model area are protected

Region      ( 3 Ch.) => F
Sub-region  ( 2 Ch.) => 00
Current year ( 2 Ch.) => 84 : 86
Periodicity ( 2 Ch.) => 00
Base year   ( 2 Ch.) => NN
Type        ( 4 Ch.) => BASB
Model area  ( 1 Ch.) => S D

      Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical expansion)
- a range defined by region codes such as "D - UK"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "D : B UK GR"
- use '\*' to select all codes as defined in the CODES file.

All codes and the range operands ":" and "-" must be separated by blanks.

The aggregate region "E12" (EUR12) is always implicitly calculated by aggregating the Member States. For regions not defined in the current region selection the BASEMOD results are got from previously calculated tables.

**Sub-region**

Sub-region selection. Specify one of the following:

- a sequence of sub-region codes to be selected such as "01 06 20"
- a range of codes defined by lower and upper codes such as "01 : 99"
- a combination of sequences and ranges such as "01 06 10 : 20".

All codes and the range operand ":" must be separated by blanks.

Sub-regions are currently not used.

The whole state has the region code "00". Always specify "00".

### Current year

Year selection. Specify one of the following:

- a sequence to be selected such as "70 75"
- a range defined by lower and upper years such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87".

All codes and the range operand ":" must be separated by blanks.

### Periodicity

Periodicity selection. Specify one of the following:

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2 Q3 Q4"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
(alphanumeric expansion)
- a range defined by periodicity codes such as "Q1 - Q4"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "01 : 06 Q4".
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

The whole year has the code "00". Other periodicity codes are currently not used.

### Base year

Protected subkey

For ex-post calculations the code for the base year is always "NN".

### Type

Protected subkey

For table output the type code "BASB" is always used with the BASEMOD program.

### Model area

Protected subkey

The calculations are always applied to supply ("S") and demand ("D").

### Type overlay definition screen

```
Base Model ----- SPEL/EC ----- Type overlay definition

Please make your definitions for
type overlay of complementary tables

Types SUPPLY => ZPAC COSC PRAC FBEC EXPC ESTC COMC
Types DEMAND => ZPAC EXPC ESTC COMC

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

#### Parameters :

##### Types SUPPLY

For complementary supply table input define an overlay hierarchy of table type codes (priority from left to right).

##### Types DEMAND

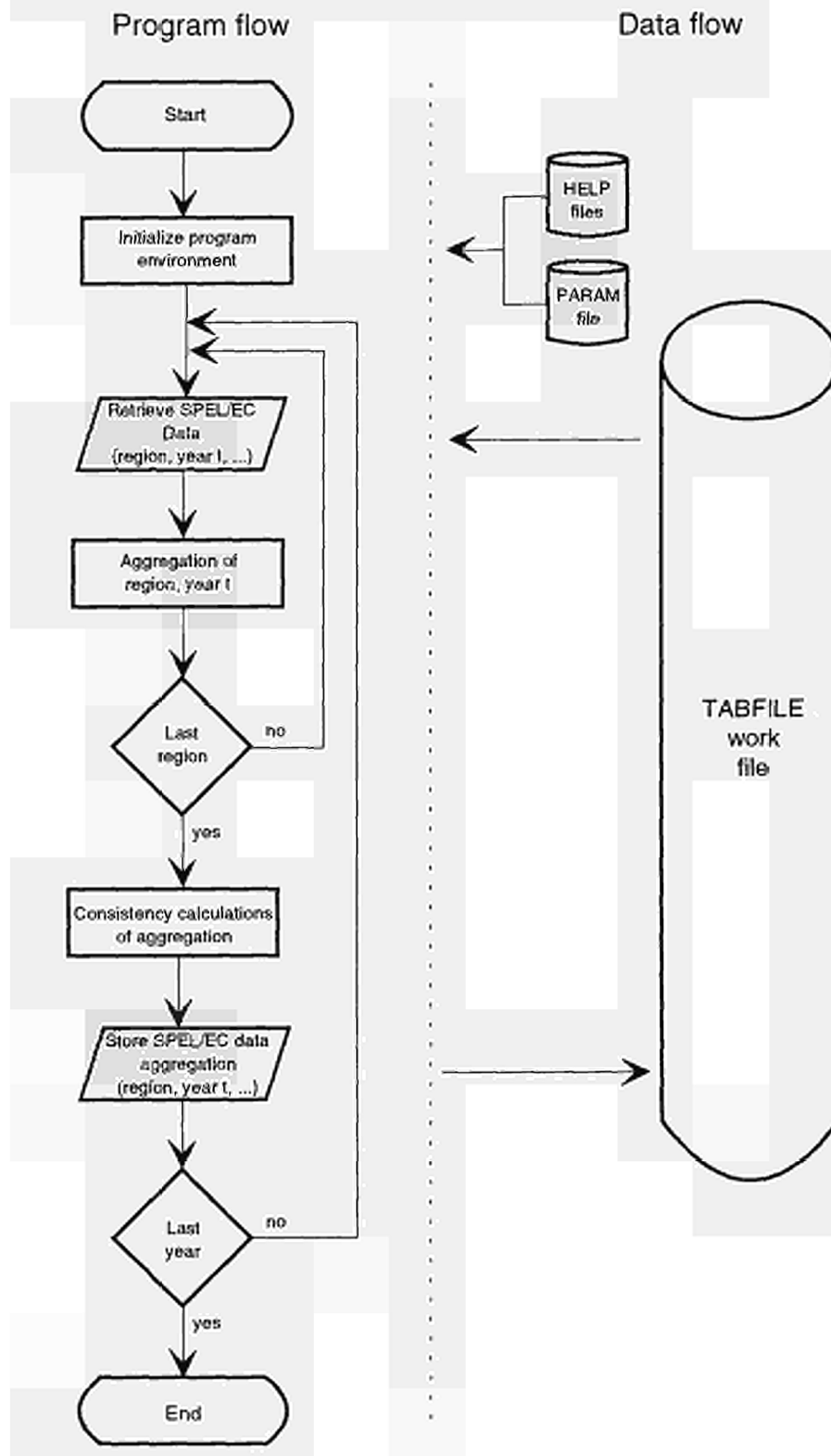
For complementary demand table input define an overlay hierarchy of table type codes.

### 4.1.2. Optional aggregation of user defined regions (AGGREG)

The aggregate region "E12" (EUR12) is implicitly calculated by the BASEMOD work step. To calculate additional aggregates defined by program parameters the AGGREG program can be used.

The input tables are results of the BASEMOD work step. For consistent calculation of the aggregates the algorithms of the BASEMOD program are used.

Figure 8 : AGGREG program flow chart



### Module startup screen

```
Region aggregation ----- SPEL/EC ----- Modul startup logo

          A G G R E G A T E   O F   S P E L   R E G I O N S

Aggregation of consistent SPEL table. The user can define
the set of regions to be aggregated.

          Current settings for general files:
          Protocol file      : SPEL LOG C

          Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF 12 key a special panel will be displayed which will allow changes to the names of general files for run time protocol (LOG-file) and the documentation file (PRINT-file).

It is useful for the program user to check the LOG-file first after the program has ended, to see quickly if the calculations have been completed successfully.

### Parameters:

None.

### Table file selection screen

```
Region aggregation ----- SPEL/EC ----- Work file selection

          Please enter file names

Table file           => SPEL-BAS TABFILE  H
Add.table file (input) => SPEL-BAS TABFILE  B

          Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```



**Parameters:****Table file**

This file is a direct access file containing SPEL tables for entries and is of TABFILE (TABFILE work file) format. It is used for table input and output.

**Add.table file (input)**

Additional TABFILE input file only opened for read access. This file is to be accessed for tables not found in the first defined table file

**Table key selection screen**

```

Region aggregation ----- SPEL/EC ----- Table key selection
                Please make your selection

Aggregate, region & sub-region ( 3 & 2 ch.) => E12 00
Current year                ( 2 ch.) => 71 : 92
Periodicity                 ( 2 ch.) => 00
Base year                   ( 2 ch.) => NN
Type                        ( 4 ch.) => BASB
Model area                  ( 1 ch.) => S

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Region aggregate, region & sub-region**

Select an aggregate code for the aggregate to be calculated. Each aggregate code consists of a region code (three characters) and a sub-region code (two characters) separated by blank.

The codes of the aggregate components for non standard aggregates (aggregates not defined in the CODES file) have to be defined in an additional panel.

**Current year**

Year selection. Specify one of the following:

- a sequence to be selected such as "70 75"
- a range defined by lower and upper bounds such as "70 : 75"
- a combination of sequences and ranges such as "70 75 80 : 87".

All codes and the range operand ":" must be separated by blanks.

### Periodicity

Periodicity selection. Specify one of the following:

- a sequence of periodicity codes to be selected such as "01 06" or "Q1 Q2 Q3 Q4"
- a range defined by lower and upper bounds such as "Q1 : Q4"  
--(alphanumerical expansion)
- a range defined by periodicity codes such as "Q1 - Q4"  
(expansion according to the definitions in the CODES file)
- a combination of sequences and ranges such as "01 : 06 Q4".
- use "\*" to select all codes as defined in the CODES file.

All codes and the range operands ":" and "-" must be separated by blanks.

The whole year has the code "00". Other periodicities are currently not used.

### Base year

Base year selection. The selection is limited to one base year.

For ex-post calculations the code for the base year is always "NN".

### Type

Type selection. The selection is limited to one type.

Specify a type such as "AGGB".

### Model area

Model area selection. The selection is limited to one model area.

Select one model area code such as "S".

For standard aggregates (aggregate is already defined in the CODES file) the aggregate components and the currency code are displayed to be checked.

### Check selection screen

```
Region aggregation ----- SPEL/EC ----- Check selection
      Please check currency and components for aggregate E12 00:
Currency:      ECU
Components:    B  00      DK  00      D  00      GR  00      E  00
               F  00      IRL 00      I  00      L  00      NL  00
               P  00      UK  00

      Select an action by moving the cursor to any topic below

                Continue aggregation
                Revise aggregate selection

      Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

### Parameters:

#### Continue aggregation

Program execution continues doing the aggregation for the displayed aggregate components

#### Revise aggregate selection

Program returns to the "Table key selection" panel.

If the aggregate is not standard, the aggregate components and the currency have to be defined in the following panel.

### Aggregate component definition screen

```

Region aggregation ----- SPEL/EC ----- Aggregate definition
      Please enter currency and components for aggregate EP 00:
Aggregate: EP 00, currency      => ECU
Aggregate: EP 00, component: 1 => E 00
Aggregate: EP 00, component: 2 => P 00
Aggregate: EP 00, component: 3 =>
Aggregate: EP 00, component: 4 =>
Aggregate: EP 00, component: 5 =>
Aggregate: EP 00, component: 6 =>
Aggregate: EP 00, component: 7 =>
Aggregate: EP 00, component: 8 =>
Aggregate: EP 00, component: 9 =>
Aggregate: EP 00, component: 10 =>
Aggregate: EP 00, component: 11 =>
Aggregate: EP 00, component: 12 =>
Aggregate: EP 00, component: 13 =>
Aggregate: EP 00, component: 14 =>
Aggregate: EP 00, component: 15 =>
Aggregate: EP 00, component: 16 =>
Aggregate: EP 00, component: 17 =>
      Enter= ok  l= Help  3= Quit  4= Exit  11= Save/load

```

### Parameters:

#### Aggregate *rrr ss*, currency

Specify currency code (three characters) of aggregate *rrr ss*, where *rrr* is the region code and *ss* is the sub-region code of the aggregate.

Currency codes permitted are: ECU, BFR, DKR, DM, DR, PTA, FF, IRL, LIT, LFR, HFL, ESC, UKL and US\$

#### Aggregate *rrr ss*, component: *n*

Specify region (three characters) and sub-region code (two characters) of *n*'th component of aggregate *rrr ss*, where *rrr* is the region code and *ss* is the sub-region code of the aggregate.

## 4.2. Short term forecast and simulation system (SFSS)

The SPEL/EC-Data forms the base from which simulations may be carried out. Short term forecast and simulations are calculated based on consistent SPEL tables of basic structure (type: BASB). For projections a "base year" must be specified, normally the last year for which the consistent SPEL/EC-Data is available. The simulation results will be consistent SPEL tables of basic structure, too (type: SFSSB).

Inside the tables, SFSS differentiates between endogenous and exogenous variables. Exogenous variables are those for which growth rates related to the base year can be specified outside the

system. Endogenous variables are calculated by the simulation system using the consistency algorithms like those in the Base Model.

For growth rates of exogenous variables "experts" are responsible. To make it easier to specify the growth rates, a first work step (SFPROP) computes system proposals for all exogenous variables. The system proposals are stored as growth rates in tables of basic structure. (type: SFPB)

These proposals may be revised by expert's judgement. The expert's data are stored as growth rates in files of SDATA format (Type: SFEB).

The projection work is done in the second step (SFS). The exogenous variables are extrapolated by growth rates. Growth rates from expert's judgement have priority over system proposals.

The definitions of endogenous and exogenous variables and the methodological background of SFSS is described in the methodological documentation "SPEL System: Methodological Documentation of the SPEL/EC Model Part 1" in general terms and for the model in the methodological documentation "SPEL System: Methodological Documentation of the SPEL/EC Model Part 3" for special numerical algorithms.

The SPEL/EC shell offers the work step selection as follows:

#### Work step selection screen

