



International Institute for  
Applied Systems Analysis  
[www.iiasa.ac.at](http://www.iiasa.ac.at)

# **The FAP Data Bank. Part 2: Updating and Aggregating - Methods and Practice**

**Sichra, U.**

**IIASA Working Paper**

**WP-84-094**

**December 1984**



Sichra, U. (1984) The FAP Data Bank. Part 2: Updating and Aggregating - Methods and Practice. IIASA Working Paper. WP-84-094 Copyright © 1984 by the author(s). <http://pure.iiasa.ac.at/2424/>

**Working Papers** on work of the International Institute for Applied Systems Analysis receive only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute, its National Member Organizations, or other organizations supporting the work. All rights reserved. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage. All copies must bear this notice and the full citation on the first page. For other purposes, to republish, to post on servers or to redistribute to lists, permission must be sought by contacting [repository@iiasa.ac.at](mailto:repository@iiasa.ac.at)

NOT FOR QUOTATION  
WITHOUT PERMISSION  
OF THE AUTHOR

**THE FAP DATA BANK  
PART 2: UPDATING AND AGGREGATING--  
METHODS AND PRACTICE**

U. Sichra

December 1984  
WP-84-94

*Working Papers* are interim reports on work of the International Institute for Applied Systems Analysis and have received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute or of its National Member Organizations.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS  
2361 Laxenburg, Austria

## FOREWORD

Understanding the nature and dimensions of the world food problem and the policies available to alleviate it has been the focal point of the IIASA Food and Agriculture Program (FAP) since it began in 1977.

National food systems are highly interdependent, and yet the major policy options exist at the national level. Therefore, to explore these options, it is necessary both to develop policy models for national economies and to link them together by trade and capital transfers. Over the years FAP has, with the help of a network of collaborating institutions, developed and linked national policy models of twenty countries, which together account for nearly 80 percent of important agricultural attributes such as area, production, population, exports, imports and so on. The remaining countries are represented by 14 somewhat simpler models of groups of countries.

To support the work a data bank was organized at the very beginning of the Food and Agriculture Program. The FAP data bank has grown in size and complexity and now contains large volumes of data obtained from different sources.

Ulrike Sichra has described the updating and aggregation methods in this paper. The organization, contents and management of the data bank are described in an accompanying paper, namely WP-84-93.

Kirit S. Parikh  
Program Leader  
Food and Agriculture Program.

## PREFACE

The FAP Data Bank is a large collection of data from different sources which constitutes a basic element in the modeling activities of the Food and Agriculture Program (FAP). This data bank was created at the very beginning of the Food and Agriculture Program and has grown ever since in size and complexity. In order to be able to describe it better and to document its contents, the vast amount of information has been split into two parts:

Part 1: Organization, Contents and Management

Part 2: Updating and Aggregating -- Methods and Practice

Part 1 is the introductory paper on which Part 2 is based. It addresses a general audience, interested in data for agricultural modeling, serving at the same time as a document for the FAP modeling activities.

Part 2, the current document, is designed for those who will take care of updating of the data bank. This volume not only assumes that the reader is familiar with Part 1, but also that he or she are experienced computer users, preferably at IIASA.

An important document to understand the idea of "aggregation", a crucial concept in the FAP Data Bank is *The Aggregation of the Agricultural Supply Utilization Accounts*, WP-83-42, IIASA. In that paper the methodology and details of the aggregations are described at length.

It is hoped that the two papers describing the data bank will satisfy a long felt need for documentation and clarification. Any suggestions for improving the understandability of the data bank description are very much welcomed.

## ACKNOWLEDGEMENTS

The nature of this paper makes it impossible to list all the persons and organizations that helped towards its publication. The main data contributions came from the following institutions:

- The Food and Agriculture Organization of the United Nations (FAO), Rome, Italy
- The International Labour Organization of the United Nations (ILO), Geneva, Switzerland; and
- The World Bank, Washington, D.C., USA

To these organizations, the Food and Agriculture Program (FAP) is deeply indebted, recognizing that without their unselfish collaboration the FAP Data Bank would hardly have come into existence. Most of the past and present staff of the FAP have been helpful in one way or another in creating the data bank. Many suggestions from both Program Leaders, Ferenc Rabar and Kirit S. Parikh, have contributed to the usefulness of the data bank.

Numerous persons in the FAP collaborating network have made available new data for their country, or have updated the existing data for it. Our deep gratitude is addressed to them. Without the dedication of Günther Fischer the data bank and its managing routines would not have evolved. Bozena Lopuch and Stefanie Hoffmann worked with great dedication on the CMEA and fertilizer data. The formatting efforts of Lilo Roggenland and Bonnie Riley can be directly seen. Without the careful reviewing done by Gerhard Krömer and Laszlo Zeold many parts would not have been clear.

And last but not least, we wish to thank all the users of the FAP data bank who by using the data, and with their questions, discovery of errors and similar actions have helped the FAP Data Bank to become a useful instrument in the modeling activities of the FAP.

## CONTENTS

1. Introduction	1
2. Convert EBCDIC Tapes to Binary FAP Tapes	2
3. Extract Time Series Needed Later	3
4. Check Beginning Years and Errors	3
5. Preparations for the Aggregation	5
5.1. Price.a - binary data file	5
5.2. In.rates - formatted file	5
5.3. Exch.all - binary file	5
5.4. In.prices - formatted file	5
5.5. In.trans. - formatted file	5
5.6. Wei.conv - formatted file	6
6. SUA Aggregations	6
6.1. Aggregate Original FAO Commodities to Main Commodities	6
6.2. Convert to Other Units of Measurement	7
6.3. Aggregation to the Large FAP Commodity List	8
6.3.1. Exceptions	8
6.3.2. Special Classification for New Zealand	8
6.4. Aggregation to the Small FAP Commodity List	10
6.4.1. Exceptions	11
7. The 10th Commodity	12
7.1. National	12
7.2. International	12
8. Prices	13
8.1. National Average Producer Prices as Reported by FAO	13
8.1.1. PGEN1 - Prices of "NES" Products	14
8.1.2. PGEN2 - Price of Sugar Refined	14
8.1.3. PGEN3 - Oil and Cake Price from Price of Crop	15
8.1.4. PGEN4 - Other Missing Prices	15
8.2. National Prices for Different Items of the Large and Small FAP Commodity Lists	15
8.2.1. Exceptions	18

8.3. World Market Prices	18
9. Aggregations of Several Countries to "one" Country	18
9.1. "World" Data	19
9.2. EEC SUA's and Macro Data	20
9.3. CMEA SUA's and Macro Data	21
10. Nutrient Contents of Agricultural Products	22
11. Reduce Prices and Quantities	24
11.1 Exceptions	24
12. Population Data	25
References	25

### Tables

Table 1: Commodities of the Detailed FAP List and their Dimensions	9
Table 2: Commodities for New Zealand	10
Table 3: Small and Detailed FAP Commodity List, Mapping	11
Table 4: Feed Factors	17
Table 5: Combination of Original and Aggregated Prices for World Price Calculations	19

### Appendices

Appendix 1: Data File Format	26
Appendix 2: Nutritive Factors Record	27
Appendix 3: Commodity Codes and Text	28
Appendix 4: FAP Countries	35



# THE FAP DATA BANK

## PART 2: UPDATING AND AGGREGATING METHODS AND PRACTICE

*U. Sichra*

### 1. Introduction

At irregular time intervals the Statistics Division of FAO, Rome, produces updates of existing Supply Utilisation Accounts (SUA) series, or computes further years of that time series. FAP receives updates on magnetic tape, if requested. Such updates are seldom received without prior request. On such occasions some written documentation is also made available, such as country and commodity lists, commodity trees, etc.

There is a fairly strict sequence of programs to be executed in order to integrate the SUA into the FAP Data Bank and to produce the aggregates needed for the modelling work within FAP. However, there are a number of steps that must be taken, as they unfortunately cannot be automatized. The updating and aggregating itself is also not a "push button" action for several reasons which will be explained later. Among others, the disk space on the VAX (IIASA) is in general not large enough, and the FORTRAN77 used there does not permit direct writing of unformatted records to tape.

This paper will only cover the updating and aggregation logic of the SUA and related time series, like prices, and population aggregations to country groups. Miscellaneous programs like *merge*, *extract*, etc., will not be discussed here. The reader is referred for these activities to WP-84-93, (U. Sichra, 1984) In short, this is the sequence of actions to be taken in order to update old data records, or make new time series for the FAP Data Bank.

- Receive new EBCDIC tape(s) from FAO
- Convert EBCDIC tape(s) to binary tape(s).
- Select FAP countries and put them on one tape.
- Compare old and new accounts, make corrections and adjustments if necessary. Merge series if wanted and/or needed.
- Calculate missing producer prices (PGEN). Merge with old prices.
- Calculate average import and export prices (price.a). Merge with old prices.
- Aggregate 600 FAO commodities to 260 main commodities (ag).
- Convert main commodities to other units (agcv).

- Aggregate 260 commodities to 27 FAP commodities (ag27).
- Aggregate 27 FAP commodities to 16 FAP commodities (ag9).
- Calculate prices for 27 and 16 commodities (vavo27 and vavo9).
- Merge SUA and prices for 27 and 16 commodities.
- Calculate the 10th price, merge to ag27 and ag9.
- Reduce prices (quantities) to 19 and 10 commodities.
- Aggregate all stages to EEC (ag, agcv, ag27 and ag9), also prices vavo27 and vavo9.
- Aggregate all stages to "one" (ag, agcv, ag27 and ag9).
- Make world market price for 10th commodity.
- Make world market prices for 27 and 16 commodities.
- Reduce world market prices to 19 and 10 commodities.
- Calculate population time series.

In the following sections each step will be explained in more detail, pointing out possible difficulties which might be encountered.

## 2. Convert EBCDIC Tapes to Binary FAP Tapes

The standard format of the SUA tapes provided by FAO is:

1600 bpi (sometimes 6250 bpi)  
EBCDIC, no label  
220 characters/record  
4400 characters/block (=20 records/block)

Each record corresponds to one data record, whose logical format can be seen in Appendix 1. For any further processing it is best not to keep the data in EBCDIC format, as this would need constant converting to ASCII or internal machine code in order to be able to process the data. A sequential, unformatted, binary storage mode has been chosen, which saves space and speeds up processing (input and output of records). On the EBCDIC tape one data record (7 codes and 16 years of data) has 220 bytes, the converted records have 116 bytes each (108 bytes of codes and data, 8 bytes of information for the storage mode of the record).

In general data tape(s) from FAO contain very large amounts of data, which on the VAX do not fit on disk at once. Therefore it is necessary to convert the EBCDIC tape to binary FAP format by writing directly to tape. The option to do this depends on:

- availability of 2 tape drives simultaneously
- a FORTRAN compiler which generates a code that can write unformatted binary records directly to tape.

Currently both requirements are met at IIASA only by the PDP 11/70. Therefore the data conversion has to be done there. The sequence of compilation and execution is:

```
ftn supbin.f -ls -lv  
dd if=/dev/rmt[0,1] ibs=4400 |a.out 6=/dev/rmt[1,0] 7=checkfile
```

The following problems have been encountered so far when doing this:

- all data from one FAO tape do not fit on one IIASA tape;
- FAO data are stored on more than one tape;
- the tape density is 6250 bpi.

