Appendix A – Taxation Management

ASYCUDA++ Functional Manual

V1.15

Appendix
A

Taxation Management.

Specific examples of ASYCUDA Taxation Management and a guide to the use of Taxation Rules.

Contents of Appendix A.

Amendment Control Grid 3
About this Appendix 4
Taxation Management 5
ASYCUDA ++ Taxation Controls 5
Steps in Building Taxation Controls 6
The steps are: 6
'Rules' - an Overview 6
Using Rules 6
SCTL Syntax 7
Control Structures in 'Rules' 7
Customs Headquarters National Configuration 9
National Tariff Rules 9
Building a Taxation Structure 9
Taxation Columns 9
Creating Taxation Columns 10
Taxation Rates 10
How the System Accesses Taxation Rates 10
Taxation Rules 11
Valuation Note Rules 11
Selection of Apportionment Method 12
General Segment Note 12
Item Section 13
Apportionment Rules 14
Value Apportionment Rule 14
Mass Apportionment Rule 15
Global and Item Rules 15
Customs Headquarters 16

General Notes 16
Taxation Processing 17
General segment level 17
Declaration item level 17
Rates of Taxation 17
Inserting Rates of Taxation 18
Validity Dates 19
Simplified Data Management 19
Default Taxation Rate Values 19
Customs Tariff Management 20
Defining The Tariff Range 20
Additional Selection 21
Update 'Actions' 21
Taxation Rules for Special Circumstances 21
Writing Special Taxation Rules 21
Tax Rules in Commodity Code 21
Tax Rules in Customs Procedure Codes 22
Tax Rules used with Agreement codes 22
Tax Rules in Countries and Preference Codes 22
Tax Rules in Market Values 23
Summary: 23
Taxation Management - Table Structures 24
Taxation Rules Editor - Syntax 24
Rule Functions 24
Rule Variables 33
Keywords 36

Amendment Control Grid

Periodically, amendments to this Reference Document will be issued. Each amendment batch will be serially numbered and dated. This Amendment Control Grid is provided in order to maintain a record of the receipt and incorporation of amendments into the Reference Document and thereby ensure that it is kept fully up to date.

1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20

About this Appendix

This Appendix brings together all of the taxation and rule functions of ASYCUDA. It explains the use of the different taxation methods and gives practical examples of them.

It covers:

- Taxation controls;
- Using rules;
- Configuration;
- Taxation processing;
- Examples of special taxation cases;
- Expanded explanations of the ASYCUDA rule functions.

Taxation Management

ASYCUDA ++ Taxation Controls

Taxation calculations are performed through a very flexible and powerful facility that allows the use of most of the declaration fields and control table data elements. Taxation is set up using **MODCHQCF** and **MODCHQ**.

ASYCUDA++ can apply all known methods and combinations of tax calculation. Taxation rates are integrated with the automated Customs tariff. Queries at Commodity Code level give a quick and simple visual display, showing the tax columns and tax rates used in calculations.

The ability to define a priority order for taxes sets the sequence for application, allowing for situations where one tax is included in the basis for calculations of another tax.