```
Program starter ----- SPEL/EC ----- Work steps selection

SFPROP : System proposals for exogenous variables
SFS     : Forecast and simulation
AGGREG  : Optional aggregation of user defined regions

Select an item by moving the cursor to any topic above

Enter= ok  1= Help  3= Quit
```

#### 4.2.1. System proposals for exogenous variables (SFPROP)

System proposals for each exogenous variable are computed from

- specified indicator time series which must extend through the projection period
- six alternative trend estimations for the original time series as outlined below.

On default for the original time series, six OLS estimations are calculated with different data transformations for the dependent variable and the trend. The data transformations are:

	dependent	trend
1.	linear	linear
2.	linear	inverse
3.	logarithmic	inverse
4.	logarithmic	logarithmic
5.	linear	logarithmic
6.	logarithmic	linear

The computational process is in three steps:

- The system computes for each indicator variable a time series of growth factors from the indicator's observations.
- For each alternative trend estimation, a time series of growth factors each from the original values in year  $t$  to the trend values in year  $t+(\text{projection period})$  are computed. The projection period may be one, two or three years, so that the computed growth reflects the projection period.
- All of the computed growth factors series are compared with growth factor series of the original data. The system proposal will be taken for the best fitting one. The fitting criteria may be either the coefficient of determination or the Theil's coefficient.

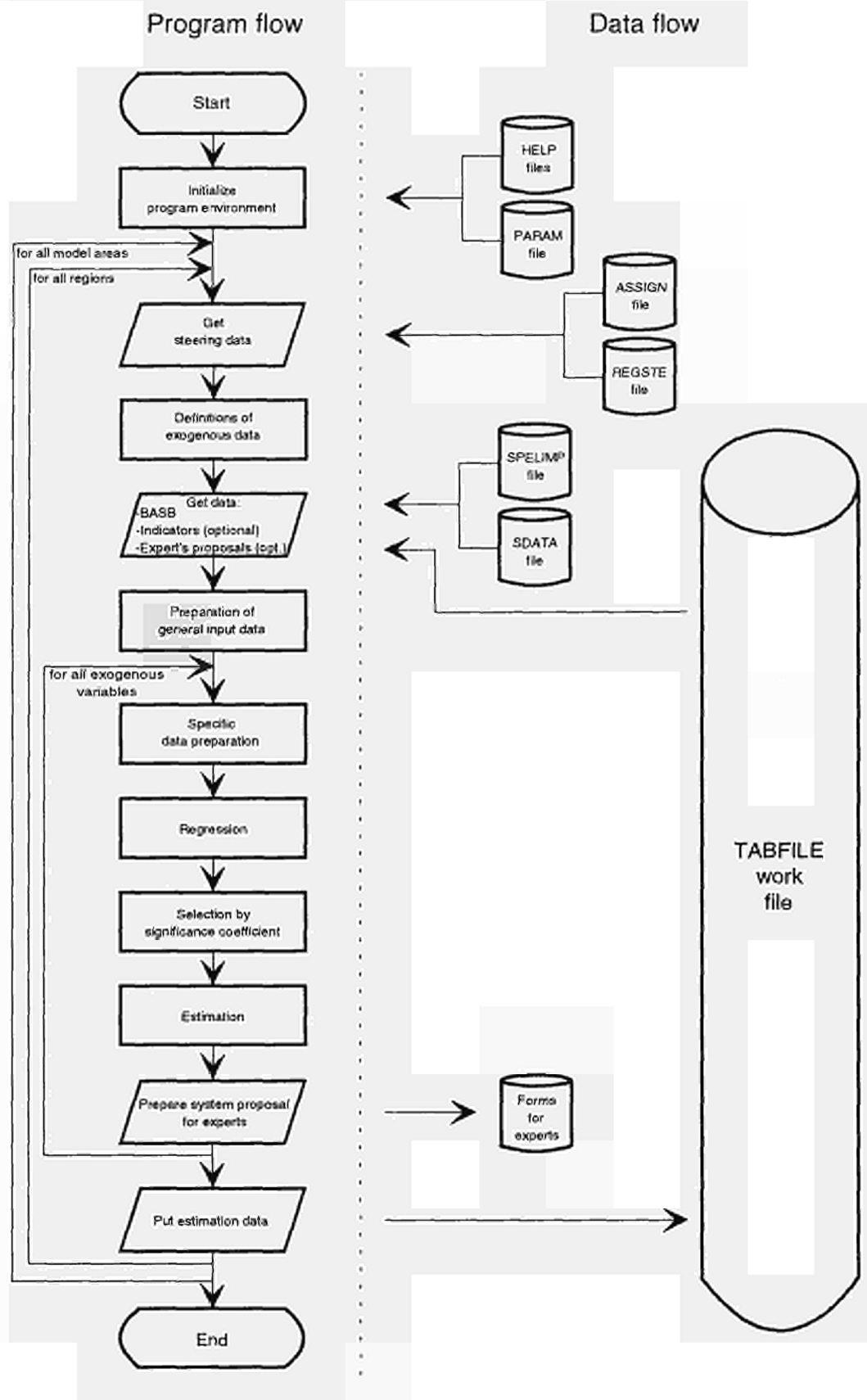
When an indicator time series is selected, for each projection year the growth rate of the indicator (base year to projection year) will be stored as the system proposal.

When one of the alternative trend estimations is selected, for each projection year the growth rate from base year's original data to the trend estimation of the projection year will be stored as the system proposal.

The indicator and the regression parameter may be modified by a REGSTE steering file.

The system proposals and some of the ex-post period data (SPEL/EC-Data) are prepared for expert's checking. Forms are offered to be updated by expert's judgement. These expert's proposals are imported into the SPEL system using the SDATA standard format.

Figure 9 : SFPROP program flow chart



### Module startup screen

```
SFSS system proposal ----- SPEL ----- Module startup logo

SYSTEM PROPOSALS FOR EXOGENOUS VARIABLES

System proposals for exogenous variables are computed from specified
indicator time series and trend values for original time series.

Current settings for general files:
Run time protocol : SPEL LOG C
Print file       : SPEL LISTING C

Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key a special panel will be displayed which will allow changes to names of general files for run time protocol (LOG-file) and documentation file (LISTING-file).

In the LOG-file the user will find all messages displayed on screen during a program run including error messages. More detailed information about working steps will be included too.

The detailed protocol for all calculations is stored in the LISTING-file

It is useful for the program user to check first the LOG-file after a program has ended to get a rapid overview of the success of estimations.

#### Parameters:

None.



**Work file selection screen**

```

SFSS system proposal ----- SPEL ----- Work file selection
                                     Please enter file names

Table file                          => SPEL-USR TABFILE H
Add.table file (input)              => SPEL-SYS TABFILE F
Regr. steering file: 1              => SFPROP REGSTE H
Regr. steering file: 2              =>
Assign file: 1                      => SFPROP ASSIGN H
Assign file: 2                      =>
Indicator file: 1                   => SFZPA1 SPELIMP A
Indicator file: 2                   => SFPRAG SPELIMP A
Indicator file: 3                   =>

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Table file**

TABFILE work file containing the consistent SPEL tables of the SPEL/EC-Data. The file will be used for input and output.

**Add. table file (input)**

Additional TABFILE input file only opened for read access. If specified, this file is then to be accessed for tables not found in the first defined table file.

**Regr. steering file: 1**

If specified, this REGSTE file contains explicit definitions for the trend estimations of the exogenous SPEL/EC-Data.

For this application the REGSTE format file includes extensions to the standard file format. The extensions are described in chapter "Extensions to steering file formats for simulation systems". The reference intervals are defined in the the panel "Regression steering" which follows.

**Regr. steering file: 2**

Additional REGSTE file, if specified

**Assign file: 1**

If specified, this ASSIGN file contains the assignments of indicator variables codes (domain codes, e.g. ZPA11234567) to SPEL table element codes.

For this application the ASSIGN format file includes extensions to the standard file format. The extensions are described in chapter "Extensions to steering file formats for simulation systems".

**Assign file: 2**

Additional ASSIGN file, if specified.

**Indicator file: 1**

If specified, this SPELIMP file contains external source data for indicator time series. The time series defined in the assign files are checked.

**Indicator file: 2**

Additional SPELIMP file, if specified.

**Indicator file: 3**

Additional SPELIMP file, if specified.

**Table key selection screen**

```
SFSS system proposal----- SPEL ----- Table key selection
      Please make your selection for table key

Region          ( 3 ch.) => D F
Sub-region      ( 2 ch.) => 00
Projection year ( 2 ch.) => 92 93
Periodicity     ( 2 ch.) => 00
Base year       ( 2 ch.) => 91
Table output type ( 4 ch.) => SFPB
Model area      ( 1 ch.) => S D

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical expansion)
- a range defined by region codes such as "D - UK"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

**Sub-region**

Sub-regions are currently not used.

The whole region has the sub-region code "00".

Always specify "00".

**Projection year**

Projection year selection. Specify one of the following:

- a sequence of years to be selected such as "93 94"
- a range defined by lower and upper years such as "92 : 94"
- a combination of sequences and ranges such as "92 93 : 94"

All codes and the range operand ":" must be separated by blanks.

**Periodicity**

Periodicities are currently not used.

The whole year has the periodicity code "00".

Always specify "00".

**Base year**

Base year of the projection.

Select one base year such as "91"

**Table output type**

Protected subkey

For table output the type code "SFPB" is always used with the SFPROP program.

**Model area**

Protected subkey.

The calculations are always applied to supply ("S") and demand ("D").

**Regression steering screen**

```

SFSS system proposal----- SPEL ----- Reference and output

      Please select parameters for regression steering

Reference interval          (start : end1 : end2) => 72 : 85 : 90
Significance coefficient    (THEIL/DETERMINATION) => DETERMINATION
Deflator                   (GDP/PPS/NONE) => NONE
Documentation level        (FULL/ROUGH/MINIMUM/NONE) => MINIMUM
Table output level         (PROPOSAL/TREND ONLY/NONE) => PROPOSAL
Output to experts level    (PRINT/CSV/GCSV/NONE) => PRINT

      Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load

```

**Parameters:****Reference interval**

Define the years for short and long reference intervals.

Syntax:

start : end1 : end2

end1 : end2 will be the short interval

start : end2 will be the long interval

For the short interval you must specify at least 4 years

start and end2 may be the same year.

The start year has to precede the end years.

Example: 72 : 85 : 90 or 80 : 80 : 90

The reference interval may be redefined for specific exogenous variables in the REGSTE steering file.