If the density is wrong (i.e. 6250) there are 2 options: send the tape back to FAO and ask them to make one with the correct density; or have it converted to 1600 or 800 bpi at another computer installation (TU, IBM, etc). For the other 2 problems there are no standard solutions. The general idea is to fit as much data as possible on one IIASA tape, but always ensuring that every country is complete on the tape (this feature is required by the aggregation programs). One can use disk storage for pieces of countries, and then merge files from disk and tape to another tape. It is very easy to adapt the program suabin.f such that it checks for a specific country code and starts writing from there, or only until there, etc. The solution depends always on the concrete situation.

Binary data generated by programs compiled with the compiler ftn on the PDP will not be readable by programs compiled with f77 or xf77 on the VAX. It is therefore necessary to again convert the binary data to a slightly different format. The compilation and execution of this program are:

```
ftn vax.f -lv -ls  
and  
a.out 1=/dev/rmt[0,1] |dd of=/dev/rmt[1,0]
```

This routine has not been built into the EBCDIC-binary conversion (supbin.f) as the possibility of checking results on the PDP should be retained. It also has proven useful a number of times to have readable SUA on the PDP, as extractions, listing, merging, correction, etc. can all be done there as well.

### 3. Extract Time Series Needed Later

The original SUA from FAO can have the following types of time series merged with them, in addition to the SUAs;

- population (commodity code = 1)
- producer prices (dimension = 3)
- total trade (commodity code = 10)
- supply and demand of agricultural products measured in calories, protein and fat (3 digit element code).

It is best to first extract each of these record types to separate files and to keep only SUA records on one file or tape.

### 4. Check Beginning Years and Errors

It is essential for most programs that all time series in one file have the same starting year (icd(6)). (For details on the organisation of the data and the meaning of the codes see Sichra, WP-84-93.) It can happen though, that time

series of some countries and commodities start at different years. In such cases it is best to write a small program which:

- reads records,
- checks for the correct year
- writes if correct, or
- shifts and writes

The program avail.f generates a long table of all countries and commodities and their beginning year. This check should also be applied to producer prices and population time series. A number of miscellaneous errors could still be found in the accounts, but in general they will only show when processing the data further (aggregate, make prices, merge, etc). Some types of errors which have been found up to now are:

- data for single years are mistyped by n magnitudes of 10;
- data is shifted, zeros are found in the middle of time series where not expected;
- unknown commodity numbers, which can also mean that a new commodity has been introduced.
- unknow elements and/or dimension, which can also indicate a new item, or a change in the logic, or simply an error.

These errors have to be corrected individually. In general it also proves useful to run the program "compare.f" or "compare1.f" which tells the difference of 2 overlapping time series in a specified period. The difference is reported in absolute and relative values. From the control output decisions can be made as to where and how to adjust the new data, especially in the connecting years.

In cases where new countries, commodities, elements and/or dimensions have been introduced, and should be included in the FAP Data Bank, care must be taken to update the files which provide text for the data listings. These are:

- Countries: The country text has to be filled with its code into the file nfao.2.
- Commodities: The commodity code, together with the group code, has to be filled into the file nfao.3.1, if it is an original FAO commodity, or should be treated as such in the listings. If it is a purely FAP commodity, it should be put into the file nfao.3.22.
- Elements: Any new elements have to be checked for matching into the pattern set (see file list.1 or list.22) in the same directory as nfao.2 and nfao.3.1 (nfao.3.22). The programs which pick up element (and dimension texts) read from the random access file bin.1 (or bin.22). This file will have to be created anew with the program def.f.
- Dimensions: The same procedure as for new elements applies here. There is only provision for 3 columns of dimension texts (i.e. 3 dimensions), although in the data file the entry for icd(5) can be any integer number.

All country, commodity, element and dimension listings are already presented in the first part of the Data Bank description, see Sichra, WP-84-93.

## **5. Preparations for the Aggregation**

In order to successfully aggregate the original SUA to main commodities, and later to 27 and 16 commodities, a number of data and parameter files have to be prepared. They may change as the accounts change, and will need to be checked, updated, corrected or remade. These files are:

- price.a (average export and import prices)
- in.rates (default extraction rates)
- exch.all (exchange rates from national currency to US\$)
- in.prices (average export prices of oilseeds, and cakes)
- in.trans and in.transx (commodity trees and exceptions)
- wei.conv (aggregation pattern and weights)

### **5.1. Price.a - binary data file**

This file is created by the program alunitp.f. If all countries over which the average will be made do not fit on one tape, the program has to be adapted to be able to read from 2 different input files sequentially.

### **5.2. In.rates - formatted file**

In.rates can be made "manually" by choosing suitable extraction rates for all commodities which partake in the aggregation. But if the number of countries and commodities involved is too large, the programs exratr.f and exav.f will calculate the rates by simply averaging over all commodities. If all time series involved do not cover the same number of years it could be best to modify the averages of those years, for which all countries do not report. This does not cause problems for the time series, as the extraction rates hardly change with time.

### **5.3. Exch.all - binary file**

When calculating prices of 27 and 16 commodities exchange rates are needed to convert US\$ to national currencies. These rates can be read from the macro data file (which also needs updating from time to time), or a separate file with exchange rates can be created. The format is the standard binary SUA records format, the exchange rates are expressed in nc/10000 US\$ (FAO style).

### **5.4. In.prices - formatted file**

In this file the average export prices for oil crops, oils and cakes are stored which will be used for splitting the imports and exports of oil crops in the main aggregation program. The information for this file can be taken from the file price.a.

### **5.5. In.trans - formatted file**

The structure of in.trans (and in.transx) is explained in detail in Fischer and Sichra, WP-83-42. In general these files don't need to be changed when new aggregations have to be done, as the files have information on structures, and not time series. It is possible, however, that tree structures of commodities in all or only some countries change, and this must be reflected in in.trans (and in.transx). Also, when new commodities are introduced, they have to be

included in this file. Both files, in.trans and in.transx have the same structure and formats. In.transx contains only those trees which are an exception for a certain country. The first 3 digits in every line specify the country for which the exceptions apply. The file has to be arranged by country code and within it by commodity code.

### 5.6. Wei.conv - formatted file

Similar remarks as for in.trans can be made for wei.conv. It does not change with time (no time series), but may change as updates of FAO can be different in terms of newly introduced commodities. The details about this file are also to be found in Fischer and Sichra, WP-83-42. There are separate wei.conv files for special aggregations.

## 6. SUA Aggregations

The aggregation of the SUAs has been explained in full detail in Fischer and Sichra, WP-83-42. This section concentrates mainly on the technical aspect (programs and files) and the exceptions which are active since the above reference has been completed.

The logical sequence of the aggregations is as follows:

input	program	output
original SUA (on tape 1)	agsua6.f	ag (to tape 2)
original SUA (on tape 1)	convsua.f	agcv (to tape 2)
agcv (on tape 2)	ag27.f	ag27 (to disk)
ag27 (on disk)	ag9.f	ag9 (to disk)

This sequence will only be kept if the aggregation is done country-wise and there are no exceptions and/or errors in the data. It can happen though, that some data are only corrected at the ag, or ag27 level, which means that intermediate manipulation will take place.

Care should be taken not to fill the disks extensively (not even /tmp). On the other hand, the output cannot be written directly to tape. Thus the results should be put on tape as they originate, taking care not to destroy or overwrite tapes (use /dev/smt and scan the tapes frequently).

### 6.1. Aggregate Original FAO Commodities to Main Commodities

On the tapes received from FAO there are accounts for about 600 commodities (260 main commodities and 340 derived commodities) for each country (see Appendix 3). Currently there is data for about 110 countries at IIASA, but only the FAP4 countries have been aggregated and checked (see Appendix 4).

After executing the main aggregation program agsua6.f only the main commodities are left (about 260). They have the same commodity codes as the original commodities (e.g. 15=wheat, 27=rice, etc), although the time series for most elements will look different now. For the accounts starting in 1966 (latest release from FAO available at FAP) some commodity trees look different from the old ones starting in 1961. There is a major change in the cotton tree.

In the old tree, and therefore in the aggregation program:

329	cotton seed is aggregated downwards to 331 and 332
331	oil of cotton seed and
332	cake of cotton seed remain as such
766	seed cotton remains
767	cotton lint is aggregated upwards to 766
768	cotton carded and
769	cotton waste are disregarded
776	linter remains

In the new accounts the code for seed cotton (766) has been changed to 328, therefore the tree would have to be changed. But in order to aggregate 767 up to 328 all would have to be on one region card and the arrays in the program agsua6.f are not large enough for this. Thus the following quick solution has been adopted:

- The program agsua6.f has to be run twice: first on all commodities but cotton (328 and 767).
- As the next step, the commodities 328 and 767 (cotton) should be extracted from the original tape for all countries and agsua6.f run again, with the parameter and control files from the subdirectory exc (exception). This is necessary because the tree now seemingly spans over a large number of commodities and the program cannot load that large amount of international prices and extraction rates. In the subdirectory there are files with reduced number of commodities in them.
- The resulting commodity 328 will then have to be recoded to 766 and merged with the result from the first run of agsua6.f

## 6.2. Convert to Other Units of Measurement

In the second step of the main aggregation all quantities are written out again, weighted with the average 1969-1971 US\$ world market price (stored in wei.conv), and, if needed, for further aggregations, with another equivalence weight (e.g. protein content, meat weight, etc. also in wei.conv). Import and export values are no longer kept, but areas and stocks are taken over.

The commodity codes remain the same as in the original accounts, but the dimension codes change. In the original SUA and after agsua6.f, dimension 1 is always quantity (measured in number or mt), and dimension 2 is value in 1000US\$ (only used for imports and exports).

After executing convsua.f dimension 1 is quantity in 1970US\$ (which is always reported), and dimension 2 is quantity in mt (available where needed for further aggregations). There are exceptions to this rule: stocks and areas (sown and harvested) in general have dimension 1 (number or ha).

No manual interference with the results from convsua.f is expected, as all corrections should have taken place for the original data or after the first aggregation, or will take place later.

### **6.3. Aggregation to the Large FAP Commodity List**

For the detailed models of the Basic Linked System (BLS), which do not take into account country specific features, the program ag27.f generates the accounts used in the parameter estimations and the model simulations. The commodities that result from this aggregation start with code 3001. The first 18 commodities (3001 to 3018) are demand and supply of agricultural products, commodity 3019 is non-agriculture, and commodities 3020 to 3027 are again supply of agricultural commodities. The supply is divided into crop and animal part, which does not mean that in commodities 3020 to 3027 consumption items will not be found as well. This comment is made because the demand model only asks for 19 commodities, and prices have to be created at that level as well.

Table 1 below shows the commodity codes, their names, the demand commodity code to which it belongs, and the dimensions available.

As pointed out above, some commodities have 2 units of measurement. If possible and available, dimension 2 (mt) will be taken for modelling activities, but for some very aggregate commodities like vegetables and roots, cocoa and tea, etc, only dimension 1 (1970US\$ quantities) are available and make sense. Beverages of alcohol are reported here, but are not carried forward to the next aggregation, as they are also included in the supply and demand of coarse grain and fruits, and their inclusion would mean double counting.

#### **6.3.1. Exceptions**

There are special weight files (wei.conv) and programs to make aggregations at this level for Kenya (wei.conv.114), for some feed programs (wei.conv.kl) and for New Zealand and Australia (wei.conv.aus). The resulting data files match special features of those models and parameter estimations. As a general rule though, modellers are discouraged to request their own aggregation patterns as the workload for the "databanker" would become very big, and a constant mapping from one aggregation pattern to another would be needed. The coding of these exceptions is also shown in Table 1.