The general principles in building the taxation system are as follows

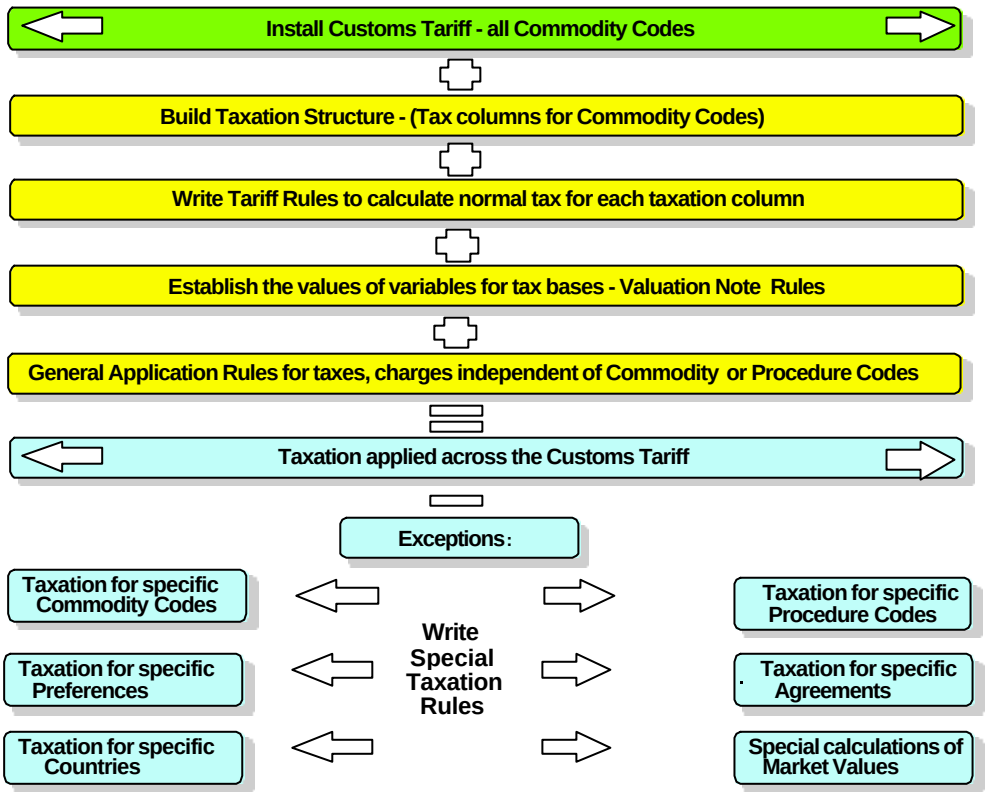


Figure A.1: Building the Taxation structure

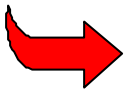
Steps in Building Taxation Controls

To build the national taxation structure and system taxation management controls, you first need

- All Customs Tariff files installed and
- An understanding of how ASYCUDA++ 'Rules' work

The steps are:

1. Determine the best approach for implementing and displaying the various taxes, e.g. within the normal tax columns, or as an 'additional column'[1];
2. Create the taxation structure that is displayed as tax columns against each Commodity Code;
3. Write Taxation Rules to 'activate' and calculate the taxes;
4. Define the Tax Base for values by using Valuation Rules;
5. Define any General Application Rules;
6. Insert rates of tax into the taxation structure ('columns').



Steps 1 to 5 are managed in **MODCHQCF** and step 6 in **MODCHQ**. Additional columns are set up in **MODCHQ** but activated through **MODCHQCF.**

'Rules' - an Overview

In building an ASYCUDA ++ System to meet the needs of a country's own particular taxation regime, specific instructions are written to the System files - these System instructions are called 'rules'. Rules are written in a simple computer programming language that is called **SCTL** ('Specific Customs Taxation Language'). The rules are all written in a similar style, (using SCTL), but are used for different purposes in controlling the actions of the System.

The main groups are:

- **Tariff Column Rules** - Linked to taxation columns in the Tariff. They define the conditions where the tax is to apply, the tax basis and rate, and the action to be performed, such as tax calculation or modification of an existing tax.
- **Valuation Note Rules** - Used to construct or calculate the value tax base (Customs Value) for the purpose of tax calculations. Data taken from the declaration Valuation Note is used in the calculations. Valuation Rules are also used to apportion freight and insurance charges across declaration item lines.
- **General Application Rules** - Used to define taxes to be applied at the highest level, i.e., taxes usually independent of any declaration procedure or commodity classification. (Computer processing fees are a good example.) There are 2 sorts, **Global** and **Item Rules**
- **Special Case MODCHQ Rules** – Used to deal with special taxation cases or tax reliefs.

Using Rules

Rules are processing instructions for the System. Rules are entered through the Rule Editors in Taxation, Valuation Note or Tariff, in **MODCHQ** and **MODCHQCF**. The Rule Editors features an automatic syntax check when a new Rule is input or an existing Rule modified.

[1] See Section 10 of the Reference Document, 'Set up and maintenance', for a full explanation of the function of additional columns.

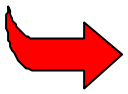
SCTL Syntax

The language and structure of 'Rules' is similar to many other programming languages.

The Rules Editor has a Look Up facility, which displays lists of available variables, functions and keywords.

The **'Variables'** list includes the data elements, captured from the declaration, that can be used to calculate the tax base; data elements such as invoice value, freight and insurance charges.

'Keywords' allow conditional selection for applying taxes, and **'Functions'** perform the tax calculations.



Variables, Keywords and Functions available for use in Rules are listed and described later in this appendix.

Control Structures in 'Rules'

The **'Rules'** language uses basic programming control structures. It uses both sequential processing and selection. Rules are usually a combination of statements that are processed in sequence, together with conditional selections using "If .. Then... Else... Endif" structures.

Examples of Rules are:

Example 1: A simple General Taxation Rule to calculate a computer processing fee charged at two currency units per item line of the declaration:

```
Rule "Computer Fee";
TotalFees IS ItmTotal * 2;
```

Explanation:

Line 1 identifies the Rule by giving it a name. "Rule" is a Rules Keyword, followed by the name given by the writer to this Rule.

Line 2 is a statement that uses the variable ItmTotal, which takes up the total number of declaration items from the general segment of the declaration. The numeric value of ItmTotal is multiplied by 2 (units of national currency), and puts the result to TotalFees - which is the calculated amount of global taxes for the declaration.