### Output to experts level

For output to the expert's judgement you can choose between three levels:

- PRINT:** the system proposal are prepared for printed forms
- CSV:** the system proposals are stored in a file of CSV format.  
This format is used to transfer data to PC standard software like EXCEL or SYMPHONY
- GCSV:** the same format as CSV format, but ";" is used as column separation.
- NONE:** nothing

#### **4.2.2. Forecast simulation (SFS)**

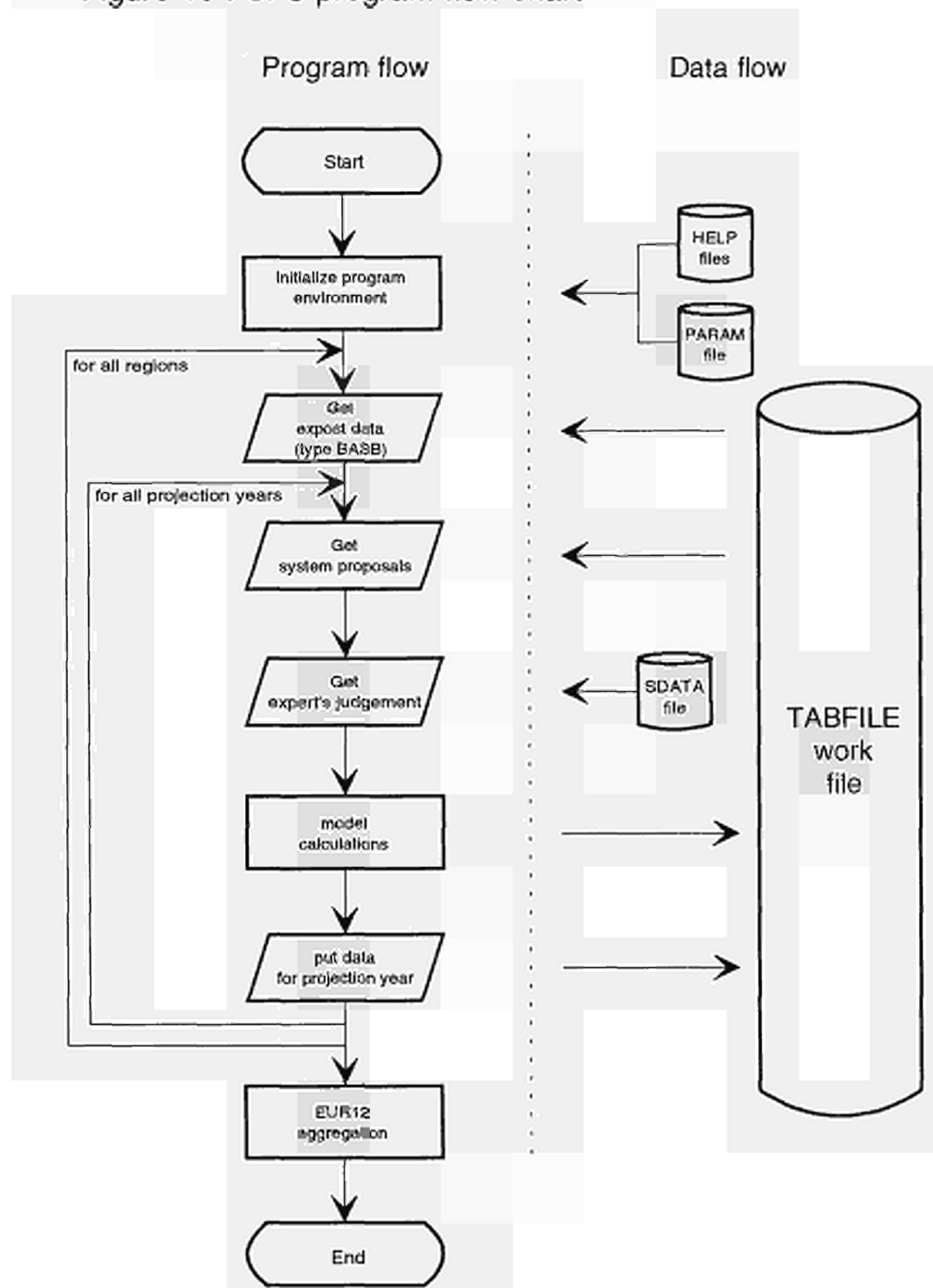
Short term forecast and simulation is done by the SFS program. For the extrapolation of exogenous variables the growth rates from expert's judgements, if available, otherwise the system proposals (SFPROP) are used.

Endogenous variables are calculated by the SFS program using the consistency algorithms like those used in the Base Model.

After the simulation or forecast work is finished for all regions, the EUR12 aggregation is done in the same way as for the Base Model.

The results of the SFS program are stored in the TABFILE work file.

Figure 10 : SFS program flow chart





**Module startup screen**

```
Short term forecast ----- SPEL ----- Module startup logo

      S H O R T - T E R M   F O R E C A S T   A N D   S I M U L A T I O N

      Extrapolation of exogenous variables by growth rates and
      calculation of endogenous variables by consistency algorithms

      Current settings for general files:
      Run time protocol : SPEL LOG C

      Enter= ok  1= Help  3= Quit  12= Gener.file
```

This screen is for information only. By pressing the PF12 key a special panel will be displayed which will allow changes to names of general files for run time protocol (LOG-file) and documentation file (LISTING-file).

In the LOG-file the user will find all messages displayed on screen during a program run including error messages. More detailed information about working steps will be included too.

It is useful for the program user to check first the LOG-file after a program has ended to get a rapid overview of the success of estimations.

**Parameters:**

None.

### Work file selection screen

```
Short-term forecast ----- SPEL ----- Work file selection
                               Please enter file names

Table file                      => SPEL-SIM TABFILE H
Add.table file (input) => SPEL-BAS TABFILE F
Expert's judgem. file => EXPERT SDATA H

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

### Parameters:

#### Table file

TABFILE work file containing the consistent SPEL tables of the SPEL/EC-Data, the system proposals.

#### Add. table file (input)

Additional TABFILE input file. If specified, this file is then to be accessed for tables not found in the first defined table file.

#### Expert's judgm. file

This SDATA file contains the growth rates for exogenous variables coming from expert's judgement (table type: SFEB). The growth rates must be average annual growth rates in percent from the base year's data to the projection year's data.

**Table key selection screen**

```

Short-term forecast----- SPEL -----Table key selection
                                Please make your table key selection

Region          ( 3 ch.) => D F
Sub-region      ( 2 ch.) => 00
Projection year ( 2 ch.) => 92 : 94
Periodicity     ( 2 ch.) => 00
Base year       ( 2 ch.) => 85
Output type     ( 4 ch.) => SFSB
Model area      ( 1 ch.) => S D

                                Enter= ok  1= Help  3= Quit  4= Exit 11= Save/load

```

**Parameters:****Region**

Region selection. Specify one of the following:

- a sequence of region codes to be selected such as "D F NL"
- a range defined by lower and upper bounds such as "AAA : ZZZ"  
(alphanumerical expansion)
- a range defined by region codes such as "D - UK"  
(expansion according to definitions in the CODES file)
- a combination of sequences and ranges such as "D : F UK GR"
- use '\*' to select all codes as defined in the CODES file

All codes and the range operands ":" and "-" must be separated by blanks.

**Sub-region**

Sub-regions are currently not used.

The whole region has the sub-region code "00".

Always specify "00".

**Projection year**

Projection year selection. Specify one of the following:

- a sequence of years to be selected such as "92 : 94"
- a range defined by lower and upper years such as "92 93"
- a combination of sequences and ranges such as "92 : 93 94"

All codes and the range operand ":" must be separated by blanks.

The maximum number of years to be selected is 3.

#### Periodicity

Periodicities are currently not used.

The whole year has the periodicity code "00".

Always specify "00".

#### Base year

Base year selection. The selection is limited to one base year.

Select one base year such as "85".

#### Output type

Type selection. The selection is limited to one type.

Select one type such as "SFBS".

#### Model area

Protected subkey.

The calculations are always applied to supply ("S") and demand ("D").

### **4.2.3. Optional aggregation of user defined regions (AGGREG )**

The aggregate region "E12" (EUR12) is implicitly calculated by the SFS work step. To calculate additional aggregates defined by program parameters the AGGREG program can be used. The input tables have to be consistent; that means they should be results of the SFS work step.

The parameters of the AGGREG program are described in the Base Model (BM) work sector documentation.

## **4.3. Extensions to steering file formats for simulation systems**

For the Short term forecast and simulation system (SFSS) the following extensions are added to the standard steering file formats REGSTE and ASSIGN.

#### REGSTE format:

The REGSTE file structure as described in "SPEL System: Technical Documentation Part 1", is defined for the regression steering records from position 1 to 33. For the SFSS two additional informations are necessary.

---

pos. 34-41 cccrrrr deflator or other divisor defined by a SPEL table element code or an abbreviation as follows:  
 GDP: price index of gross domestic product  
 PPS: index Purchasing Power Standard  
 NONE: no deflator (nominal price development)

pos. 42 blank

pos 43-50 cccrrrr SPEL table element code of a independent variable for non trend estimations

**ASSIGN format:**

For SFSS multiple assignments of external data source codes to a single SPEL table element code are allowed to define alternate indicator time series.

**Example of an ASSIGN file including SFSS extensions:**

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
B SWHESWHE = ZPA11111012;
B SWHESWHE = ZPA13110012;
B SWHESWHE = ZPA13110010;
*
B 'RYE RYE ' = ZPA11112112;
B 'RYE RYE ' = ZPA13114012;
B 'RYE RYE ' = ZPA13114010;
*
B OATSOATS = ZPA11112212;
B OATSOATS = ZPA13116012;
B OATSOATS = ZPA13116010;
*
* .....
* .....
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

## 5. UTILITIES

The utilities can be independently used from the model components. The selection is done by the following screen.

### Utilities selection screen

```
Program starter ----- SPEL ----- Utilities selection

DASERV : Maintenance of TABFILE files
FILTER : Selection and Conversion of SPELIMP and SDATA files
EV      : General estimation and evaluation program
DAOUT   : SPEL data base output

Link of the SPEL disks
File list of the output disk
CMS command level (type RETURN when finished)

Select an item by moving the cursor to any topic above

Enter= ok  1= Help  3= Quit
```

### 5.1. Maintenance of TABFILE files (DASERV)

The file format of the TABFILE work files cannot be managed by conventional editors.

The DASERV program offers many services for TABFILE work file maintenance:

- the creation and organization of new files
- the reorganization of existing files
- the listing and printing of data

For further detailed information on the services of the DASERV program see "SPEL System: Technical Documentation Part 1".

## 5.2. Selection and conversion of SPELIMP and SDATA files (FILTER)

The FILTER program works on sequential files of SPELIMP and SDATA format

It offers:

- to reduce the amount of data in SPELIMP or SDATA files by item selection
- to convert codes (e.g. original domain codes to SPEL table element codes)
- to convert file format, SPELIMP to SDATA or vice versa.

All three facilities can be combined. For the code conversion a set of ASSIGN files must be defined.

For further detailed informations on the FILTER program see "SPEL System: Technical Documentation Part 1".

## 5.3. General estimation and evaluation program (EV)

The EV program offers general evaluations and single equation regression estimations based on the standard file formats TABFILE and SDATA. It may be used for ad-hoc exploitations.

Selecting this utility the SPEL/EC shell will display a panel to specify the STEER control file.

### STEER file selection screen

```
Evaluation program ----- SPEL/EC ----- STEER file selection
                                     Please enter STEER file name

STEER file      => TEST  STEER  A

Enter= ok  1= Help  3= Quit
```

**Parameters:**

**STEER file**

The STEER file contains the control statements for the EV program.  
It must be of STEER format.

If no STEER file name is entered, the EV program will start in the dialog mode to enter the commands directly from the keyboard and to execute the operations immediately.  
Use \$STOP to leave the dialog mode.

For further detailed informations on the EV program see "SPEL System: Technical Documentation Part 1".

## **5.4. SPEL data base output (DAOUT)**

Selections of data from TABFILE files can be printed and exported by the DAOUT utility. The data selection is done by using a list of selection criteria, referring to the identifiers for the SPEL data dimensions (region, sub-region, year, periodicity, base year, table type, table column and table row). There are four different output modes:

- printing file
- SDATA file
- file of CSV format for import to PC standard software, e.g. EXCEL or SYMPHONY
- file of German CSV format (including ";" as column separator)

The data dimensions can be transposed for output, except for SDATA output mode..

For further detailed information on the DAPRINT program see "SPEL System: Technical Documentation Part 1".



## 5.5. Link of the SPEL disks

To check mode and access of the operator/user's disks the SPEL/EC shell offers the following panel:

### Disk link screen (e.g. for users)