The files for Kenya can be found in the directory "special", the files for the feed programs in the directory "klaus", those for New Zealand and Australia in the directory "aus". Some programs, however, already include those special features (can be seen in the comment lines of the programs).

#### **6.3.2. Special Classification for New Zealand.**

In order to take into account the special characteristics of New Zealand's agriculture, and the availability of data there, a somewhat different aggregation has been made for that country. This exception has been isolated from the others as the aggregation pattern is different, and only one FAP commodity list is created here. The special pattern is shown in Table 2. The programs and control file to handle it are stored in a special directory called "rae" (named after the person who worked on it), and/or have the string "rae" in the name.

It is important to remember that this aggregation is unique in that there is no reduction (see later) and no small commodity list either. Most of the commodities are also not suitably comparable with those resulting from other aggregations.



**Table 1: Commodities of the Detailed FAP List and their Dimensions.**

commodity codes - 3000...

supply	commodity demand		"good dimension"
1		wheat	(1)1000\$ (2)mt
2		rice	(1)1000\$ (2)mt(milled)
3		coarse grain	(1)1000\$ (2)mt
4		vegetable oil	(1)1000\$ (2)mt
5		protein feed	(1)1000\$ (2)mt pr
6		sugar	(1)1000\$ (2)mt
7		bov+ov meat	(1)1000\$ (2)mt
8		pork	(1)1000\$ (2)mt
9		poultry+eggs	(1)1000\$ (2)mt pr
10		dairy prod	(1)1000\$ (2)mt
11		veget+roots	(1)1000\$
12		fruits+nuts	(1)1000\$
13		fishery prod	(1)1000\$ (2)mt pr
14		coffee	(1)1000\$ (2)mt
15		cocoa+tea	(1)1000\$
16		bev.of alcoh	(1)1000\$
17		fibres	(1)1000\$
18		industr.crops	(1)1000\$
19		non agricult	(1)1000\$
20	4	bov+ov fat	(1)1000\$ (2)mt
21	4	pig fat	(1)1000\$ (2)mt
22	4	poultry fat	(1)1000\$ (2)mt
23	4	fish oil	(1)1000\$ (2)mt
24	5	meat meal	(1)1000\$ (2)mt pr
25	5	fish meal	(1)1000\$ (2)mt pr
26	17	wool,hides	(1)1000\$
27	17	pig hides	(1)1000\$

Exceptions for the Feed Programs:  
commodity codes: 3700...

28		potatoes	(1)1000\$
29		cassava	(1)1000\$
30		eggs	(1)1000\$ (2)m tpr

Exceptions for the New Zealand and Australia:  
commodity codes: 3400...

28		ovine meat	(1)1000\$ (2)mt
29		wool	(1)1000\$
30		ovine fat	(1)1000\$

**Table 2: Commodities for NZ**  
**Rae Aggregation, weight file: wei.conv.rae**

	commodity	dimension
32..		
1	wheat	2 (mt)
2	rice	2 (mt milled)
3	coarse grain	2 (mt)
4	vegetable oil	2 (mt)
5	prot.feed	2 (mt)
6	sugar	2 (mt)
7	bovine meat	2 (mt)
8	---	
9	pig+poultry+egg	2 (mt)
10	milk	2 (mt)
11	---	
12	veg+roots+fru+nuts	1 (1000 \$70)
13	fish	2 (mt)
14	---	
15	coffee+cocoa+tea	1 (1000 \$70)
16	dist.alcoh	1 (1000 \$70)
17	fibres	1 (1000 \$70)
18	ind.crops	1 (1000 \$70)
19	offals cattle	2 (mt)
20	offals sheep	2 (mt)
21	pig fat	2 (mt)
22	poultry fat	2 (mt)
23	fish oil	2 (mt)
24	meat meal	2 (mt)
25	fish meal	2 (mt)
26	silk+hides cattle+ovine	1 (1000 \$70)
27	pig skin	1 (1000 \$70)
28	ovine meat	2 (mt)
29	stearine, degreas+tallow	2 (mt)
30	wool	1 (1000 \$70)

#### 6.4. Aggregation to the Small FAP Commodity List

The last step in the SUA aggregations is to generate the 16 commodities which are used in the simplified FAP models. Although all related programs have the suffix 9 (ag9.f vavo9.f etc.) there are 16 commodities in the simplified commodity list, which follows the same logic as the previous aggregation. The 16 commodity files will always have the suffix 9; this has historical reasons (within FAP) and will therefore not be changed.

There are 9 agricultural demand and supply commodities (3501 to 3509), one non agricultural commodity (3510) which is identical to 3019 above, and 6 more supply commodities (3511 to 3516). This time there is only one quantity measure, dimension 2 (mt) or 1 (1970US\$) depending on the commodity. Animal stocks and area are always reported with dimension 1 (as for ag27). In Table 3 one can see how the mapping from 27 to 16 commodities is done, which are the dimensions in ag9, and to which demand commodities the supply 3511 to 3516

will be aggregated to. Also for ag9 prices have to be created at the 10 commodity level.

**Table 3. Small and Detailed FAP Commodity List, Mapping**

commodity SO..	detailed models 3001 - 3027		35	simplified models 3501 - 3516	
		"good dim"			"good dim"
1	wheat	(2)mt	1	wheat	(2)mt
2	rice	(2)mtm	2	rice	(2)mt milled
3	coarse grain	(2)mt	3	coarse grain	(2)mt
4	vegetable oil	(2)mt	8	other food	(1)1000\$70
5	protein feed	(2)mt pr	7	protein feed	(2)mt prot
6	sugar	(2)mt	8	other food	(1)1000\$70
7	bov+ov meat	(2)mt	4	bov+ov meat	(2)mt
8	pork	(2)mt	6	other meat	(2)mt protein
9	poultry+eggs	(2)mt pr	6	other meat	(2)mt protein
10	dairy prod	(2)mt	5	dairy prod	(2)mt milk
11	veget+roots	(1)1000\$	8	other food	(1)1000\$70
12	fruits+nuts	(1)1000\$	8	other food	(1)1000\$70
13	fishery prod	(2)mt pr	6	other meat	(2)mt protein
14	coffee	(2)mt	8	other food	(1)1000\$70
15	cocoa+tea	(1)1000\$	8	other food	(1)1000\$70
16	bev.of alcoh	(1)1000\$	8	other food	(1)1000\$70
17	fibres	(1)1000\$	9	industr.crops	(1)1000\$70
18	industr.crops	(1)1000\$	9	industr.crops	(1)1000\$70
19	non agricult	(1)1000\$	10	non agric	(1)1000\$70
reduction			reduction		
20	4	bov+ov fat (2)mt	11	8	bov+ov fat (1)1000\$70
21	4	pig fat (2)mt	12	8	other fat (1)1000\$70
22	4	poultry fat (2)mt	12	8	other fat (1)1000\$70
23	4	fish oil (2)mt	12	8	other fat (1)1000\$70
24	5	meat meal (2)mt pr	13	7	meat meal (2)mt protein
25	5	fish meal (2)mt pr	14	7	fish meal (2)mt protein
26	17	wool,hides (1)1000\$	15	9	wool,hides (1)1000\$70
27	17	pig hides (1)1000\$	16	9	pig hides (1)1000\$70
<b>Exceptions for the Feed Programs:</b>					
28		potatoes (1)1000\$70	17		potatoes (1)1000\$70
29		cassava (1)1000\$70	18		cassava (1)1000\$70
30		eggs (2)mt protein	19		eggs (2)mt protein
<b>Exceptions for the New Zealand and Australia:</b>					
28		ovine meat (2)mt	17		ovine meat (2)mt
29		wool (1)1000\$70	18		wool (1)1000\$70
30		ovine fat (1)1000\$70	19		ovine fat (1)1000\$70

#### 6.4.1. Exceptions

For the feed programs and the New Zealand and Australia country models there is also a special program which makes the aggregation according to the special pattern to ag9. The remarks about the exceptions at the 27 commodity level also apply here.

## 7. The 10th commodity

Both the small (ag9) and the large (ag27) FAP commodity lists have a non-agriculture commodity, for which only production and producer price are calculated.

### 7.1. National

The program y10cal.f generates production and producer prices for non agriculture at the 16 commodity level, which will have to be merged into the corresponding file. The non-agriculture production and price at the 27 commodity level can be generated by recoding the output of y10cal.f to 3019 (instead of the original 3510).

The quantity (in 1970US\$) of the non-agriculture commodity (3019 and 3510) is computed by dividing the constant 1970 non-agriculture GDP by the exchange rate for 1970:

$$q(t) = \text{gdpna70}(t) / \text{exchr}(70)$$

and the price is the ratio from non-agriculture current GDP to non-agriculture constant 1970 GDP, multiplied by the exchange rate for 1970:

$$p(it) = \text{gdpna}(t) / \text{gdpna70}(t) * \text{exchr}(70)$$

### 7.2. International

There is also a world market price required for the 10th commodity, which results from executing the program nonagr.f. From the world tables and other sources one can get data on total exports in current and constant 1970 prices per country, in national currency. From the next section it will be clear how to compute world market prices for the agricultural products at the 16 and 27 commodity level.

With the help of exchange rates the world market price for the 10th commodity is calculated using the following formula:

$$pwe_t(10) = \frac{\sum_{j=1}^{C_i} \text{EXT}_{j,t} / \text{exch}_{j,t} - \sum_{i=1}^{N_i} \left[ \sum_{j=1}^{C_i} \text{EXA}_{j,t}(i) \right] * pwe_t(i)}{\sum_{j=1}^{C_i} \text{EXT70}_{j,t} / \text{exch}_{j,70} - \sum_{i=1}^{N_i} \left[ \sum_{j=1}^{C_i} \text{EXA}_{j,t}(i) \right] pwe_{70}(i)} * 10^8$$

where

$pwe_t(10)$	= world price for the non agriculture, in year t
$\text{EXT}_{j,t}$	= total exports, country j, at current value, year t
$\text{EXT70}_{j,t}$	= total exports, country j, at constant 70 value, in year t
$\text{exch}_{j,t}$	= exchange rate, country j, year t in nc/US\$

$exch_{j,70}$	= exchange rate, country j, year 1970 in nc/US\$
$EXA_{j,t}(i)$	= agriculture exports volume, country j, commodity i, year t
$pwe_t(i)$	= world price for commodity i, year t
$pwe_{70}(i)$	= world price for commodity i, year 1970
$C_j$	= number of countries which aggregate to "world" (FAP countries)
$N_i$	= number of agriculture commodities.

i.e. the difference of current total exports and agricultural exports, divided by the difference of constant 1970 total exports and agricultural exports.

The given agricultural exports have the dimension 1 or 2 (mt or 1970US\$) depending on the commodity. The factors  $10^{**n}$  are required because of the storage mode of the exchange rates.

## **B. Prices**

There are prices of agricultural products at four levels of aggregation: original, ag, ag27 and ag9 and four elements: production, feed, food and other utilisation. For the first 2 groups of SUA (600 original and 260 aggregated commodities) only producer prices are available, and they are identical for both, as the production of main commodities does not change from original SUA to ag. They come directly from FAO, on the SUA tapes, merged with the other elements and dimensions of the corresponding commodity.

### **B.1. National Average Producer Prices as Reported by FAO**

FAO reports on producer prices in their publications, which are actual producer prices supplied by official sources. There are as well computerized time series on average producer prices which serve the purpose of evaluating the agricultural production in terms of current value. In what follows the average producer prices shall be discussed. The sources of these prices are multiple, they can be, by order of preference:

1. Official publications
2. Answers by national authorities to FAO questionnaires or direct queries
3. Other national publications
4. International publications
5. Unofficial publications and documents
6. Estimates using different techniques.

For more information on the estimates and the adjustments applied to certain commodities the Statistics Division of FAO should be consulted directly. The computerized time series presented by FAO contain prices for the main commodities, and sometimes for their derivatives as well. Ideally the time series are complete, for all years of coverage (16 years in the current design of the SUAs). But in some cases data is missing for beginning and/or ending years. For the following countries the original producer price has to be divided by 100 (by using the program c100.f):

10	Australia
33	Canada
54	Denmark
104	Ireland
229	UK
231	USA

For Egypt (59) the divisor has to be 1000.

In the case of FAP, it is problematic if the prices of some derived products are missing, like oils and cakes of oil crops, sugar refined, offals and fats of animals. The FAP commodity selection treats these products as main commodities and therefore a producer price is needed when calculating the aggregated producer prices (detailed and small FAP lists).

In order to solve these problems (missing years and missing commodities) 4 computer programs have been written which calculate the needed time series of producer prices. After running these programs the results have to be merged with the original FAO prices in order to have the complete set of producer prices for original data and first stage of aggregation (ag).

#### 8.1.1. PGEN1 - Prices of 'NES' Products

A number of commodities are a basket of various products, which all together receive the name xxx-NES (e.g. tropical fruits NES). In general there is no price available for these commodities, and PGEN1 will generate it.

The price of a NES-product is the average price of similar products:

$$p\text{-nes} = \sum (p_1 + p_2 + p_3 + \dots + p_n) / n$$

In a similar way, if a price of the similar products is missing, the NES price is taken instead.

#### 8.1.2. PGEN2 - Price of Sugar Refined

Quite frequently the price for sugar refined is not reported, rather the price of sugar beets and sugar cane. From these two the sugar price is calculated, as a function of prices of beets and cane, quantities of beets and cane and extraction rates:

$$ps = (sc*pc + sb*pb) / (erc*sc + erb*sb) / ers$$

where:

ps:	sugar price
pc:	sugar cane price
pb:	sugar beet price
sc:	sugar cane production
sb:	sugar beet production
erc:	sugar cane extraction rate
erb:	sugar beet extraction rate
ers:	sugar raw extraction rate

### 8.1.3. PGEN3 - Oil and Cake Price from Price of Crop

Similar to the sugar problem, prices for oils and cakes of oil crops have to be generated from the main product, following the formula:

$$p(i) = p_{crop} / 2 * \text{extr}(i)$$

where:

- $p(i)$  = price of oil(1) or cake(2)
- $p_{crop}$  = price of crop
- $\text{extr}(i)$  = extraction rate of oil(2) or cake(2)

In this program beer prices are computed in an identical way.

### 8.1.4. PGEN4 - Other Missing Prices

For a number of products there is no producer price available, and the method devised is to calculate the price by applying extraction rates to existing prices, or averaging over other prices. Typically products for which this method applies are meat, offals and fats of animals, as well as hair, hides and skins.

After having calculated missing prices (some years, or the whole time series) by the above methods, it is advisable to compare the results with existing FAO average producer prices. It has been observed quite frequently, that although the method applied by FAP seems to be similar to the one applied by FAO, there is a nearly constant ratio between FAP and FAO prices. In order to remain consistent with FAO, when the discrepancies have been large (i.e. >5%), the ratio has been applied to the FAP prices and the missing price only added to the FAO prices. For this, the meaningful program "MESSAGE" has been used.

## 8.2. National Prices for Different Items of the Large and Small FAP Commodity Lists

For the detailed and the small FAP commodity classification an aggregation procedure has been devised, in order to generate national prices for production, feed, food and other utilisation of agricultural products.

The following aggregation formula has been used:

$$p(i,k) = \frac{\sum_{j=1}^{N_i} pp(j,k) * q(j,k)}{\sum_{j=1}^{N_i} q(j,k)}$$

where

- $p(i,k)$  = national price of aggregate commodity i, element k (k = production, feed, food or other utilization)
- $pp(j,k)$  = producer price from FAO for commodity j (or substitute, if not available), for element k (k=production, feed, food or other utilization)

- $q(j,k)$  = quantity of commodity j, element k  
 $N_i$  = number of commodities j that aggregate to commodity i

The substitute price  $pp(j,k)$  which is taken in case of a non available producer price for commodity j is calculated differently depending on the element k it is needed for. The following rules apply:

[if applicable, the first of the prices below is taken, otherwise the second, in the worst case the third (which always exists)].

#### National Producer Price (k=5)

1.  $pp$  = national producer price as reported by FAO
2. if  $q_{exp}(j) \geq 0.05 * q_{prod}(j)$  for all years:  
 $pp$  = national export price, i.e.  $val_{exp}(j) / q_{exp}(j) * exchr$
3.  $pp$  = international export price, i.e.

$$\sum_{k=1}^{N_k} val_{exp}(j,k) / \sum_{k=1}^{N_k} q_{exp}(j,k) * exchr$$

#### National Feed, Food and Other Utilisation Prices (k=10,14,15)

1. if  $q_{prod}(j) \geq 0.4 * (q_{imp}(j) + q_{prod}(j))$  for all years:  
 $pp$  = national producer price as reported by FAO
2.  $pp$  = national import price, i.e.  
 $val_{imp}(j) / q_{imp}(j) * exchr$
3.  $pp$  = international export price

Depending on the commodity j, and the target commodity i, to which they are aggregated, some feed prices (k=10) are multiplied by the factors shown in Table 4.

The program which calculates these prices is called `newvalvol.f` (`valvol` is short for value per volume), and accordingly, the resulting price time series have the name `vavo27` and `vavo9` ("value per volume"). The program `newvalvol.f` reports about national producer prices not found or equal to zero. If the time series is not there at all, the next price is taken, according to the logic described above. If for a year the price equals to zero, it is up to the programmer how to proceed. In the SUA 1961-76 care was taken to have either complete time series for producer prices, or leave them out all together for those products where not enough data was available.

In the 1966-81 accounts there is the problem that some countries only report prices up to 1980, or 79, or even 78. The following solution was adopted here: for each country the "ending" year is known, and given as parameter in the control input (`in.natprices`). If prices of a commodity are missing before that year, the producer price is rejected. Any other solution, like interpolation or extrapolation can also be thought of, and the program changed accordingly.

Following the FAO coding practice, prices have dimension 3 (unit value). But at the 27 commodity level one runs into problems, as there are commodities with quantities in 2 dimensions, and consequently there should also be 2 price



**Table 4. Feed Factors**

---

3001	wheat	all	0.9
3002	rice	all	0.9
3003	coarse grain	all	1.0
		barley	0.95
3006	sugar	all	0.8
3007	bov+ov meat	all	0.3
3008	pork	all	0.25
3009	poultry and eggs	all	0.25
3010	dairy	all	0.25
3011	veg and roots	all	0.9
		potatoe	0.77
3012	fruits and nuts	all	0.5
3017	fibres	all	0.3
3020	bov+ov fat	all	0.6
3021	pig fat	all	0.6
3022	poultry fat	all	0.6
3023	fish oil	all	0.6

---

dimensions.

The following coding logic was adopted:

- If a commodity has only 1 quantity dimension (in such cases dimension 1, 1970US\$) the corresponding price has dimension 3 (unit value per 1970US\$).
- If a commodity has 2 quantity dimensions (dimension 1 = 1970US\$, dimension 2 = mt), the price for the first quantity measure (dimension 1) has dimension 4, the price for the second quantity measure (dimension 2) has dimension 3.

The reasons for this convention:

first: history.

second: the "good" price, which will be used in the model, always has dimension 3.

Care should be taken to always address the correct dimensions for the different commodities of the detailed FAP list. Although this might seem confusing to the reader, it becomes clear when dealing with the data. It is unfortunate, however, that the listing program does not provide for 4 dimensions and the text printed in the case of dimension 4 is "horrible".

So far only the prices at the 27 and 16 commodity level have been merged with the SUAs of the corresponding level and not the prices at the first aggregation level. At the same time the price files have also been kept separate. Aggregations to ag27 and ag9 have therefore been merged with vavo27 and vavo9 correspondingly, and then written to tape again.

### 8.2.1. Exceptions

For the pattern required for Kenya, for the feed program and for New Zealand and Australia the program `newvalvol.f` can also create the prices. A switch in this program will produce the correct output.

### 8.3. World Market Prices

The ideas behind the calculations of the world market prices can be found in Sichra, WP-84-95. Here only the mechanics will be explained. A number of programs have to be run to prepare files for the actual computation of world market prices. At first the "world" SUAs are needed at all 4 levels of aggregation, i.e.

<code>or.one</code>	original SUA
<code>ag.one</code>	main commodities
<code>ag27.one</code>	large FAP commodity list
<code>ag9.one</code>	small FAP commodity list

The creation of these files is explained in the next section. The smallest export price at the original level (no processing) and after the first aggregation (with processing) is calculated with the program `smallp.f`. A price will only be suitable to participate in the selection of the smallest, if the quantity exported is at least 3% of the "world" exports (`or.one` or `ag.one`). Once run over the original commodities, `smallp.f` gives raw material prices, and once run over the aggregated list, it produces a price with processing included.

As this program might take a long time and block a tape drive (especially when receiving a low priority), it has been written such that it only can take export elements (value and quantity). Thus these elements have to be extracted first to disk. If the program `smallp.f` has to be read directly from the full accounts, it will have to be changed accordingly. The next stage is to calculate world import and export prices for the small and the large FAP commodity lists. The programs which make this are called `onevavor.f` and `onevavoag.f`. The difference between them is only the treatment of rice as explained in the paper about world market prices, see Sichra, WP-84-95.

The last program which finally calculates the world market prices is called `worldp.f`. It selects the items with or without processing, according to the scheme shown in Table 5. The prices are then "reduced" to the 19 and 10 demand commodities correspondingly.

There is a shell file which first creates the world aggregates and then executes all programs in the correct sequence in order to generate the world market prices. This shell file is called `ww.run`. The price for the 10th commodity (non-agriculture) has then to be merged into this price file.

### 9. Aggregations of Several Countries to "one" Country

In a similar way one aggregates country-wise several commodities into one commodity by using appropriate weights, one can aggregate commodity wise several countries to make one country.

In general no weights are needed for this procedure, unless prices are calculated or values are not expressed in a common currency (e.g. in the macro data file).

Aggregations of several countries into one are used for making:

**Table 5. Combination of Original and Aggregated Prices for World Price Calculations**

commodity large FAP list	no	yes	reason
		proc	
wheat	x		
rice	x		but $p = q27 * p31 * 0.67 / q27 * 0.67$
coarse grain	x		
vegetable oil	x		
protein feed	x		
sugar		x	is a processed product
bov+ov meat		x	there is little trade in fresh meat, and the processing price also includes trade in offals
pork	x		
poul+eggs	x		
dairy prod		x	most trade is done in butter, milk powder and cheese
veg+roots		x	the raw material price is often higher than the price for the processed good
fruits+nuts	x		
fishery prod		x	the trade in fresh products is not representative
coffee	x		
cocoa+tea	x		
bev.of alcoh	x		
fibers		x	there is no trade in seed cotton, only linter, which has too low a price
indust.crops	x		
bov+ov fat	x		
pig fat		x	it is not traded as such, but as "lard" (derived commodity)
poultry fat		x	it is not traded as such but as "rendered"
fish oil	x		
meat meal	x		
fish meal	x		
wool + hides		x	the processed good is mainly traded
pig hides		x	the smaller price was chosen

- "world" (icd(2)=0) out of all FAP countries
- EEC (888) SUAs and macro data
- CMEA (777) SUAs and macro data

### 9.1. "World" Data

For the general one-country aggregation the program onectr.f can be used. It simply adds up, commodity-wise, over all countries of the input file, all elements and dimensions it finds.