```

Program starter ----- SPEL/EC ----- Link of the SPEL disks

          Disks currently accessed and their access mode

C (Output/work files)      => R/W
F (SPEL/EC system files)  => R/O
I (SPEL system files)     => R/O
E (SPEL operator files)   => R/O
H (SPEL/EC simulation data) => R/O
J (SPEL/EC Base Model data) => R/O
L (SPEL/EC external data) => R/O

Enter R/W for read and write access and R/O for read only access

Enter= ok  1= Help  3= Quit

```

### Parameters for users and operators:

#### C (Output/work files)

The C disk is a temporary disk containing program output files or program LOG-files.

Both operators and users have access to this disk. It is linked for R/W access.

#### F (SPEL/EC system files)

The F disk is the SPEL/EC system disk providing the executable SPEL programs.

Both operators and users have access to this disk. It is linked for R/O access as default.

This disk can be linked in R/W mode by operators only.

#### I (SPEL system files)

The I disk is the SPEL system disk providing general programs.

Both operators and users have access to this disk. It is linked for R/O access as default.

This disk can be linked in R/W mode by operators only.

### Parameters for operators only:

#### E (SPEL operator files)

The E disk is a special disk for the operator.

Its access is restricted to the operator. It is normally linked for R/O access as default.

The disk can be linked in R/W mode.

#### H (SPEL/EC simulation data)

The H disk is the SPEL/EC disk for model simulation results.

Its access is restricted to the operator. It is normally linked for R/O access as default

The disk can be linked in R/W mode.

#### J (SPEL/EC Base Model data)

The J disk is the SPEL/EC disk for the intermediate and final Base Model data (SPEL/EC-Data).

Its access is restricted to the operator. It is normally linked for R/O access as default.

The disk can be linked in R/W mode.

#### L (SPEL/EC external data)

The L disk is the SPEL/EC disk for exogenous data mostly stored in SPELIMP files.

Its access is restricted to the operator. It is normally linked for R/O access as default.

The disk can be linked in R/W mode.

## 5.6. File list of the output disk

Selecting this item the CMS FILELIST command is invoked for the C disk. A list of all existing files on this disk will be displayed.

To come back to the SPEL/EC shell the user has to quit the FILELIST (enter QUIT or press PF3).

## 5.7. CMS command level

Selecting this item the SPEL/EC shell will be left for temporary use of the CMS dialog system. In this level all CMS facilities are available. The user just has to type "RETURN" to come back to the SPEL/EC shell.

This item may be selected to check data files or to call the system editor

## 6. EXPLOITATION

The EXPLOIT program offers exploitation results for ex-post and ex-ante data in a set of fixed structured tables. The exploitation details and listings may be modified to reflect current political questions.

### Module startup screen

```

Exploitation ----- SPEL/EC ----- Modul startup logo

                E X P L O I T A T I O N

Use this module to generate preformatted preset tabular
listings .

To utilise you must ensure that SPEL-EC Data tables and/or
a forecast simulation have been prepared.

Current settings for general files:
Protocol file      : SPEL LOG C

Enter= ok  1= Help  3= Quit  12= Gener.file

```

Tablular listings may be prepared for the Agriculture Income Situation.

The components of Agriculture Income Situation which are available can be grouped as follows:

- Whole Sector - which gives detailed information measured as value added, of each Member State and EC as a whole
- Differentiated by Branch generation - Time series showing the annual development of output and input use elements!
- Economic Accounts - Development of volume and prices using EAA accounting definitions, comparing year t to the preceding year.

### Group selection screen

```
Exploitation ----- SPEL/EC ----- Group selection

Components of Agricultural Income Situation

Whole Sector
Differentiated by Branch
Economic Accounts

Select an item by moving the cursor to any topic above

Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

The information on the current version may be obtained from the SPEL team Luxembourg.

## **7. ANNEX A: ERROR MESSAGES AND WARNINGS**

The utilization of the SPEL/EC Model is controlled by a series of panels within the overall structure. Protocol panels are also displayed. For all these panels an on-line help system is available. This means that the operator/user can get on line help information for all panel fields and for all errors and warnings. For further detailed information on the help system see "SPEL System: Technical Documentation Part 1".

Annex A is a list of all error messages and warnings.

The beginning of each message is marked by "-" followed by the message text. If the message is directly related to a program or the SPEL/EC shell, the name appears in brackets following the message text. All other messages may appear in several programs as well as in the SPEL/EC shell.



- -Assignment file missing (ORIGIN)  
You have specified at least one input file of SPELIMP format.  
An additional assignment file of ASSIGN format has to be allocated.  
Please specify an ASSIGN file.
- -Block name missing  
No block name defined.  
The input argument is totally blank.
- -Block name too long  
One of the block names exceeds the maximum length of 20 characters.
- -Compare entry not found (DASERV)  
For an entry in the reference selection the specified compare entry was not found.  
Remember that the selected service was executed incomplete.  
You can use the 'List file directory' service to determine the missing entry.
- -Components missing (AGGREG)  
You must define 1 (minimum) up to 17 (maximum) components.  
component 1 ... component 17: select a region code (3 ch.) and a sub-region code (2 ch.) separated by blanks such as "D 00".
- -Conflicting min./max. (DASERV)  
Your choice for the estimated average number of data elements conflicts with your choice for the minimum or maximum number of data elements.
- -Device not available  
A device may be a whole disk or part of a disk.  
The device specified does not exist in your current operating system.  
Please specify the name of an existing device.
- -Device read only  
A device may be a whole disk or part of a disk.  
The specified device is write protected, which means you

may only read from it.

-

-Directory space exceeded

The new entries to be created require more directory space than available in all currently open TABFILE files.

Remember that the selected service was executed incompletely.

You can use the 'List file directory' service of the program DASERV to find out what work is already done.

The 'List file status' service shows you the amount of 'deleted' directory space which can be made reusable by the 'Free unused space' service. Remember that the directory space displayed refers to all currently open TABFILE files. In case of copy operations from one file into another the required space is the sum of the space for both files.

If the 'Free unused space' service does not provide sufficient directory space, check whether it is possible to reduce the number of simultaneously open TABFILE files.

-

-Directory wrong

The internal file directory has been damaged by a bad program. Please restore an actual backup copy of the affected file.

In emergencies you can use the 'Free unused space' service of the program DASERV to remove any inconsistencies found, but then all affected entries will be lost. The result is always a formal correct file, but there is no guarantee that the consistency check will detect all affected data.

-

-Display text

The panel displays a text.

If the text is longer or some lines have more than 72 characters please use the PF-keys to page forward, backward, to the right or to the left.

-

-Duplicates not allowed

For the specifications of keys and subkeys duplicates are not allowed.

Revise your selection.

-

-Empty table read

The table read from the TABFILE file is empty.

-

-Empty table stored

The table stored into the TABFILE is been empty.

-



-Entry already exists

The key you want to rename to already exists.

-  
-Entry data wrong

The entry to be read has been damaged by a bad program.

You have to recreate this entry.

-  
-Entry not found

The specified entry was not found in the open files.

-  
-Entry size different (DASERV)

The entry sizes (number of columns and rows) of the selected files are not equal. It is therefore not possible to work on those files.

You have to make sure that the entries to be worked on are of the same size.

You possibly tried to work on entries from different table files or of different types (B and C) or you have selected different model areas.

-  
-Entry too large (DASERV)

One or more entries in the current entry selection are too large to be processed by the selected service.

Entries like these are not generally illegal. There may be special programs which are able to manage these entries.

Please use the 'List file directory' service to determine which of the selected entries are too large.

-  
-Entry too large

The specified tables are of a size (number of columns and rows) which cannot be handled by the current version of the running program.

If you like to work on the specific table, please inform programming group.

-  
-Erase not successful

The erasure of the specified file has not been successful.

-  
-Error FILDEF SORTOUT

Please call system programmer to check the program.

CMS FILEDEF has ended with error.

-

-Error file

The error file will contain the messages of error conditions which have occurred during program execution.

-

-Error filedef clear

Please call system programmer to check the program.

CMS FILEDEF has ended with error.

-

-Error in CMS filedef

Please call system programmer to check the program.

CMS FILEDEF has ended with error.

-

-Error write PARAM file

Write error has occurred writing the parameter file.

Please call system programmer to check the program.  
(disk full?).

-

-Error writing status file

Write error has occurred writing the scratch file.

Check whether there is enough space on the device the scratch file is to be written on (disk full?).

-

-Errors in input file (ORIGIN)

There are illegal data in one of the specified input files.

Please check the input files for legality of data contained.

-

-Extensions missing

You have set ACCESS = 'DIRECT', but the extensions are not defined.

FOPEN is not able to open the direct access file.

-

-Field is totally blank

No conversion done, because the input field is totally blank.

-

-File already exists

---

The file you have specified to be new already exists.

Please use a different file name or erase the old file.

-  
-File already in use

The file has been opened more than once.

This is only a warning.

-  
-File already open

The file to be opened is already open.

It is not possible to open it once more.

-  
-File full

The entries to be stored require more file space than available in the TABFILE file.

Remember that the selected service was executed incomplete.

You can use the 'List file directory' service of the program DASERV to find out what work is already done.

The 'List file status' service shows you the amount of 'deleted' records which can be made reusable by the 'Free unused space' service. Remember that the number of records displayed refers to a specific TABFILE file.

If the 'Free unused space' service does not provide sufficient space you have to create a new TABFILE file specifying a greater 'Maximum number of entries'. You may use the 'Create new file' service of the program DASERV.

-  
-File name missing

You must specify a file name.

If no meaningful preset was displayed, please check your parameter file.

-  
-File name selection

Please enter file names as required for the selected service.

You can get special help information by placing the cursor on any displayed file selection field and then pressing the help key.

To select a file from a list of existing files please use wild-card characters such as "\*" (e.g. "\* SDATA \*").

-  
-File not found

The file you have specified as an existing one does not exist.

Please use a different file name or change file status.

Check whether you have access to the device needed.

-  
-File not open

Please call system programmer to check the program.

The unit argument used for calling DAPUT, DAGET, DACLSE is not yet opened.

-  
-File not specified

Please call system programmer to check the program.

There was no unit specified by calling DAGETK.

-  
-File SORTIN not existing

The SORTIN file for external sort does not exist.

-  
-Files missing

You have not specified any files.

To get further information on file definitions put the cursor on 'Work file selection' in the previous panel (after pressing PF 3 key) and press the help key.

-  
-Fortran open error

Please call system programmer to check the program.

Fortran open has ended with error.

-  
-General file selection

Depending on your application you can specify a subset of the following files:

Protocol file

Error file

Print file

Scratch file

-  
-Help information for neutral panel fields

The cursor is placed on a neutral panel field, but this field is of no importance.

-  
-Illegal aggregate name (AGGREG)

You have selected an illegal aggregate name.  
Please specify region code (3 ch.) and sub-region code (2 ch.)

separated by blanks such as "E12 00" or "D 00".

-  
-Illegal access mode

For parameter ACCESS you can select one of the following:

- SEQUENTIAL the file is a sequential data set
- DIRECT the file is a direct access file
- ' ' FOPEN will set ACCESS = SEQUENTIAL

-  
-Illegal argument SRTFLD

The sortfield definitions are illegal.

See descriptions for sortfields of external sort program:

Bonn : SYNCSORT  
Luxembourg: VMSORT

-  
-Illegal assignments

There are illegal assignments in the processed file.

Please check the ASSIGN file.

-  
-Illegal base year (FILTER)

You have to select one base year.

Note the subkey length of two characters for input.

-  
-Illegal component (AGGREG)

Please select an aggregate component with a region code (3 ch.) and a sub-region code (2 ch.) separated by blanks such as "F 00".

-  
-Illegal cursor position

The selection by cursor has failed.

Select one of the highlighted topics.

-  
-Illegal doc. level (TREND)

You have selected an illegal documentation level.

Please select either "FULL", "ROUGH", "MINIMUM" or "NONE".

Get more information by pressing PF 1 when the cursor is placed on the input field description or see chapter "SPEL Programs" in the Technical Documentation.

-  
-Illegal file name

The file name entered is not a legal file name for the current operating system.