The resulting output has to be sorted, as the time series might be out of

sequence. There are several sort programs which differ in the size of files they are able to sort (sort.f, sort1.f, sort2.f). During the sorting process creation dates and status indicator of time series are lost, but this is not a problem, as both items are not meaningful in a highly aggregated stage.

It is important to note that all countries of the input file will be lumped together in the program onectr.f. Aggregations over all FAP4 countries have the string "one" somewhere in the filename, and are treated as "world" data. Earlier "one" files, which were built over a somewhat larger country list, are therefore slightly larger.

## 9.2. EEC SUAs and Macro Data

Currently the following countries are aggregated to generate EEC data:

15	Belgium-Luxembourg
54	Denmark
68	France
78	Federal Republic of Germany
104	Ireland
106	Italy
150	Netherlands
229	UK

For this the corresponding countries have to be extracted first to a separate file, and this will then be the input for the program onectr.f. In order to aggregate the SUAs, including prices, to form one EEC "country" (code (888)), prices which are reported in national currencies, have to be converted to a common unit (EUROs in this case) and aggregated by weighting them with the quantities:

$$p(888) = \frac{\sum_{i=1}^{nctr} p(i) * \text{exch}(i) * q(i)}{\sum_{i=1}^{nctr} q(i)}$$

The program oneeec.f aggregates quantities (identical to onectr.f) and takes care of all types of prices (producer, food, feed, other). It uses an existing file of exchange rates from national currencies to EUROs (exch2.euro). The file exch2.euro is the output of the program excheuro2.f, which reads formatted conversion rates (stored in the file euro.form) and creates binary records in SUA format. For the time series starting in 1966 further exchange rates from national currencies to euros have been calculated, and the resulting binary file is "exch2.euro.66". The aggregations for this period are carried out in the same way as for the previous periods.

In order to aggregate further years of the SUA to EEC, it is necessary to update the exchange rates up to the last year of the SUAs, adapt the program excheuro2.f and create a new binary file "exch2.euro". For the exceptions (feed programs, NZ and Australia) the special aggregation patterns are accounted for in the program "oneeec.f" and are marked with comment lines. With an input parameter one can tell whether the aggregation should be done on the 27 or the 16 commodity level. There is no aggregation program ready made to aggregate prices at the 260 or even 600 commodity level. There has been no need for it so far, but it should not be difficult to change oneeec.f if needed, to perform that

job.

There are 2 more programs which aggregate macro data and fertilizer data of all member countries of EEC to one EEC-country: onectr.f and oferteec.f.

The program onectr.f does not do the macro-aggregation in the current version, but a few commands have to be changed and adapted, which should cause no problems to the user.

### 9.3. CMEA SUAs and Macro Data

The following seven countries have been aggregated to form the "country" CMEA with icd(2)=777:

27	Bulgaria
51	CSSR
77	Hungary
97	GDR
173	Poland
183	Romania
228	USSR

Additionally a further aggregate, with all the above countries except the USSR build a country group with icd(2)=776. There are SUA time series for both aggregates for the years 1961 to 1976 (with 1961 to 1965 according to the latest release of FAO), and from 1966 to 1981 only for the whole of CMEA (777).

Also some macro data is available for these aggregates (1961-1976). As the national accounting system is somewhat different from the one for market economies the commodity codes differ as well. They are:

commodity	element	dimension	explanation
1	14	1	labour force total in 1000
1	16	1	labour force agriculture in 1000
3110	2	1	fertilizer, nitrogen consumption in mt
3110	5	1	fertilizer, total consumption in mt
3112	1	1	capital stock (cst), total stock, in mil rubles
3112	2	1	capital stock (cst), agricultural capital, in mil rubles
3202	1	1	net material product, total, in mil rubles
3202	3	1	net material product, agriculture, in mil rubles

The above data can be found on files starting with the prefix "mcrec...", and "fert...". In order to calculate value aggregates, exchange rates from national currency to rubles are needed for all of the participating countries. The code for these is:

commodity	element	dimension	explanation
3113	13	1	exchange rate 100000nc per ruble

National producer prices for each of the CMEA countries only exist between 1965 and 1974. Due to the fact that a significant part of the modelling for CMEA countries is carried out with country specific models, and thus with special data from those countries, no further actions were taken in FAP to construct the CMEA aggregates for more years than the ones immediately available from FAO. The aggregates for CMEA are made with the program onectr.f, for all different levels of aggregation (or, ag, ag27 and ag9). The control file in.onectr provides necessary parameters. In the shell file onectr.run all files involved in the aggregation can be seen. The exchange rates required in the shell file are not relevant for the CMEA aggregations and can be neglected, however, a file must be specified.

Before running the aggregation the countries which form CMEA have to be extracted to a separate file. The aggregate file (after onectr.f) is not necessarily in the correct order (by commodity, within it by element, within it by dimension). Thus it has to be sorted, with the program sort2.f (large enough for the biggest commodity list). Note that sort2.f does *not* create a new file if a resorting is not needed. The resulting aggregations are only for quantities, in all quantity dimensions, and *not* for prices.

#### 10. Nutrient Contents of Agricultural Products

For most of the commodities included in the SUA there are figures available on the nutrient contents per unit of that commodity.

Together with the first SUA released to FAP a file with these values was files which contain the accounts (see Appendix 2). Also this file has been converted to binary format and it constitutes one of the files of the FAP data bank. In this case, however, the structure of the FAO file has not been taken over completely. In the FAP data bank this file needs the following read or write statement:

```
read(iu,end=991) (jcd(i),i=1,5),(y(j),j=1,3)
```

For FAP purposes only jcd(4) and jcd(5) (country and commodity code) are of interest.

The three factors stored in y are;

y(1)	calories	(nr/100 gr)
y(2)	protein	(gr/100 gr)
y(3)	fat	(gr/100 gr)

The file is organized in a similar way as the SUA files: by increasing country code, and within a country by increasing commodity code. It is important to remember that for country code 0 (international factor) there are conversion factors for every main commodity and for most derived products. For each individual country the reporting of conversion factors is not complete, and one might want to use the international factor (country code 0) when the needed values for a specific country and commodity cannot be found. The program foodvavo.f (discussed below) only uses the international factors for reasons of simplicity.

The file of nutritive factors from FAO has been split into 2 files at FAP:

nutc.bin.w, which contains the international factors (country code 0),  
and  
nut.bin.n, which covers all other countries.

The first file (nutc.bin.w) has small differences when compared to the FAO file. This results from correcting some typing mistakes encountered in the FAO file. As a result from some research carried out by FAP a number of years ago, the format of the file nutc.bin.w was enlarged to make room for 11 instead of 3 nutritive factors, the resulting file is called mix.nut.

The read/write for this file should be:

```
read(iu,end=992) (jcd(i),i=1,5),(z(j),j=1,11)
```

The contents of the z(i) is:

z(1)	calories	kcal/100gr
z(2)	protein (low)	gr/100gr
z(3)	protein (high)	gr/100gr
z(4)	fat	gr/100gr
z(5)	calcium	mg/100gr
z(6)	iron	mg/100gr
z(7)	vitamin a	iu/100gr (international units)
z(8)	thiamine	mg/100gr
z(9)	riboflavin	mg/100gr
z(10)	niacin	mg/100gr
z(11)	vitamin c	mg/100gr

This time a much larger number of dimensions are generated (not following the standard FAO coding), in order to fit in all the nutritional values and be able to distinguish between different volumes in the large FAP commodity list. As the text listing programs are not able to cope with dimension codes larger than 3, the mapping of dimensions to nutrient content will be given here, and can also be found in comment lines in the program foodvavo.f

meaning	"good" dimensions	"other"	"good" per capita
calories	11	31	51
protein (low)	12	32	52
protein (high)	13	33	53
fat	14	34	54
calcium	15	35	55
iron	16	36	56
vitamin a	17	37	57
thiamine	18	38	58
riboflavin	19	39	59
niacin	20	40	60
vitamin c	21	41	61

For calculating the nutritive content of selected elements of the FAP commodity lists, one can run the program `foodvavo.f` which takes its nutritive factors from the file `mix.nut`.

This program, while generating the nutritive content of calories, proteins, fat, thiamine, etc., per unit (mt, US\$, etc), also calculates the per caput per day value of the elements. In the case of element 5; production, it would be the number of calories produced per caput per day, for element 14; food, the number of calories consumed by humans per day.

It should be very easy to adapt the program `foodvavo.f` in a way suitable to calculate other aspects of demand and supply, measured in nutritive values.

## 11. Reduce Prices and Quantities

The FAP models belonging to the BLS consist of a supply and a demand module. Similarly, the commodity classifications of the detailed models and the simplified models have distinct supply and demand commodities (see Table 3).

The programs that generate SUAs and prices for the large and the small commodity lists do not acknowledge the division of the models into demand and supply part. In the case of quantities, it is straightforward to add up those quantities of supply commodities which jointly form a demand commodity, e.g. protein feed, fish meal and meat meal constitute together the demand for protein feed.

When it comes to prices it is necessary to weight the prices to be added up with the corresponding quantities. For this purpose a "reduction" program has been written, which takes as input the reduction pattern (see Table 3) and generates prices for the "reduced" commodity lists.

The program is called `reduce.f` and has to be run separately for the detailed (`ag27`) and the small (`ag9`) FAP commodity lists. Each type of price (production, feed, food, other) has to be run separately as well.

### 11.1. Exceptions

For the special aggregations for the feed program, there is also a special reduction program called `"reducekl.f"`.



## 12. Population Data

Population data is kept in the FAP data bank in the same format as SUA time series. The time series for total population have always been included in the SUA tapes from FAO Rome. At FAP they are kept in two files:

- separately, in the file pop.bin
- in the macro data file.

FAO originally only provided FAP with total population, but the modelling work required agriculture and non-agriculture labour force as well. To generate these time series ILO data was used, see Sichra WP-84-93. In its latest release (66-81) the FAO SUA tapes also have information on:

commodity	element	dimension	
1	1	1	total population in (1000s)
1	14	1	ag pop / tot pop * 10**4 (ratio)
1	15	1	rur pop / tot pop (ratio)
1	16	1	tot lab / tot pop (ratio)
1	17	1	ag lab / tot lab (ratio)

From these time series the population time series needed by FAP can be constructed using the program "pop.f". The resulting data records are:

commodity	element	dimension	
1	1	1	total population (in 1000s)
1	14	1	total labour force (in 1000s)
1	16	1	agriculture labour force (in 1000s)
1	17	1	non-agriculture labour force (in 1000s)

They have been merged into the corresponding all.xxx file. It will be noticed that the assigning of element codes in FAP is not consistent with the FAO practice (history !!).

## REFERENCES

- U. Sichra. *The FAP Data Bank, Part 1: Organization, Contents and Management*. WP-84-93. International Institute for Applied Systems Analysis, Laxenburg, December 1984.
- G. Fischer and U. Sichra. *The Aggregation of the Agricultural Supply Utilization Accounts*. WP-83-42. International Institute for Applied Systems Analysis, Laxenburg, 1983.
- U. Sichra. *World Prices for the Detailed and Small FAP Commodity Lists*. WP-84-95. International Institute for Applied Systems Analysis, Laxenburg, 1984.

**APPENDIX 1: Data File Format**

record position	field length /class	field name
1-2	2N	sub-system
3-5	3N	country
6-9	4N	commodity
10-11	2N	element
12	1N	dimension
13-21	9	(blank)
22-23	2N	first year indicator
24-34		first year data
+ 35		+ data status indicator
to	192A/N	to
204-214	(16 x 12)	sixteenth year data
+ 215		+ data status indicator
216-219	4N	date
220	1	(blank)

**APPENDIX 2: Nutritive Factors Record**

record position	field length /class	field name
1-2	2N	sub-file
3-4	2N	sub-system
5	1N	language
6-8	3N	country
9-12	4N	commodity
13-15	3	(blank)
16	1	sequence (blank)
17-22	6N	calorie conv.factor
23-28	6N	protein conv.factor
29-34	6N	fat conv.factor
35-76	42	(blank)
77-136	60	(blank)
137-196	60	(blank)
197-200	4N	date

APPENDIX 3: Commodity Codes and Text.

0001	population	0108	cereals nes
0002	macroecon.1	0109	infant food
0003	macroecon.2	0110	wafers
0010	total trade	0111	flour of cereals
0012	land use	0112	bran of cereals
0013	irrigation	0113	cereal prep nes
0014	land use	0116	potatoes
0015	wheat	0117	flour of potatoes
0016	flour wheat	0119	potato starch
0017	bran wheat	0121	potato tapioca
0018	macaroni	0122	sweet potatoes
0020	bread	0125	cassava
0022	pastry	0126	flour of cassava
0023	wheat, starch	0127	cassava tapioca
0024	wheat, gluten	0128	cassava dried
0027	rice, paddy	0129	cassava starch
0028	rice, husked	0136	taro (coco yam)
0031	rice, milled	0137	yams
0032	rice, broken	0149	roots and tubers nes
0034	rice, starch	0150	flour of roots and tuber
0035	bran rice	0151	roots and tubers dried
0036	oil of rice brn	0156	sugar cane
0037	cake ricebrn	0157	sugar beets
0041	breakf cereals	0158	cane sugar
0044	barley	0159	beet sugar
0046	barley, pearl	0161	sugar crops nes
0049	malt barley	0162	sugar (centrifugal, raw)
0050	malt extract	0163	sugar (noncentrifugal)
0051	beer barley	0164	sugar refined
0056	maize	0165	molasses
0058	flour maize	0167	sugar nes
0059	bran maize	0168	sugar and syrups nes
0060	oil maize	0169	sugar confectionery
0061	oake maize	0170	beet pulp
0063	maize gluten	0171	bagasse
0064	starch maize	0176	sugars flavoured
0066	beer maize	0181	beans, dry
0067	white maize	0187	broad beans, dry
0068	pop corn	0191	peas, dry
0071	rye	0195	chick-peas
0072	flour rye	0197	cow peas, dry
0073	bran rye	0201	pigeon peas
0075	oats	0205	lentils
0076	oats, rolled	0210	vetches
0079	millet	0211	lupins
0080	flour millet	0212	pulses nes
0081	bran millet	0216	flour of pulses
0082	beer millet	0217	brazil nuts
0083	sorghum	0220	cashew nuts
0084	flour sorghum	0221	chestnuts
0085	bran sorghum	0222	almonds
0086	beer sorghum	0223	walnuts
0089	buckwheat	0224	pistachios
0090	flour buckwh	0225	kolanuts
0091	bran buckwh	0226	hazelnuts
0092	quinoa	0230	areca nuts (betel)
0101	canary seed	0231	cashew nuts shelled
0103	mixed grain	0232	almonds shelled
0104	flour mix gr	0233	walnuts shelled
0105	bran of mix gr	0234	hazelnuts shelled
	population		nuts nes
	macroeconomics one		
	macroeconomics two		
	total trade		
	land use		
	irrigation		
	land use (bis)		
	wheat		
	flour of wheat		
	bran of wheat		
	macaroni		
	bread		
	pastry		
	wheat starch		
	wheat gluten		
	rice, paddy		
	rice, husked		
	rice, milled		
	rice, broken		
	rice, starch		
	bran of rice		
	oil of rice bran		
	cake of rice bran		
	breakfast cereals		
	barley		
	barley, pearled		
	malt of barley		
	malt extracts		
	beer of barley		
	maize		
	flour of maize		
	bran of maize		
	oil of maize		
	cake of maize		
	maize gluten		
	starch of maize		
	beer of maize		
	white maize		
	pop corn		
	rye		
	flour of rye		
	bran of rye		
	oats		
	oats, rolled		
	millet		
	flour of millet		
	bran of millet		
	beer of millet		
	sorghum		
	flour of sorghum		
	bran of sorghum		
	beer of sorghum		
	buckwheat		
	flour of buckwheat		
	bran of buckwheat		
	quinoa		
	canary seed		
	mixed grain		
	flour of mixed grain		
	bran of mixed grain		

0235	preprd nuts	0232	oake cotton	0232	cake of cotton seed
0236	soybeans	0233	linseed	0233	linseed
0237	oil soyabean	0234	oil linseed	0234	oil of linseed
0238	oake soybean	0235	cake linseed	0235	cake of linseed
0239	soya sauce	0236	hempseed	0236	hempseed
0240	soya paste	0237	oil hempseed	0237	oil of hempseed
0241	soya curd	0238	oake hempseed	0238	oake of hempseed
0242	groundnuts	0239	oilseeds nes	0239	oilseeds nes
0243	groundnut she	0240	oil vq or ns	0240	oil of veget origin nes
0244	oil groundnt	0241	oak oilsd ns	0241	oakes of oilseeds nes
0245	oake groundt	0242	oil meals	0242	flour/meal of oilseeds
0249	coconuts	0243	cabbages	0243	cabbages
0250	coconuts,des	0244	artichokes	0244	artichokes
0251	copra	0245	asparagus	0245	asparagus
0252	oil coconuts	0246	lettuce	0246	lettuce
0253	oake coconut	0247	spinach	0247	spinach
0256	palm kernels	0248	tomatoes	0248	tomatoes
0257	palm oil	0249	juice of tomatoes	0249	juice of tomatoes
0258	oil,palm ker	0250	tomato paste	0250	tomato paste
0259	oake,palm ker	0251	peeld tomato	0251	peeled tomatoes
0260	olives	0252	cauliflower	0252	cauliflower
0261	olive oil	0253	pumpk+sq+grd	0253	pumpkins, squash, gourds
0262	olive,pres	0254	cucumbr+gherk	0254	cucumbers and gherkins
0263	karite nuts	0255	eggplants	0255	eggplants
0264	karit nt but	0256	chill+pep,grn	0256	chillies+peppers, green
0265	castor beans	0257	onions,green	0257	onions+shallots, green
0266	oil cast bns	0258	onions, dry	0258	onions, dry
0267	sunflwr seed	0259	garlic	0259	garlic
0268	oil sunf sd	0260	beans, green	0260	beans, green
0269	oake sunf sd	0261	peas, green	0261	peas, green
0270	rapeseed	0262	brd bean, grn	0262	broad beans, green
0271	oil rapeseed	0263	string beans	0263	string beans
0272	oake rapeseed	0264	carrots	0264	carrots
0273	olive resid	0265	green corn (maize)	0265	green corn (maize)
0274	oil oliveres	0266	mushrooms	0266	mushrooms
0275	tung nuts	0267	chicory roots	0267	chicory roots
0276	tung oil	0268	veg prod fresh or dried	0268	veg prod fresh or dried
0280	safflower	0269	carobs	0269	carobs
0281	oil safflower	0270	vegetables fresh nes	0270	vegetables fresh nes
0282	oake safflwr	0271	vegetables dried nes	0271	vegetables dried nes
0289	sesame seed	0272	vegetables canned nes	0272	vegetables canned nes
0290	oil ses sd	0273	juice of vegetables nes	0273	juice of vegetables nes
0291	oake ses sd	0274	vegs dehydrated	0274	vegs.dehydrated 055.1
0292	mustard seed	0275	vegs pr by vinegar	0275	vegs pr by vinegar 55.51
0293	oil must sd	0276	vegs pr nes	0276	vegs pr nes 55.52
0296	poppy seed	0277	vegetables frozen	0277	vegetables frozen
0297	oil pop sd	0278	bananas	0278	bananas
0298	oake pop sd	0279	plantains	0279	plantains
0299	melonseed	0280	oranges	0280	oranges
0305	tallow seeds	0281	juice of oranges	0281	juice of oranges
0306	veg tallow	0282	lang.mand.clement.satsma	0282	lang.mand.clement.satsma
0307	stillingia oil	0283	lemons and limes	0283	lemons and limes
0310	kapok fruit	0284	grapefruit and pomelo	0284	grapefruit and pomelo
0311	kapokseed sh	0285	grapefruit juice	0285	grapefruit juice
0312	kapoksee shed	0286	citrus fruit nes	0286	citrus fruit nes
0313	oil of kapok	0287	citrus fruit juice nes	0287	citrus fruit juice nes
0314	oake kapok	0288	apples	0288	apples
0328	seed cotton	0289	oider	0289	oider
0329	cottonseed	0290	pears	0290	pears
0331	oil cotton s				