-  
-Illegal file specific.

The file specification entered (including wild-card characters) is not a legal file specification for the current operating system.

-  
-Illegal file status

For parameter STATUS you can select one of the following:

- NEW           the file must not exist
- OLD           the file must exist
- UNKNOWN      FOPEN checks whether the file exists or not
- MOD           FOPEN first checks the file modus.  
                  For CMS FILEDEF option DISP (disposition) will be MOD,  
                  that means the read/write pointer is positioned  
                  after the last record in the disk file.
- SCRATCH      the file must not exist. It will be created temporarily  
                  for working and erased at the end of program.
- ' '           = UNKNOWN

-  
-Illegal file type                           (FILTER)

The file format of the input file is illegal for access to FILTER.

The input file has to be of SDATA or SPELIMP format. Files of old SDATA format (binary and character) as used in previous versions of SPEL-B can be recognized for input file, too.

FILTER identifies the format of the input files automatically.

-  
-Illegal for ranges                         (DASERV)

For all subkeys where the "copy to", "compare with" or "Rename" option are used, range specifications are not allowed.

-  
-Illegal for ranges

If you specify rename subkeys, ranges are not allowed.

-  
-Illegal input

The specification for a key or subkey is illegal and cannot be analysed.

Please note the length specifications behind the key or subkey name.

---

Get help by placing the cursor on the subkey concerned and press the PF 1 key.

-  
-Illegal interval (TREND)

The definition of the regression intervals is illegal.

Please use the following format : start : end1 : end2  
p. exp : 73 : 80 : 88

The first interval has to include at least four years.

Get more information by pressing PF 1 when the cursor is placed on the input field description or see chapter "SPEL Programs" in the Technical Documentation.

-  
-Illegal interval item

The specified regression steering file contains an illegal item for interval.

The items have to be integers from 01 to 99.

Check the REGSTE file for legality of interval items.

For further detailed information about transformation types see chapter "User Interface" in the Technical Documentation.

-  
-Illegal interval ranges

The specified regression steering file contains an illegal definition of the regression interval.

The format has to be : start : end1 : end2  
e.g. : 73 : 80 : 88

The first interval must include at least 4 years. The end years must be greater than the start year. The second end year may be the same as the first end year.

Check the REGSTE file for legality of defined regression intervals.

For further detailed information about regression interval see chapter "User Interface" in the Technical Documentation.

-  
-Illegal item

The input item could not be read because of illegal characters entered.

Please check specified item with regard to definitions in the CODES file or press the PF 1 key to get help on the subkey when the cursor is placed on the subkey concerned.

-  
-Illegal key length def.

The key length definition for one subkey is not an integer value.

Check the key length definitions.

-  
-Illegal model area (FILTER)

You have to select one model area.

Note the subkey length of one character for input.

-  
-Illegal model area

You have selected an illegal model area.

Select one of the following model areas:

"S" for supply

"D" for demand

"T" for trade

-  
-Illegal out. level (TREND)

You have selected an illegal output option.

Please select either "NONE", "MISSINGS", "FULL TREND" or "COMPLETED".

Get more information by pressing PF 1 when the cursor is placed on the input field description or see chapter "SPEL Programs" in the Technical Documentation.

-  
-Illegal periodicity

The specified regression steering file contains an illegal definition of periodicity.

Check the REGSTE file for legality of defined periodicity.

-  
-Illegal range

The defined range could not be accepted because of illegal sequence of the specified bounds.

Please check specified bounds resp. codes with regard to the alphabetical order resp. to the defined sequence according to the CODES file.

-  
-Illegal RECFM

For parameter RECFM you can select one of the following:

- F fixed length

- FB fixed blocked, should not be used with TERMINAL or READER

- V variable length



- VB variable blocked, should not be used with TERMINAL or READER
- A ASA print control characters

An additional character "A" may be used with any of the valid record format settings (e.g. FA, FBA ...).

-  
-Illegal RECFM/LRECL (FILTER)

One of the specified work files has an illegal record format or an illegal logical record length. The cursor will be placed on the file name of the file concerned.

For ASSIGN file check file allocation for:  
RECFM = F and LRECL = 80

For SDATA file check file allocation for:  
RECFM = F and LRECL = 72

For SPELIMP file check file allocation for:  
RECFM = F and LRECL = 37

-  
-Illegal RECFM/LRECL (ORIGIN)

One of the specified work files has an illegal record format or an illegal logical record length. The cursor will be placed on the file name of the file concerned.

For ASSIGN file check file allocation for:  
RECFM = F and LRECL = 80

For SDATA file check file allocation for:  
RECFM = F and LRECL = 72

For SPELIMP file check file allocation for:  
RECFM = F and LRECL = 37

-  
-Illegal RECFM/LRECL (TREND)

The steering file has an illegal record format or an illegal logical record length.

Check file allocation for:  
RECFM=F and LRECL=38

-  
-Illegal record (FILTER)

The file header line of the input file of type SDATA is illegal.

It has to contain "\$STANDARD" at the beginning of the first non-comment record in the file.

Please check input file.

-  
-Illegal records found

The ASSIGN file contains illegal records.

Please check protocol file for further

information about illegal records.

-

-Illegal ref. number

The application program has tried to open a file by a reference number which is not allowed or by a reference number already in use.

-

-Illegal region number (FILTER)

You are working on an input file of old SDATA format. This file contains a numerical region code which is not defined.

Check input file for legality of numerical codes.

-

-Illeg. signif.coeff. (TREND)

You have selected an illegal significance coefficient.

Please select either "THEIL" or "DETERMINATION".

Get more information by pressing PF 1 when the cursor is placed on the input field description or see chapter "SPEL Programs" in the Technical Documentation.

-

-Illegal sub-region

You have selected an illegal sub-region code other than "00".

The model works on whole regions only, sub-regions are currently not used. The sub-region code for the whole region is "00".

Always specify "00".

-

-Illegal text length (DASERV)

The text length entered is illegal.

A legal text length is in the range from 2 to 255.

According to SPEL rules the comment text to be stored must not exceed 80 characters.

Enter "80".

-

-Illegal type (FILTER)

You have to select one type.

Note the subkey length of four characters for input.

-

-Illegal type of regress.

The specified regression steering file contains illegal types of regression (legal transformation types are LINLIN, LN LN , etc.).

---

Check the REGSTE file for legality of transformation types.

For further detailed information about transformation types see chapter "User Interface" in the Technical Documentation.

-  
-Illegal type select. (TREND)

The syntax of the input entered is not correct.

You have to separate the codes for output and input tables by " = ", separated by blanks (e.g. COMC = ZPAC COSC).

-  
-Incompatible file

The file specified is not a TABFILE file (it has not been created with the program DASERV).

Specify a TABFILE file.

To select a file from a list of existing files please use wild-card characters such as "\*" (e.g. "\* TABFILE \*").

-  
-Incompatible keys

The two files to be processed have a different key structure. Either the total key length is different or the lengths of subkeys are different.

The selected service cannot work on different key structures.

-  
-Inconsistencies found (DASERV)

Inconsistent file or directory detected. Please restore a current backup copy of the affected file

In emergencies you can use the 'Free unused space' service to remove all inconsistencies found, but then all affected entries will be lost.

The result is always a formally correct file, but there is no guarantee that the data are correct.

-  
-Incorrectly typed password (DASERV)

You have entered different passwords for new password.

Repeat the new password in both lines or press Enter without writing anything to keep the old password.

-  
-Input out of range (DASERV)

The lower limit is computed by the program. This is the number of bytes required for the whole key and descriptive text plus 8 bytes. Entries of "packed" type will require 4 more bytes. The

upper limit is a program constant.

-  
-Key length def. missing

The key length definition for one subkey is missing.  
Check your parameter file.

-  
-Key length error

The key length required by the program does not match the key length in the TABFILE file.

-  
-Key length exceeded (DASERV)

The total of the subkey lengths must be equal to the length of the whole key displayed at the top of the subkey panel.

For SPEL the length of the entry key will usually be 16 characters.

-  
-Key length not reached (DASERV)

The total of the subkey lengths must be equal to the length of the whole key displayed at the top of the subkey panel.

For SPEL the length of the entry key will usually be 16 characters.

-  
-Key not continuous (DASERV)

The sequence of subkeys must not have a gap.

-  
-Key selection

Please define the specifications for the whole key or for subkeys if existing.

-  
-Key structure not found

There is no key structure in the accessed parameter blocks.  
Check your parameter file.

-  
-LRECL conflict

The argument for LRECL (logical record length) for an existing file differs from the current logical record length of a file to be processed.

Check file allocation for LRECL and make sure that it is equal to the record length required by the current program.

-

---

-LRECL not defined

You have set ACCESS = 'DIRECT', but LRECL is not defined.

FOPEN is not able to open a direct access file without any file descriptions.

-Major table file missing (TREND)

You have selected an additional table file, while the major table is missing.

You have to select either a major table file (and possibly an additional table file) or if you do not select any table file, TREND will just check the REGSTE file for legality of trend estimation steering statements.

-MESSA1 not initialis.

The message display subroutine MESSA1 is not yet initialised.

Please call subroutine MESSON(RC).

-Mismatching array sizes

The size of the table (i.e. the number of elements = product of the number of columns by the number of rows) required by the program does not match the size of the existing table.

-Misplaced range symbol

The range selection is illegal.

The range symbols (":" and "-") have to be placed between the lower and upper bound values, always separated by blanks (e.g. AAAA : CCCC).

-Model part selection

You are able to work for different model areas.

To get a special program environment, e.g. definitions of table dimensions or alphanumeric variable names, select one of the displayed model areas.

-Modul startup logo

This screen is for information only. It displays the current settings for general files.

You may change the names of general files by pressing PF12. Then a selection panel will be displayed which will allow changes.

You have three possibilities:

- you change the names of general files and press Enter, then the specified settings for general files are used just for the

current program run.

- you change the names of general files, press PF11 and enter a new name for parameter file on which you want to save the new settings and press PF11; then the new settings will be used for the current and future program runs.
- you press PF12 twice and then load another parameter file. To select a file from a list of existing files use wild-card characters such as "\*" (e.g. "\* PARAM \*") and then press PF12.

Attention: File names have to be specified for the displayed general files.

-  
-Must be a multiple of 4 (DASERV)

The logical record length if specified must be a multiple of 4.

-  
-Name and length required (DASERV)

Please specify the name and the length for each subkey.

-  
-New parameter loaded

A new parameter environment has been loaded.  
The program starts again at the first panel using the new parameters.

-  
-No dot found  
There has been no dot in the character string.  
Nevertheless conversion has been done.

-  
-No free scratch file  
The number of scratch files to be opened exceeds the maximum of 20 files.

-  
-No matching files  
There are no files matching your selection by wild-card characters.

-  
-No parameter found  
You have specified a parameter file which does not contain the parameter block for the current program.  
Please check parameter file.

-  
-No protocol file found

No protocol file has been allocated  
for this program run.

-  
-Not a real number

The input item cannot be read because of illegal characters  
entered nor can it be converted to the destination data type.

Legal inputs have to be real numbers (e.g. "-3.", "3.", "3.5",  
"3.5E10", "-3.5E-10").

Check your input and the data file for legality of numbers.

-  
-Not an integer number

The input item cannot be read because of illegal characters  
entered nor can it be converted to the destination data type.

Legal inputs have to be integer numbers (e.g. "-3", "3").

Check your input and the data file for legality of numbers.

-  
-Not enough disk space

The disk space needed is not available on the device.

Please decrease the space requirements or specify  
another device.

-  
-Not enough disk space

The new TABFILE file to be created requires more additional disk  
space than available on the specified disk.

The service has not been executed.

To create the new TABFILE you have either to specify fewer or  
smaller entries or free some disk space or specify another disc.

-  
-Not right justified

The character value is not right justified  
on the character string.

-  
-Number too small

Your input is meaningless.  
Specify more entries.

-  
-Only comments (FILTER)

The input file contains only records filled with comments.

Please check input file.

-  
-Output file not specif.

None of the open TABFILE files was opened for output.

-  
-Output file not the last

Please call system programmer to check the program.

The output file has to be the TABFILE file opened last.

-  
-Output not allowed (SPEL)

The TABFILE file is not open for output.

-  
-Parameter added

An additional parameter has been added to the parameter block.

-  
-Parameter block not loaded

The parameter block to be used for a program run is not loaded.

Check PMLOAD statement and the definition for 'Additional parameter blocks' in the parameter file.

-  
-Parameter block protect.

The parameter block is protected for updates.

You cannot save your parameter setting.

-  
-Parameter blocks missing

At least one parameter block is not specified in your parameter file.

Check parameter file.

-  
-Parameter file

General control file of PARAM format containing external parameters to define the program environment.

For further detailed information about general control files see chapter "User Interface" in the Technical Documentation.

-  
-Parameter not found

At least one parameter necessary for program execution is not available in any parameter block.

Check parameter file.

-  
-Parameter saved



The actual parameter setting has been saved.

-  
-Parameter syntax error

One of the parameter definitions has an illegal syntax.

Check parameter file.

For further detailed information see chapter "User Interface" in the Technical Documentation.

-  
-PF-keys for modul startup logo

Enter = ok : Continue program flow.

PF 1 = Help : Get help information. The help key is context sensitive. That means, you can get special help depending on the cursor position.

PF 3 = Quit : Quit the panel.

PF12 = General file:  
You may modify the names of the general files displayed.

-  
-PF-keys for save/load parameter

PF 1 = Help : Get help information. The help key is context-sensitive which means you can get special help depending on cursor position.

PF 3 = Quit : Quit the panel.

PF 4 = Exit : Exit the program.

PF11 = Save : Save current parameter setting.

PF12 = Load : Load parameter file.

-  
-PF-keys for standard display

Depending on the size of information to be listed, different levels of PF-keys to control the listing are active.

The active keys are displayed at the bottom of the screen.

Enter = ok : Continue program flow.

PF 1 = Help : Get help information. The help key is context-sensitive. That means, you can get special help depending on cursor position.

PF 3 = Quit : Quit the current panel.

PF 5 = Top : Move the screen to the top of the information.

PF 6 = Bottom : Move the screen to the bottom of the information.

PF 7 = Page up : Move the screen one page up.

PF 8 = Page down : Move the screen one page down.

PF 9 = Page left : Move the screen one page left.

PF10 = Page right: Move the screen one page right.

Move screen operations are not available if the whole information fits on one screen page. Help information other than PF-key information may be available depending on the application program.

-  
-PF-keys for standard menus

Enter = ok : Continue program flow.

PF 1 = Help : Get help information. The help key is context-sensitive which means you can get special help depending on the cursor position.

PF 3 = Quit : Quit the panel.

PF 4 = Exit : Exit the program.

PF11 = Load/save parameters.

-  
-Premature end of file (ORIGIN)

One of the specified input files of SDATA format has an unexpected end.

Please check whether the input files are complete.

-  
-Print file

The print file is reserved for printer listings to be printed later on.

You may also directly specify the printer name.

-  
-Protocol file

In the protocol file you will find a copy of all messages ever displayed on screen during program execution. More detailed information about work steps will be included as well.

-  
-Range not allowed here (DASERV)

For all subkeys where the "copy to", "compare with" or "Rename" option is used, range specifications are not allowed.

-  
-Range not allowed here

If you specify rename subkeys, ranges are not allowed.

-  
-Read error (FILTER)

Error has occurred reading the input file.

Please check file and make sure that you have access to the specified device.

-  
-RECFM conflict

The argument for RECFM (record format) for an existing file differs from the current record format length of a file to be processed.

Check file allocation for RECFM and make sure that it is equal to the record format required by the current program.

-  
-Record length too small

The record length of the output file is too small.

The minimum record length is determined by the length of the key, the length of the text and the entry type ("packed" or "normal").

The specifications for the creation of the output file by the program DASERV do not allow the entry to be written into the output file.

If you cannot cut the text part of the entry, use the program DASERV to create a new file with correct specifications and copy the existing entries into the new file.

The old file may be deleted.

-  
-Ref. number illegal

The reference number for the file protocol is illegal.

-  
-Save/load parameter

Every SPEL program has to be initialised by external parameters saved on general control files of type PARAM.

To get a better overview the parameters are collected in blocks referring to special applications.

To make sure that all programs have a well defined environment, default parameters for all applications are saved on a file named DEFAULT (in CMS: "DEFAULT PARAM \*").

When invoking a program run you may define a special parameter file, if missing the default file will be used.

You may save a modified parameter environment by pressing PF11. N.B.: saving the current parameter environment does not mean that there will be an updating of the old parameter file. Only the parameters for the current program will be saved in the parameter file. If you don't specify a new parameter file name to save the current parameter settings, parameter blocks not belonging to the environment of the current program will be lost in the old parameter file.

You may load a new parameter environment by pressing PF12.

For both load and save the displayed parameter file is used.

For further detailed information about general control files see chapter "User Interface" in the Technical Documentation.

-  
-Scratch file

The scratch file is a temporary work file and will be internally used by the program and erased at the end of program.

Please ensure that enough space is available on the specified device for creating a scratch file.

-  
-Scratch file write error (DASERV)

The current service uses a 'scratch file' for storing intermediate information. A system error during a write operation on this file has occurred.

This error occurs when the related disk has too little space.

You may reduce the scratch space needed by selecting fewer entries for a service at one time. Otherwise please free some disk space by deleting files no longer used and then retry this service.

-  
-Selection by cursor

The panel displays a list for selection.

Put the cursor on the topic to be selected and press Enter.

-  
-Sequence error (FILTER)

The record sequence of the input file is not correct.

This file format has got a well-defined record sequence.

For further detailed information about legal record sequences see chapter "Data Management" in the Technical Documentation.

Check the processed file for legality of record sequences.

-  
-Sort error:

The external sort program has ended due to error.

See descriptions for sortfields of external sort program:

Bonn : SYNCSORT  
Luxembourg: VMSORT

-  
-Status conflict

---

A file name used for a scratch file already exists.

-  
-Structure not defined

The specified structure is not defined.

Check subkey specification in the key selection panel for legality:

- Make sure that you have specified table type codes with "B" or "C" for the fourth character and check if the codes are defined in the CODES file.
- Ensure that you have specified a legal model area. Legal model areas are "D", "S", "T".

Later you may have to check the internal file directory using the 'List file directory' service in program DASERV.

-  
-Subkey 5 protected (ORIGIN)

This subkey is protected.

The subkey for base year will be always 'NN' when putting original data (ex-post data) into the SPEL system.

-  
-Subkey length not found

There is no subkey length definition in the accessed parameter blocks.

Check your parameter file.

-  
-Subkey protected

This subkey is protected.

You must not make any input.

-  
-Subkey too long

In the specification of a key or subkey one or more elements are too long.

Please note the length specifications behind the key or subkey name.

-  
-TABFILE only READ access (TREND)

You have specified an output level for table output, but the specified file of TABFILE format is write protected, which means you may read from it, but you must not write into it.

Please make sure that you have specified the correct TABFILE.

-  
-Too few data records (DASERV)

Normally the field for the number of data records should be left blank. In this case the program estimates the value from the maximum number of entries and the user estimates for the number of data elements. If a value is entered for the number of data records, it must be at least 1.

-  
-Too many assign. records

The number of assignment statements exceeds the maximum of 4000.

If you have used an ASSIGN file, reduce the number of assignment statements in the ASSIGN file.

Otherwise inform your system programmer to check program constants.

-  
-Too many base years (FILTER)

The base year selection is limited to one base year.

Revise your base year selection.

-  
-Too many block names

The number of blocks exceeds the maximum of 20.

-  
-Too many digits

In the specification of a key or subkey one or more elements are too long.

Please notice the length specifications behind the key name.

-  
-Too many entries in file

The additional number of new entries to be created exceeds the maximum number of entries which had been specified when creating the TABFILE file currently used for output.

Remember that the selected service was executed incomplete.

You can use the 'List file directory' service of the program DASERV to find out what work is already done.

The 'List file status' service shows you the amount of 'deleted' entries which can be made reusable by the 'Free unused space' service. Remember that the number of entries displayed refers to a specific TABFILE file.

If the 'Free unused space' service does not provide sufficient space you have to create a new TABFILE file specifying a greater 'Maximum number of entries'. You may use the 'Create new file' service of the program DASERV.

-  
-Too many files open

The application program has too many files in use

at the same time.

Please close all files which are no longer needed.

-  
-Too many files selected

You have specified more files than the system is able to work on.

The files displayed are incomplete with regard to your selection.

If the desired files are not displayed, you must specify a smaller number of files.

-  
-Too many items

You have selected an illegal number of items.

Please check the number of items and make sure you don't exceed the maximum number.

-  
-Too many model areas (FILTER)

The model area selection is limited to one model area.

Revise your model area selection.

-  
-Too many model areas

You have selected too many model areas.

You must not select more than one of the following areas:

"S" for supply

"D" for demand

"T" for trade

-  
-Too many parameters

The number of parameters exceeds the maximum of 500.

-  
-Too many TABFILE files

The attempt to open another TABFILE file has failed, because the maximum number of TABFILE files to be open simultaneously has already been reached (the maximum is 10).

-  
-Too many types (FILTER)

The type selection is limited to one type.

Revise your type selection.

-

-Too many years (FILTER)

You have selected too many years.

The maximum number of years to be selected is 47.  
Make sure you do not exceed the limit.

-  
-Too many years (TREND)

The definition of the regression intervals exceeds the  
maximum number of years.

The limit for the long interval is 30 years.

-  
-Two dimensions supported (DASERV)

For the selected service only tables of two  
dimensions are supported.

-  
-Undefined input code (TREND)

You have selected one or more undefined codes for input tables.

Please check the specified type codes with regard to the  
definitions in the CODES file.

-  
-Undefined output code (TREND)

You have selected an undefined code for output table.

Please check the specified type code with regard to the  
definitions in the CODES file.

-  
-Undefined PF-key

The PF-key you have pressed is not defined.

All defined PF-keys are displayed at the bottom of the panel.

-  
-Undefined type code

One of the specified type codes is not defined in the  
CODES file.

Please check the specified type codes with regard to  
the definitions in the CODES file.

-  
-Unit not yet opened

The unit to be closed has not been opened yet.

-  
-Value too great

The value cannot be stored on an INTEGER\*4 variable,  
because it is greater than 2147483647.



-  
-Wrong number of keys

The number of rename specifications for a key or subkey does not fit the number of the original specifications.

Because the number of elements in the second line is different from the first, the program cannot establish the correct relationships.

Revise the selection ensuring that the number of codes is the same in both lines.

-  
-Wrong structure size (DASERV)

The size information in the entry data structure definition does not match the current entry size (Number of rows and columns of TABFILE tables).

Ensure that the current entry structure definition is updated.

Please call system programmer to check program. (Beforehand the entry structure definition is completed by means of subroutine DAGETC. The routine must be recompiled after changes in table structure).