0523	quinoes	quinoes	0654	dregs,br+dis	dregs from brewing+dist.
0526	apricots	apricots	0655	veg root fod	vegetables+roots,fodder
0530	sour cherry	sour cherries	0656	coffee,green	coffee, green
0531	cherries	cherries	0657	coffee roast	coffee roasted
0534	peaches	peaches and nectarines	0658	coffee subst	coffee subst cont coffee
0536	plums	plums	0659	coffee extr	coffee extracts
0537	plums, dried	plums, dried (prunes)	0661	cocoa beans	cocoa beans
0541	stone fruit	stone fruit nes. fresh	0662	cocoa powder	cocoa powder
0542	pome fruit	pome fruit nes, fresh	0663	cocoa paste	cocoa paste oake
0544	strawberries	strawberries	0664	cocoa butter	cocoa butter
0547	raspberries	raspberries	0665	choc prod ns	chocolate products nes
0549	gooseberries	gooseberries	0666	???????	???????
0550	currants	currants	0667	tea	tea
0552	blueberries	blueberries	0671	mate	mate
0554	cranberries	cranberries	0674	tea nes	tea nes
0558	berries nes	berries nes	0677	hops	hops
0560	grapes	grapes	0687	pepper w/l/b	pepper,white/long/black
0561	raisins	raisins	0689	pimento	pimento, allspice
0563	must grapes	must of grapes	0692	vanilla	vanilla
0564	wine	wine	0693	cinnamon	cinnamon (canelle)
0565	vermth simil	vermouths and similar	0698	cloves	cloves, whole+stems
0567	watermelons	watermelons	0702	nutmeg	nutmeg, mace, cardamoms
0568	mel ino cant	melons incl cantaloupes	0711	anise	anise, bacian, fennel
0569	figs	figs	0723	spices nes	spices nes
0570	figs, dried	figs, dried	0737	oil citronell	oil of citronella
0571	mangoes	mangoes	0748	peppermint	peppermint
0572	avocados	avocados	0753	ess oils nes	essential oils nes
0574	pineapples	pineapples	0754	pyrethrum	pyrethrum
0575	pineapple can	pineapples, canned	0755	pyret extr.	pyrethrum extract
0576	pineap juice	pineapple juice	0756	pyret maro	pyrethrum maro
0577	dates	dates	0766	seed cotton	seed cotton
0600	papayas	papayas	0767	cotton lint	cotton lint
0603	frt trop nes	fruit tropical fresh nes	0768	cotton cardd	cotton carded combed
0604	fr trp dr ns	fruit tropical dried nes	0769	cotton waste	cotton waste
0619	fruit nes	fruit fresh nes	0770	cotton linter	cotton linter
0620	fruit dr nes	fruit dried nes	0771	flax raw	flax fibre raw
0622	fruit juice	fruit juice nes	0773	flax fibre	flax fibre and tow
0623	fruit pr nes	fruit prep nes	0774	flax tow	flax tow waste
0624	flour fruit	flour of fruit	0777	hemp fibre	hemp fibre and tow
0632	???????	???????	0780	jute	jute
0633	bev non-alc	beverages non-alcoholic	0782	jute-like	jute-like fibres
0634	bev dis alc	beverages dist alcoholic	0788	ramie	ramie
0635	straw, husks	straw, husks	0789	sisal	sisal
0636	maize fd+sil	maize for forage+silage	0800	agave nes	agave fibres nes
0637	sorghum fs	sorghum for forage+silag	0809	abaca	abaca (manila hemp)
0638	rye grass fs	rye grass,forage+silage	0821	fibre nes	fire crops nes
0639	grasses fs	grasses nes,forage+silag	0826	tobacco	tobacco leaves
0640	clover fs	clover for forage+silage	0828	cigarettes	cigarettes
0641	alfalfa fs	alfalfa for forage+silag	0829	cigars	cigars cheroots
0643	legumes fs	legumes nes,forage+silag	0831	tobacco prod	tobacco products nes
0644	cabbage fod	cabbage for fodder	0836	nat rubber	natural rubber
0645	pumpkins fod	pumpkins for fodder	0837	rubber dry	rubber natural dry
0646	turnips fod	turnips for fodder	0839	natural gums	natural gums
0647	beets fodder	beets for fodder	0840	com feed cat	compound feed, cattle
0648	carrots fod	carrots for fodder	0841	com feed pou	compound feed, poultry
0649	swedes fod	swedes for fodder	0842	com feed pig	compound feed, pigs
0650	leaves+tops	leaves and tops	0845	com feed oth	compound feed, other or nes
0651	forage prod	forage products nes	0846	glut feed&me	gluten feed and meal
0652	veg prod	veg prod for feed	0850	feed sup	feed supplements
0653	food wastes	food wastes	0851	nonprot nitr	non protein nitrogens

0852	concentr nes	other concentrates, nes	0958	hide w buffl	hides wet-salted buffal
0853	vitamins	vitamins	0959	hide d buffl	hides dry-salted buffal
0854	feed additiv	feed additives	0972	ind buffmeat	indigenous buffalo meat
0855	feed mineral	feed minerals	0973	bio buffmeat	biological buffalo meat
0857	hay non-leg	hay non-leguminous	0974	lambs	lambs
0858	hay legumin	hay (clover, lucerne, etc)	0975	lamb meat	lamb meat
0859	hay unspcif	hay (unspecified)	0976	sheep	sheep
0860	range past	range pasture	0977	mutton lamb	mutton and lamb
0861	improv past	improved pasture	0978	offals sheep	offals of sheep, edible
0862	fl/meal miso	flour/meal, leg, veg, roots	0979	fat of sheep	fat of sheep
0864	calves	calves	0982	sheep milk	sheep milk
0865	veal	veal	0983	butter sheep	butter+ghee (sheep milk)
0866	cattle	cattle	0984	sheep cheese	cheese of sheep milk
0867	beef veal	beef and veal	0985	sk milk shee	skim sheep milk
0868	offals cattl	offals of cattle, edible	0987	wool, greasy	wool, greasy
0869	fat cattle	fat of cattle	0988	wool, scoured	wool, scoured
0870	beef boneless	beef and veal, boneless	0994	grease wool	grease incl lanolin wool
0872	beef dss	beef dried salt smoked	0995	sheepskins	sheepskins, fresh
0873	meat extract	meat extracts	0996	skin w sheep	skin wet-salted sheep
0874	sausage beef	sausages beef and veal	0997	skin d sheep	skin dry-salted sheep
0875	beef prep	beef preparations	0998	skin nes sh	skin nes sheep
0876	beef canned	beef canned	0999	skinwool sh	skin with wool sheep
0882	cow milk	cow milk, whole, fresh	1007	wool shoddy	wool shoddy
0885	cream, fresh	cream, fresh	1008	hair carded	hair carded or combed
0886	butter, cows	butter of cow milk	1009	wool waste	wool hair waste
0887	ghee, cows	ghee (from cow milk)	1012	ind sheepmeat	indigenous sheep meat
0888	sk milk cows	skim milk of cows	1013	bio sheepmeat	biological sheep meat
0889	wh milk, cond	whole milk, condensed	1014	kids	kids
0890	whey, condens	whey, condensed	1015	kids meat	kids meat
0894	wh milk, evap	whole milk, evaporated	1016	goats	goats
0895	skmilk evap	skim milk, evaporated	1017	goat meat	goat meat
0896	sk milk cond	skim milk, condensed	1018	offals goats	offals of goats, edible
0897	cowmilk dry	dry whole cow milk	1019	fat of goats	fat of goats
0898	milk sk dr o	dry skim oow milk	1020	goat milk	goat milk
0899	dry buttermilk	dry buttermilk	1021	goat cheese	cheese of goat milk
0900	dry whey	dry whey	1025	goatskins	goatskins, fresh
0901	cheese w oow	cheese (whole cow milk)	1026	skin w goat	skins wet-salted goats
0903	whey, fresh	whey, fresh	1027	skin d goat	skins dry-salted goats
0904	cheese s cow	cheese (skim oow milk)	1028	skin nes goa	skins nes goats
0917	casein	casein	1032	ind goatmeat	indigenous goat meat
0919	cattle hides	cattle hides, fresh	1033	bio goatmeat	biological goat meat
0920	hide w cattl	hides wet-salted cattle	1034	pigs	pigs
0921	hide d cattl	hides dry-salted cattle	1035	pigmeat	pigmeat
0922	hide n cattl	hides nes cattle	1036	offals pigs	offals of pigs, edible
0927	skin f calve	skins fresh of calves	1037	fat pigs	fat of pigs
0928	skin w calve	skins wet-salt calves	1039	bacon pigs	bacon-ham of pigs
0929	skin d calve	skins dry-salt calves	1041	sausages pig	sausages pig meat
0930	skin n cattl	skins nes calves	1042	meat pr pig	meat preparations pigs
0944	ind cattmeat	indigenous cattle meat	1043	lard	lard
0945	bio cattmeat	biological cattle meat	1044	pigskins	pigskins, fresh
0946	buffaloes	buffaloes	1045	skin w pigs	skin wet-salted pigs
0947	buffalo meat	buffalo meat	1046	skin d pigs	skin dry-salted pigs
0948	offal buffal	offals of buffalo, edible	1047	skin nes pig	skin nes pigs
0949	fat buffalo	fat of buffalo	1055	ind pigmeat	indigenous pigmeat
0951	buffalo milk	buffalo milk	1056	bio pigmeat	biological pig meat
0952	butter buffl	butter of buffalo milk	1057	chickens	chickens
0953	ghee buffalo	ghee (from buffalo milk)	1058	chicken meat	chicken meat
0954	milk sk buff	skim milk of buffalo	1059	offal chickn	offals liver of chickens
0955	chees buffl	cheese of buffalo milk	1060	meat pr chck	meat preparations chick
0957	buffalo hide	buffalo hides, fresh	1061	meat od chok	meat canned chicken

1062 hen eggs  
1063 eggs 1 hen  
1064 eggs dry hen  
1065 fat poultry  
1066 fat r poult  
1067 hen eggs no  
1068 ducks  
1070 ind duckmeat  
1071 bio duckmeat  
1072 geese  
1077 ind geesmeal  
1078 bio geesmeal  
1079 turkeys  
1087 ind turkmeat  
1088 bio turkmeat  
1089 poultry meat  
1091 eggs ex hen  
1092 oth egg (no)  
1094 ind ohokmeat  
1095 bio ohokmeat  
1096 horses  
1097 horsemeat  
1098 ??????  
1100 hair horses  
1102 horse hides  
1103 hide w horse  
1104 hide d horse  
1105 hide y horse  
1107 asses  
1110 mules  
1120 ind horsemeat  
1121 bio horsemeat  
1122 ind ass meat  
1123 bio ass meat  
1124 ind mulemeat  
1125 bio mulemeat  
1126 camels  
1127 meat camel  
1128 offals camel  
1129 fat camel  
1130 camel milk  
1133 hides camel  
1134 hides w camel  
1135 hides d camel  
1136 hide u camel  
1137 ind camlmeat  
1138 bio camlmeat  
1140 ??????  
1141 ??????  
1163 game meat  
1164 meat dry nes  
1166 meat nes  
1167 offals nes  
1168 oils animal  
1171 animals nes  
1172 meat pr nes  
1173 meat meal  
1174 fish meal  
1181 beehives  
1182 honey

1183 hen eggs  
1185 eggs liquid hen  
1186 eggs dry whole yolks hen  
1187 fat of poultry  
1195 fat of poultry rendered  
1213 hen eggs (no)  
1214 ducks  
1215 indigenous duckmeat  
1216 biological duckmeat  
1217 geese  
1218 indigenous geese meat  
1219 biological geese meat  
1221 turkeys  
1222 indigenous turkey meat  
1223 biological turkey meat  
1225 poultry t (excl hen)  
1242 eggs, exc hen eggs  
1243 indigenous chicken meat  
1259 biological chicken meat  
1274 horses  
1275 horsemeat  
1276 ??????  
1277 hair of horses  
1293 horse hides, fresh  
1294 hides wet-salted horses  
1295 hides dry-salted horses  
1296 hides unspecified horses  
1297 asses  
1299 mules  
1300 indigenous horsemeat  
1301 biological horse meat  
1302 indigenous ass meat  
1303 biological ass meat  
1304 indigenous mule meat  
1305 biological mule meat  
1306 camels  
1307 meat of camels  
1308 offals of camel, edible  
1309 fat of camels  
1310 camel milk  
1311 camel hides, fresh  
1312 hides wet-salted camels  
1313 hides dry-salted camels  
1314 hides unsp camel  
1315 indigenous camel meat  
1316 biological camel meat  
1317 ??????  
1318 ??????  
1319 game meat  
1320 meat, dried, nes  
1321 meat nes  
1324 offals nes  
1325 animal oils and fats nes  
1326 live animals nes  
1327 meat prepared nes  
1328 meat meal  
1330 fish meal  
1331 beehives  
1332 honey

beeswax  
cocoon reel  
silk, raw  
cocoon unr  
fur skins  
hides nes fr  
hide w s  
hide ds  
hide nes  
leather used  
hair fine  
hair coarse  
stearine  
degras  
oils fish  
tallow  
food prep  
margarine  
fats prep ns  
food wastes  
oils boiled  
oils hydrogen  
fatty acids  
res fatty s  
org mat 29  
seeds  
spermaceti  
waxes veg  
protein  
soil mach  
agr mach nes  
tractors all  
tractors agric tot  
tractors agric crawler  
tractors agric wheel  
garden tract  
hary thresh  
milking mash  
lindane  
d d t  
b h c  
other chlor  
aldrin elo  
carbammates i  
other herbiold  
toxaphene  
fenitrothion  
parathion  
malathion  
oth org phosph  
chlorobenzil  
bot linsc pyr  
bot inso oth  
miner oils  
arsenicals  
bromides  
oth fumigant  
other insect  
dinitro comp  
dithiocarbam

beeswax  
cocoon, reelable  
silk, raw and waste  
cocoon, unreelable  
fur skins  
hides+skins nes. fresh  
hide wet-salted  
hide dry-salted  
hide nes  
leather used and waste  
hair fine animal  
hair coarse nes  
lard stearine oil  
degras  
oils fish mar mamm  
tallow  
food prep nes  
margarine + shortening  
fat preparations nes  
food wastes prep feed  
oils boiled etc  
oils hydrogenated  
fatty acids oils  
res fatty subst  
crude organic materls 29  
seeds  
spermaceti  
waxes veg  
protein concentrates  
soil machinery  
agr machinery nes  
tractors all  
tractors agric tot  
tractors agric crawler  
tractors agric wheel  
garden tractors  
harvester-threshers  
milking machines  
lindane  
d d t  
b h c  
other chlor hydrocarbons  
aldrin and sim insectic  
carbammates insecticide  
other herbicides  
toxaphene  
fenitrothion  
parathion  
malathion  
oth org phosph insectio  
chlorobenzilate  
pyrethrum  
other botanical insectio  
mineral oils  
arsenicals  
bromides  
other fumigants  
other insecticides  
dinitro compounds  
dithiocarbammates