## 8. INDEX

- AGGREG program, 157, 180, 202
- aggregate, 186
- aggregate code, →code
- aggregate component, 186
- aggregation, 202
- aggregation of user defined regions, 157, 180
- aggregation to EC level, 174
- ASSIGN file, 167, 192, 205
- ASSIGN format, 202, 203
- Base Model, 157, 172, 174, 187, 197, 202
- base year, → differentiation criteria
- BASEMOD program, 157, 174, 177
- basic structure, 186
- batch mode, 165
- best fit criteria, 169, 195
- CMS command level, 160, 208
- CMS dialog system, 208
- code
  - aggregate code, 183, 185
  - domain code, 192, 205
  - domain type code, 166
  - external data source code, 167, 203
  - model area code, 184
  - original item code, 167
  - periodicity code, 179, 184, 193, 202
  - region code, 178, 183, 186, 193, 201
  - SPEL table element code, 167, 192, 203, 205
  - sub-region code, 178, 183, 186, 193, 201
  - type code, 179, 180, 193
- CODES file, 185, 186, 193, 201
- coefficient of determination, 169, 188, 195
- completion by subjective estimation, 157, 168
- completion by trend estimation, 157, 169
- consistency at Member State level, 157, 174
- consistency at regional level, 174
- control parameter, 168
- current year, → differentiation criteria
- DAOUT program, 206
- DASERV program, 204
- data compilation, 157, 167
- data extraction and compilation, 157, 162, 164
- data format conversion, 157, 165
- data preparation, 156, 157, 162, 174
- data selection, 157, 167
- data set completion, 157, 162, 168
- data transformation, 168, 169, 188, 195
- DATCON program, 157, 164, 165, 166, 167
- DEFAULT PARAM, 165
- deflator, 195, 203
- dependent variable, 188
- differentiation criteria, 167
  - base year, 179, 184, 193, 202
  - current year, 179, 183
  - model area, 179, 184, 194, 202
  - periodicity, 179, 184, 193, 202
  - projection year, 193, 201
  - region, 178, 193, 201
  - region aggregate, 183
  - sub-region, 178, 193, 201
  - table output type, 193
  - type, 179, 184, 202
- documentation level, 195
- domain code, → code
- domain type code, → code
- EC aggregation level, 157, 174
- endogenous variable, 186, 187, 197
- Error messages and warnings, 211
- estimation
  - non trend estimation, 203
  - single equation regression estimation, 205
  - subjective estimation, 168
  - trend estimation, 168, 169, 188
- EV program, 157, 168, 169, 205, 206

- 
- ex-post period, 188
  - exogenous data, 208
  - exogenous data file, 165
  - exogenous variable, 186, 187, 188, 194, 195, 197, 200
  - expert, 168, 187
  - expert's checking, 188
  - expert's judgement, 187, 188, 196, 197, 200
  - expert's proposal, 188
  - EXPLOIT program, 209
  - EXPLOITATION, 209
  - exploitation, 161, 205
  - export data, → DAOUT program
  - external data source code, → code
  - external source data, 164, 192
  - external source format, 165
  - file list of the output disk, 208
  - FILTER program, 157, 164, 167, 205
  - fitting criteria, 188
  - forecast simulation, 157, 197
  - format conversion, 165
  - general estimation and evaluation program, 205
  - growth factor, 188
  - growth factors series, 188
  - growth rate, 186, 187, 188, 197, 200
  - independent variable, 168, 203
  - index Purchasing Power Standard, 203
  - index Purchasing Power Standard (PPS), 195
  - indicator file, 192
  - indicator time series, 188, 192
  - indicator variable, 188
  - link of the SPEL disks, 207
  - LISTING-file, 176, 190
  - LOG-file, 176, 182, 190, 199, 207
  - maintenance of TABFILE files, 204
  - model area, → differentiation criteria
  - model area code, → code
  - model calculations, 156, 157, 172
  - model component, 156, 157, 160, 161
  - model component selection, 161
  - model simulation result, 208
  - non trend estimation, → estimation
  - operator, 160, 161, 162, 164, 165, 207, 208, 211
  - ORIGIN program, 157, 164, 167
  - original item code, → code
  - output level, 195
  - output to experts level, 196
  - panel, 156, 160, 161, 162, 164, 165, 167, 168, 176, 182, 186, 190, 199, 205, 207, 211
    - display panel, 160
    - edit panel, 160
    - selection panel, 160, 161, 164
  - PARAM file, 165
  - periodicity, → differentiation criteria
  - periodicity code, → code
  - price index of gross domestic product, 203
  - price index of gross domestic product (GDP), 195
  - print tables, → DAOUT program
  - PROFILE EXEC file, 160
  - program control parameter, 166
  - program parameter, 165
  - program parameter block, 166
  - reference interval, 194
  - region, → differentiation criteria
  - region code, → code
  - regression, 168
    - regression equation, 168
    - regression parameter, 188
    - regression steering file, 191
    - regression steering record, 202
    - simple regression, 169
  - REGSTE file, 169, 188, 191, 194, 195

- REGSTE format, 191, 202
- SDATA file, 167, 200, 205
- SDATA format, 167, 177, 187, 205
- selection and conversion of SPELIMP and SDATA files, 205
- SFPROP program, 157, 187, 188, 197
- SFS program, 157, 187, 197, 202
- SFSS, → Short term forecast and simulation system, 186, 187, 202, 203
- Short term forecast and simulation system, 157, 172, 186, 202
- significance coefficient, 195
- single equation regression estimation, → estimation
- SPEL data base output, 206
- SPEL disk
  - A disk, 160
  - C disk, 207, 208
  - E disk, 208
  - F disk, 207
  - H disk, 208
  - I disk, 207
  - J disk, 208
  - L disk, 208
  - operator/user's disk, 207
  - SPEL operator system disk, 160
  - SPEL system disk, 160, 207
  - SPEL/EC disk, 208
  - SPEL/EC disk base data, 160
  - SPEL/EC disk exogenous data, 160
  - SPEL/EC disk for model simulation results, 208
  - SPEL/EC disk model simulation results, 160
  - SPEL/EC system disk, 160, 207
  - temporary disk, 160, 207
- SPEL environment, 160
  - operator environment, 160
  - user environment, 160
- SPEL EXEC command, 160
- SPEL EXEC program, 160
- SPEL logo, 161
- SPEL operator file, 208
- SPEL programs, 207
- SPEL system, 155
- SPEL system file, 207
- SPEL table, 174, 177
- SPEL table element code, → code
- SPEL team, 160, 165, 210
- SPEL/EC Base Model data, 208
- SPEL/EC external data, 208
- SPEL/EC Model, 155, 156, 157, 160, 211
- SPEL/EC shell, 160, 162, 164, 165, 166, 168, 187, 205, 207, 208, 211
- SPEL/EC simulation data, 208
- SPEL/EC system file, 207
- SPEL/EC-Data, 160, 174, 186, 188, 191, 200
- SPELIMP file, 167, 192, 205, 208
- SPELIMP format, 164, 165, 167, 205
- STEER file, 168, 169, 205, 206
- STEER format, 169
- sub-region, → differentiation criteria
- sub-region code, → code
- subjective estimation, → estimation
- system proposal, 187, 188, 195, 196, 197, 200
- system proposals for exogenous variables, 157, 188
- TABFILE file, 191, 200, 206
  - TABFILE work file, 164, 167, 174, 177, 183, 191, 197, 200, 204
- TABFILE format, 177, 183, 205
- tape unit, 165
- tape-id, 165, 166
- Theil's coefficient, 188, 195
- time series, 168, 188
- trend estimation, → estimation
- TREND program, 157, 168, 169
- type, → differentiation criteria, 180
- type code, → code
- User, 165
- user, 160, 161, 207, 211

user interface, 156

utilities, 160, 161, 204

utility selection, 161

work file, 195

work sector, 156, 157, 160, 162

work sector selection, 162, 172

work step, 156, 157, 160, 162

work step selection, 164, 168, 174, 187



**ES** Clasificación de las publicaciones de Eurostat**TEMA**

- 1 Estadísticas generales (azul oscuro)
- 2 Economía y finanzas (violeta)
- 3 Población y condiciones sociales (amarillo)
- 4 Energía e industria (azul claro)
- 5 Agricultura, silvicultura y pesca (verde)
- 6 Comercio exterior (rojo)
- 7 Servicios y transportes (naranja)
- 8 Medio ambiente (turquesa)
- 9 Diversos (marrón)

**SERIE**

- A Anuarios
- B Coyuntura
- C Cuentas, encuestas y estadísticas
- D Estudios y análisis
- E Métodos
- F Estadísticas rápidas

**GR** Ταξινόμηση των δημοσιεύσεων της Eurostat**ΘΕΜΑ**

- 1 Γενικές στατιστικές (βαθύ μπλε)
- 2 Οικονομία και δημοσιονομικά (βιολετί)
- 3 Πληθυσμός και κοινωνικές συνθήκες (κίτρινο)
- 4 Ενέργεια και βιομηχανία (μπλε)
- 5 Γεωργία, δάση και ολιείο (πράσινο)
- 6 Εξωτερικό εμπόριο (κόκκινο)
- 7 Υπηρεσίες και μεταφορές (πορτοκαλί)
- 8 Περιβάλλον (τουρκουάζ)
- 9 Διάφορο (κοφέ)

**ΣΕΙΡΑ**

- A Επετηρίδες
- B Συγκυρία
- C Λογαριασμοί, έρευνες και στατιστικές
- D Μελέτες και αναλύσεις
- E Μέθοδοι
- F Τοχεία στατιστικές

**IT** Classificazione delle pubblicazioni dell'Eurostat**TEMA**

- 1 Statistiche generali (blu)
- 2 Economia e finanze (viola)
- 3 Popolazione e condizioni sociali (giallo)
- 4 Energia e industria (azzurro)
- 5 Agricoltura, foreste e pesca (verde)
- 6 Commercio estero (rosso)
- 7 Servizi e trasporti (arancione)
- 8 Ambiente (turchese)
- 9 Diversi (marrone)

**SERIE**

- A Annuari
- B Tendenze congiunturali
- C Conti, indagini e statistiche
- D Studi e analisi
- E Metodi
- F Note rapide

**DA** Klassifikation af Eurostats publikationer**EMNE**

- 1 Almene statistikker (merkeblå)
- 2 Økonomi og finanser (violet)
- 3 Befolkning og sociale forhold (gul)
- 4 Energi og industri (blå)
- 5 Landbrug, skovbrug og fiskeri (grøn)
- 6 Udenrigshandel (rød)
- 7 Tjenesteydelser og transport (orange)
- 8 Miljø (turkis)
- 9 Diverse statistikker (brun)

**SERIE**

- A Årbøger
- B Konjunkturoversigter
- C Regnskaber, tællinger og statistikker
- D Undersøgelser og analyser
- E Metoder
- F Ekspresoversigter

**EN** Classification of Eurostat publications**THEME**

- 1 General statistics (midnight blue)
- 2 Economy and finance (violet)
- 3 Population and social conditions (yellow)
- 4 Energy and industry (blue)
- 5 Agriculture, forestry and fisheries (green)
- 6 Foreign trade (red)
- 7 Services and transport (orange)
- 8 Environment (turquoise)
- 9 Miscellaneous (brown)

**SERIES**

- A Yearbooks
- B Short-term trends
- C Accounts, surveys and statistics
- D Studies and analyses
- E Methods
- F Rapid reports

**NL** Classificatie van de publicaties van Eurostat**ONDERWERP**

- 1 Algemene statistiek (donkerblauw)
- 2 Economie en financiën (paars)
- 3 Bevolking en sociale voorwaarden (geel)
- 4 Energie en industrie (blauw)
- 5 Landbouw, bosbouw en visserij (groen)
- 6 Buitenlandse handel (rood)
- 7 Diensten en vervoer (oranje)
- 8 Milieu (turkoois)
- 9 Diverse statistieken (bruin)

**SERIE**

- A Jaarboeken
- B Conjunctuur
- C Rekeningen, enquêtes en statistieken
- D Studies en analyses
- E Methoden
- F Spoedberichten

**DE** Gliederung der Veröffentlichungen des Eurostat**THEMENKREIS**

- 1 Allgemeine Statistik (Dunkelblau)
- 2 Wirtschaft und Finanzen (Violett)
- 3 Bevölkerung und soziale Bedingungen (Gelb)
- 4 Energie und Industrie (Blau)
- 5 Land- und Forstwirtschaft, Fischerei (Grün)
- 6 Außenhandel (Rot)
- 7 Dienstleistungen und Verkehr (Orange)
- 8 Umwelt (Türkis)
- 9 Verschiedenes (Braun)

**REIHE**

- A Jahrbücher
- B Konjunktur
- C Konten, Erhebungen und Statistiken
- D Studien und Analysen
- E Methoden
- F Schnellberichte

**FR** Classification des publications de l'Eurostat**THÈME**

- 1 Statistiques générales (bleu nuit)
- 2 Économie et finances (violet)
- 3 Population et conditions sociales (jaune)
- 4 Énergie et industrie (bleu)
- 5 Agriculture, sylviculture et pêche (vert)
- 6 Commerce extérieur (rouge)
- 7 Services et transports (orange)
- 8 Environnement (turquoise)
- 9 Divers (brun)

**SÉRIE**

- A Annuaire
- B Conjoncture
- C Comptes, enquêtes et statistiques
- D Études et analyses
- E Méthodes
- F Statistiques rapides

**PT** Classificação das publicações do Eurostat**TEMA**

- 1 Estatísticas gerais (azul escuro)
- 2 Economia e finanças (violeta)
- 3 População e condições sociais (amarelo)
- 4 Energia e indústria (azul)
- 5 Agricultura, silvicultura e pesca (verde)
- 6 Comércio externo (vermelho)
- 7 Serviços e transportes (laranja)
- 8 Ambiente (turquesa)
- 9 Diversos (castanho)

**SÉRIE**

- A Anuários
- B Conjuntura
- C Contas, inquéritos e estatísticas
- D Estudos e análises
- E Métodos
- F Estatísticas rápidas





European Communities — Commission

**SPEL System — Technical documentation**

Luxembourg: Office for Official Publications of the European Communities

1992 — 244 pp. — 21.0 x 29.7 cm

Theme 5: Agriculture, forestry and fisheries (green cover)

Series E: Methods

ISBN 92-826-4498-7

Price (excluding VAT) in Luxembourg: ECU 16

The technical documentation of the SPEL System consists of two comprehensive parts for users and operators working with the SPEL on-line system. It outlines the principal technical approach for the whole SPEL System, which is one of user interface flexibility and transparency. Transparency means that data can be traced back to the basic sources and the underlying assumptions.