1333 seed dress m  
 1334 seed dress o  
 1335 sulphur  
 1336 lime sulphur  
 1337 urea derivat  
 1338 aromatic op  
 1339 carbamates h  
 1340 copper comp  
 1341 oth fungicid  
 1342 2,4,-d  
 1343 mopa  
 1344 2,4,5-t  
 1345 triazines  
 1346 anticoagul  
 1347 oth rodent  
 1348 pesticid nes  
 1350 plant gr reg  
 1352 methoxychlor  
 1353 aliphatic cp  
 1354 nematocides  
 1355 el phosphide  
 1356 oar tetrachl  
 1357 pesticoides  
 1360 nitrogfertilz  
 1361 ammon sulph  
 1362 ammon nitrat  
 1363 amm sul nitr  
 1364 sodium nitr  
 1365 calcium nitr  
 1366 calcium oyan  
 1367 urea  
 1368 amm phosph n  
 1369 other nitr fer  
 1370 oth compl n  
 1371 ammonia d ap  
 1372 calo am nitr  
 1374 nit fert nes  
 1375 phosphfertilz  
 1376 sing superph  
 1377 cono superph  
 1378 basic slag  
 1379 amm phosph p  
 1380 other phos fer  
 1381 oth compl p  
 1385 phos fer nes  
 1386 potashfertiliz  
 1387 potas sulph  
 1388 muriate 4S  
 1389 muria 20-45  
 1390 crude salts  
 1391 oth pot fert  
 1392 oompl fer k  
 1396 pot fert nes  
 1397 fert m nes  
 1398 nat sod nitr  
 1399 phosphat nat  
 1400 pot salt nat  
 1401 fertil organ  
 1402 ammonia  
 1403 phosph acid

1404 seed dress org  
 1405 seed dress others  
 1406 sulphur  
 1410 lime sulphur  
 1411 urea derivat  
 1412 aromatic compounds  
 1501 carbamates herbicide  
 1502 copper compounds  
 1503 other fungicides  
 1504 2,4,-d  
 1505 mopa  
 1506 2,4,5-t  
 1507 triazines  
 1508 antioogulants  
 1509 other rodentioides  
 1510 pestioides nes  
 1511 plant growth regulators  
 1514 methoxychlor  
 1515 aliphatic compounds  
 1516 nematocides  
 1517 aluminum phosphide  
 1518 carbon tetrachloride  
 1519 pesticoides  
 1520 nitrogenous fertilizers  
 1521 ammonium sulphate  
 1522 ammonium nitrate  
 1523 ammonium sulphat nitrate  
 1524 sodium nitrate  
 1527 calcium nitrate  
 1528 calcium cyanide  
 1529 urea  
 1530 ammonium phosphate (n)  
 1531 other nitrogenous fert  
 1532 oth complex fert (n)  
 1533 ammonia dir application  
 1534 calcium ammonium nitrate  
 1535 nitrogen fertilizers nes  
 1536 phosphate fertilizers  
 1537 single superphosphate  
 1540 concent superphosphate  
 1541 basic slag  
 1542 ammonium phosphat (p20S)  
 1543 other phosphate fertl  
 1544 other complex fert (p20S)  
 1545 phosphate fertiliz nes  
 1546 potash fertilizers  
 1547 potassium sulphate  
 1548 muriate over 4S k20  
 1549 muriate 20-45 k20  
 1550 crude salts to 20 k20  
 1553 other potash fertilizers  
 1554 complex fertilizer (k20)  
 1555 potash fertilizers nes  
 1556 fertilizers manuf nes  
 1557 natural sodium nitrate  
 1558 natural phosphates  
 1559 natural potassium salts  
 1562 fertilizers organic  
 1563 ammonia  
 1564 phosphoric acid

1404 sulphur acid  
 1405 agric lime  
 1406 gypsum  
 1410 int comb eng  
 1411 electric motor  
 1412 trucks farms  
 1501 frwtr diad f  
 1502 frwtr fz whl  
 1503 frwtr fillet  
 1504 frwtr fz flt  
 1505 frwtr oured  
 1506 frwtr canned  
 1507 frwtr pr nes  
 1508 frwtr meals  
 1509 frwtr bdy oil  
 1510 frwtr liver oil  
 1511 frwtr meal of  
 1514 dmrs1 fresh  
 1515 dmrs1 fz whl  
 1516 dmrs1 fillet  
 1517 dmrs1 fz flt  
 1518 dmrs1 oured  
 1519 dmrs1 canned  
 1520 dmrs1 pr nes  
 1521 dmrs1 meals  
 1522 dmrs1 bdy oil  
 1523 dmrs1 lvr oil  
 1524 dmrs1 meal of  
 1527 pelagic frsh  
 1528 pelgc fz wht  
 1529 pelgc fillet  
 1530 pelgc fz flt  
 1531 pelgc cured  
 1532 pelgc canned  
 1533 pelgc pr nes  
 1534 pelgc meals  
 1535 pelg bdy oil  
 1536 pelg lvr oil  
 1537 pelg meal of  
 1540 marine nes f  
 1541 marine fz whl  
 1542 marin fillet  
 1543 marin fz flt  
 1544 marin oured  
 1545 marin pr nes  
 1546 marin canned  
 1547 marin meals  
 1548 marn bdy oil  
 1549 marn lvr oil  
 1550 marn meal of  
 1553 crstaceans f  
 1554 crstc frozen  
 1555 crstc cured  
 1556 crsto canned  
 1557 crstc pr nes  
 1558 crsto meals  
 1559 crst meal of  
 1562 mlfusos frsh  
 1563 molso frozen  
 1564 molso oured

1404 sulphuric acid  
 1405 agricultural lime  
 1406 gypsum  
 1410 internal combust engines  
 1411 electric motors  
 1412 trucks on farms  
 1501 freshwater diadrom fresh  
 1502 freshwater frozen whole  
 1503 freshwater fillets  
 1504 freshwater frozen filets  
 1505 freshwater cured  
 1506 freshwater canned  
 1507 freshwater prep nes  
 1508 freshwater meals  
 1509 freshwater body oils  
 1510 freshwater liver oils  
 1511 freshwater meal fr offal  
 1514 demersal marine fish frsh  
 1515 demersal frozen whole  
 1516 demersal fillets  
 1517 demersal frozen fillets  
 1518 demersal cured  
 1519 demersal canned  
 1520 demersal prep nes  
 1521 demersal meals  
 1522 demersal body oils  
 1523 demersal lvr oils  
 1524 demersal meal of  
 1527 pelagic frsh  
 1528 pelagic fz wht  
 1529 pelagic fillet  
 1530 pelagic fz flt  
 1531 pelagic cured  
 1532 pelagic canned  
 1533 pelagic prep nes  
 1534 pelagic meals  
 1535 pelagic body oils  
 1536 pelagic liver oils  
 1537 pelagic meal from offal  
 1540 marine fish nes fresh  
 1541 marine nes frozen whole  
 1542 marine nes fillets  
 1543 marine nes frozen fillet  
 1544 marine nes cured  
 1545 marine nes canned  
 1546 marine nes prep nes  
 1547 marine nes meals  
 1548 marine nes body oils  
 1549 marine nes liver oils  
 1550 marine nes meal fr offal  
 1553 crustaceans fresh  
 1554 crustaceans frozen  
 1555 crustaceans cured  
 1556 crustaceans canned  
 1557 crustaceans prep nes  
 1558 crustaceans meals  
 1559 crustaceans meal f offal  
 1562 molluscs exol cephalop frsh  
 1563 molluscs frozen  
 1564 molluscs cured

1565	molsc canned	molluscs canned
1566	molsc meals	molluscs meals
1567	molsc meal of	molluscs meal from offal
1570	cephlp fresh	cephcpods fresh
1571	ophlp frozen	cephalopods frozen
1572	ophlp cured	cephalopods cured
1573	cphlp canned	cephalopods canned
1574	cphlp pr nes	cephalopods prep nes
1575	ophlp meals	cephalopods meals
1576	cphl meal of	cephalopods meal f offal
1579	aquto mammal	aquatic mammals
1580	aq m meat	aquatic mammals meat
1581	aq m meals	aquatic mammals meals
1582	aq m oils	aquatic mammals oils
1583	aq m prep ns	aquatic mammals prepnes
1584	aq m meal of	aqua mammal meal f offal
1587	aquto anim f	aquatic animals nes frsh
1588	aq a cured	aquatic animals cured
1589	aq a meals	aquatic animals meals
1590	aq a prep ns	aquatic animals prep nes
1591	aq a meal of	aqua anim meal fr offal
1594	aqutc plants	aquatic plants
1595	aq p dried	aquatic plants dried
1596	aq p prep ns	aquatic plants prep nes
1599	fish tot val	fishery total value

APPENDIX 4: FAP countries (\* = FAP4)

EEC and Japan		Developing Asia	
15	Belgium - Luxembourg (*)	16	Bangladesh
54	Denmark (*)	100	India(*)
68	France (*)	101	Indonesia(*)
78	Federal Rep. of Germany (*)	102	Iran
104	Ireland(*)	103	Iraq
106	Italy(*)	116	Korean Democr. Rep
110	Japan(*)	165	Pakistan(*)
150	Netherlands(*)	171	Philippines
229	UK(*)	216	Thailand
888	Total EEC		
CMEA		Latin America	
27	Bulgaria(*)	9	Argentina(*)
51	Czechoslovakia(*)	21	Brazil(*)
77	German Democratic Republic(*)	138	Mexico(*)
97	Hungary(*)	170	Peru
173	Poland(*)	236	Venezuela
183	Romania(*)		
228	USSR(*)		
777	Total CMEA		
Rest of Europe		Other Countries	
11	Austria(*)	10	Australia(*)
67	Finland(*)	33	Canada(*)
84	Greece(*)	41	China(*)
162	Norway	156	New Zealand(*)
174	Portugal(*)	202	South Africa
203	Spain(*)	231	USA(*)
210	Sweden(*)		
211	Switzerland		
223	Turkey(*)		
248	Yugoslavia		
Developing Africa			
4	Algeria		
59	Egypt(*)		
62	Ethiopia		
114	Kenya(*)		
143	Morocco		
159	Nigeria(*)		
206	Sudan		
212	Syria		
215	Tanzania		
222	Tunisia		