Part 1 entitled **OPERATOR/USER'S GUIDE: BASICS** describes the fundamental concepts for implementation. It begins with the system requirements, followed by an outline of formal structuring, coding, and concludes with principles of data management and user interfaces. A supplementary chapter contains a description of multi-purpose programs and utilities.

Part 2 entitled **OPERATOR/USER'S GUIDE: SPEL/EC MODEL** describes the generation of the SPEL/EC model database, respecting the methodological ideas and the technical basics. The guide outlines, step by step, the model components (data preparation and model calculations) as they are embedded in the program shell. The guide concludes with a presentation of utilities and exploitation programs offered to the SPEL/EC user.



**Venta y suscripciones • Salg og abonnement • Verkauf und Abonnement • Πωλήσεις και συνδρομές  
Sales and subscriptions • Vente et abonnements • Vendita e abbonamenti  
Verkoop en abonnementen • Venda e assinaturas .**

**BELGIQUE / BELGIË**

**Moniteur belge / Belgisch Staatsblad**  
Rue de Louvain 42 / Leuvenseweg 42  
B-1000 Bruxelles / B-1000 Brussel  
Tél. (02) 512 00 26  
Fax (02) 511 01 84

Autres distributeurs / Overige verkooppunten

**Librairie européenne / Europese boekhandel**

Rue de la Loi 244 / Wetstraat 244  
B-1040 Bruxelles / B-1040 Brussel  
Tél. (02) 231 04 35  
Fax (02) 735 08 60

**Jean De Lannoy**

Avenue du Roi 202 / Koningslaan 202  
B-1060 Bruxelles / B-1060 Brussel  
Tél. (02) 538 51 69  
Télex 63220 UNBOOK B  
Fax (02) 538 08 41

**Document delivery:**

**Credoc**

Rue de la Montagne 34 / Bergstraat 34  
Bte 11 / Bus 11  
B-1000 Bruxelles / B-1000 Brussel  
Tél. (02) 511 69 41  
Fax (02) 513 31 95

**DANMARK**

**J. H. Schultz Information A/S**

**EF-Publikationer**

Ottiliavej 18  
DK-2500 Valby  
Tlf. 36 44 22 66  
Fax 36 44 01 41

**DEUTSCHLAND**

**Bundesanzeiger Verlag**

Breite Straße  
Postfach 10 80 06  
D-W 5000 Köln 1  
Tel. (02 21) 20 29-0  
Telex ANZEIGER BONN 8 882 595  
Fax 2 02 92 78

**GREECE/ΕΛΛΑΔΑ**

**G.C. Eleftheroudakis SA**

International Bookstore  
Nikis Street 4  
GR-10563 Athens  
Tel. (01) 322 63 23  
Telex 219410 ELEF  
Fax 323 98 21

**ESPAÑA**

**Boletín Oficial del Estado**

Trafalgar, 29  
E-28071 Madrid  
Tel. (91) 538 22 95  
Fax (91) 538 23 49

**Mundi-Prensa Libros, SA**

Castelló, 37  
E-28001 Madrid  
Tel. (91) 431 33 99 (Libros)  
431 32 22 (Suscripciones)  
435 36 37 (Dirección)  
Télex 49370-MPLI-E  
Fax (91) 575 39 98

Sucursal:

**Librería Internacional AEDOS**

Consejo de Ciento, 391  
E-08009 Barcelona  
Tel. (93) 488 34 92  
Fax (93) 487 76 59

**Libreria de la Generalitat de Catalunya**

Rambla dels Estudis, 118 (Palau Moja)  
E-08002 Barcelona  
Tel. (93) 302 68 35  
302 64 62  
Fax (93) 302 12 99

**FRANCE**

**Journal officiel  
Service des publications  
des Communautés européennes**

26, rue Desaix  
F-75727 Paris Cedex 15  
Tél. (1) 40 58 75 00  
Fax (1) 40 58 75 74

**IRELAND**

**Government Supplies Agency**

4-5 Harcourt Road  
Dublin 2  
Tel. (1) 61 31 11  
Fax (1) 78 06 45

**ITALIA**

**Licosa Spa**

Via Duca di Calabria, 1/1  
Casella postale 552  
I-50125 Firenze  
Tel. (055) 64 54 15  
Fax 64 12 57  
Telex 570466 LICOSA I

**GRAND-DUCHÉ DE LUXEMBOURG**

**Messageries Paul Kraus**

11, rue Christophe Plantin  
L-2339 Luxembourg  
Tél. 499 88 88  
Télex 2515  
Fax 499 88 84 44

**NEDERLAND**

**SDU Overheidsinformatie**

Externe Fondsen  
Postbus 20014  
2500 EA 's-Gravenhage  
Tel. (070) 37 89 911  
Fax (070) 34 75 778

**PORTUGAL**

**Imprensa Nacional**

Casa da Moeda, EP  
Rua D. Francisco Manuel de Melo, 5  
P-1092 Lisboa Codex  
Tel. (01) 69 34 14

**Distribuidora de Livros  
Bertrand, Ld."**

**Grupo Bertrand, SA**  
Rua das Terras dos Vales, 4-A  
Apartado 37  
P-2700 Amadora Codex  
Tel. (01) 49 59 050  
Telex 15798 BERDIS  
Fax 49 60 255

**UNITED KINGDOM**

**HMSO Books (PC 16)**

HMSO Publications Centre  
51 Nine Elms Lane  
London SW8 5DR  
Tel. (071) 873 2000  
Fax GP3 873 8463  
Telex 29 71 138

**ÖSTERREICH**

**Manz'sche Verlags-  
und Universitätsbuchhandlung**

Kohlmarkt 16  
A-1014 Wien  
Tel. (0222) 531 61-0  
Telex 112 500 BOX A  
Fax (0222) 531 61-39

**SUOMI**

**Akateeminen Kirjakauppa**

Keskuskatu 1  
PO Box 128  
SF-00101 Helsinki  
Tel. (0) 121 41  
Fax (0) 121 44 41

**NORGE**

**Narvesen information center**

Bertrand Narvesens vei 2  
PO Box 6125 Etterstad  
N-0602 Oslo 6  
Tel. (2) 57 33 00  
Telex 79668 NIC N  
Fax (2) 68 19 01

**SVERIGE**

**BTJ**

Tryck Traktorvägen 13  
S-222 60 Lund  
Tel. (046) 18 00 00  
Fax (046) 18 01 25

**SCHWEIZ / SUISSE / SVIZZERA**

**OSEC**

Stampfenbachstraße 85  
CH-8035 Zürich  
Tel. (01) 365 54 49  
Fax (01) 365 54 11

**ČESKOSLOVENSKO**

**NIS**

Havelkova 22  
13000 Praha 3  
Tel. (02) 235 84 46  
Fax 42-2-264775

**MAGYARORSZÁG**

**Euro-Info-Service B.T.**

Rádáy u. 24/B  
H-1092 Budapest  
Tel. (1) 36 1 118  
Fax (1) 36 1 72 83

**POLSKA**

**Business Foundation**

ul. Krucza 38/42  
00-512 Warszawa  
Tel. (22) 21 99 93, 628-28-82  
International Fax&Phone  
(0-39) 12-00-77

**CYPRUS**

**Cyprus Chamber of Commerce and Industry**

Chamber Building  
38 Grivas Digenis Ave  
3 Deligiorgis Street  
PO Box 1455  
Nicosia  
Tel. (2) 449500/462312  
Fax (2) 458630

**TÜRKIYE**

**Pres Gazete Kitap Dergi  
Pazarlama Dağıtım Ticaret ve sanayi  
AŞ**

Narlibahçe Sokak N. 15  
Istanbul-Cağaloğlu  
Tel. (1) 520 92 96 - 528 55 66  
Fax 520 64 57  
Telex 23822 DSVO-TR

**CANADA**

**Renouf Publishing Co. Ltd**

Mail orders — Head Office:  
1294 Algoma Road  
Ottawa, Ontario K1B 3W8  
Tel. (613) 741 43 33  
Fax (613) 741 54 39  
Telex 0534783

Ottawa Store:

61 Sparks Street  
Tel. (613) 238 89 85

Toronto Store:

211 Yonge Street  
Tel. (416) 363 31 71

**UNITED STATES OF AMERICA**

**UNIUPUB**

4611-F Assembly Drive  
Lanham, MD 20706-4391  
Tel. Toll Free (800) 274 4888  
Fax (301) 459 0056

**AUSTRALIA**

**Hunter Publications**

58A Gipps Street  
Collingwood  
Victoria 3066

**JAPAN**

**Kinokuniya Company Ltd**

17-7 Shinjuku 3-Chome  
Shinjuku-ku  
Tokyo 160-91  
Tel. (03) 3439-0121

**Journal Department**

PO Box 55 Chitose  
Tokyo 156  
Tel. (03) 3439-0124

**RUSSIA**

**CCEC (Centre for Cooperation with  
the European Communities)**

9, Prospekt 60-let Oktyabrya  
117312 Moscow  
Tel. 007 095 135 52 87  
Fax 007 095 420 21 44

**ISRAEL**

**ROY International**

PO Box 13056  
41 Mishmar Hayarden Street  
Tel Aviv 69865  
Tel. 00972 3 496 108  
Fax 00972 3 544 60 39

**SINGAPORE**

**Legal Library Services Ltd**

STK Agency  
Robinson Road  
PO Box 1817  
Singapore 9036

**AUTRES PAYS**

**OTHER COUNTRIES  
ANDERE LÄNDER**

**Office des publications officielles  
des Communautés européennes**

2, rue Mercier  
L-2985 Luxembourg  
Tél. 499 28 1  
Télex PUBOF LU 1324 b  
Fax 48 85 73/48 68 17

Price (excluding VAT) in Luxembourg  
ECU 16



OFICINA DE PUBLICACIONES OFICIALES DE LAS COMUNIDADES EUROPEAS  
KONTOR ET FOR DE EUROPÆISKE FÆLLESSKABERS OFFICIELLE PUBLIKATIONER  
AMT FÜR AMTLICHE VERÖFFENTLICHUNGEN DER EUROPÄISCHEN GEMEINSCHAFTEN  
ΥΠΗΡΕΣΙΑ ΕΠΙΣΗΜΩΝ ΕΚΔΟΣΕΩΝ ΤΩΝ ΕΥΡΩΠΑΪΚΩΝ ΚΟΙΝΟΤΗΤΩΝ  
OFFICE FOR OFFICIAL PUBLICATIONS OF THE EUROPEAN COMMUNITIES  
OFFICE DES PUBLICATIONS OFFICIELLES DES COMMUNAUTÉS EUROPÉENNES  
UFFICIO DELLE PUBBLICAZIONI UFFICIALI DELLE COMUNITÀ EUROPEE  
BUREAU VOOR OFFICIËLE PUBLIKATIES DER EUROPESE GEMEENSCHAPPEN  
SERVIÇO DAS PUBLICAÇÕES OFICIAIS DAS COMUNIDADES EUROPEIAS

ISBN 92-826-4498-7