Example 2. Calculating tax according to the Commodity Code

```
Rule "Customs Duty";
If TypProc = "4" Then
 Action IS DoTax("ICD", "1", StatVal, Rate(ComCod), (StatVal * Rate
(ComCod) / 100));
Endif;
```

Explanation:

Line 1 is the name of the Rule; the Keyword Rule, followed by the name the writer gives to the Rule, in this case "Customs Duty".

Line 2 is selective, restricting application of the Rule to TypeProc = "4".

TypeProc is the declaration General Procedure Code, taken from the general segment of the declaration. In this case "4" means imports for home use, restricting the tax to import declarations.

Line 3 is the action statement. It uses the Rules Function called DoTax( )

```
Action Is DoTax( "ICD" , "1" ,StatVal, Rate(ComCod), StatVal * Rate
(ComCod)/100);
```

The function **DoTax("TaxCod", "MP", TaxBase, Rate, Amount)** carries out the following action

- Recognises **"ICD" - "TaxCod"** (Tax Code) - as Import Customs Duty
- Recognises from **"1" - "MP"** (Method of Payment) that the tax is to be paid rather than "0", which means not payable.

StatVal (Statistical Value) is a variable that represents the **TaxBase** calculated (by a Valuation Rule) from declaration data.

Rate(ComCod) is the **Rate** of tax and is taken up from the Commodity Code column.

StatVal * Rate(ComCod)/100) is the calculated **Amount** of tax, using the TaxBase and the Rate of tax

Example 3. Relief of the normal (full) Customs Duties:

```
Rule "EEC";
If PreferCod = "EEC" Then
  Num01 IS Rate( ComCod );
  Num02 IS ( StatVal * Num01 / 100 );
  If Num01 > 0 Then
    Action IS RelTax( "ICD" , "EEC" , StatVal , Num01 , Num02 );
  Endif;
  If Num01 = 0 Then
    Action IS RelTax( "ICD" , "EEC" , StatVal , 0 , 0 );
  Endif;
Endif;
```

Explanation:

This Rule applies if PreferCod (Item Preference Code) - taken from the declaration data - is "EEC".

The numeric values of the tax rate (taken from the Commodity Code column) and the tax calculation (tax base multiplied by rate) have been given to the working variables called Num01 and Num02. This is to make the 'action' line easier to read.

The function in this case is RelTax( ).

RelTax ("TaxCod", "Relcode", TaxBase, Rate, Amount) carries out the following action

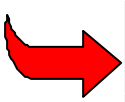
"**TaxCod**" is '**ICD**" - Import Customs Duty

"**RelCode"** is for statistical purposes and allocates the amount of tax 'lost' by the relief to Relief Code "EEC".

Num01 - **Rate(ComCod)** is the Rate of tax and is taken from Commodity Code column "EEC".

Num02 - **Statval * Rate(ComCod)/100)** is the calculated Amount of tax, using the TaxBase and the Rate of tax applying for "EEC".

(The second 'action' statement - "If Num01 = 0" is to allow for rates of duty of "0").



The Num0* variables do not initialise automatically. They must be assigned to the rule if that is what you want. If you need to carry a variable from rule to rule you can use this feature as a sort of accumulator.

Customs Headquarters National Configuration

MODCHQCF is used to build the national taxation structure, to apply taxes with general application (independent of declaration procedures or Commodity Codes), and to define actions that the System will use to calculate taxes. This is managed through a combination of Global, Item, National Tariff and Taxation Rules.

National Tariff Rules

When details of a Tariff classification or Commodity Code within the National Tariff are displayed, the screen looks like the example below:

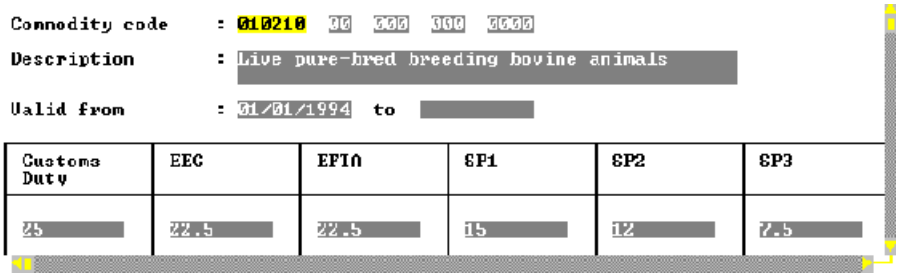


Figure A.2: MODCBR: Reference: Customs Tariff: Commodity Code: Taxation

The column headings (labels) describe the tax on the Commodity, and the boxes contain the matching rate of taxation. Any combination of these taxes can be applied to a transaction - either a number of taxes, or by selecting alternatives. This is controlled through ' Taxation Rules'.

Building a Taxation Structure

The example above shows an existing taxation structure. 'Tariff Rules' allow the building of the 'central body' of the Tariff, by defining the number of columns in the Tariff, assigning titles to the columns, and setting a priority order for the order in which taxes are to be calculated. The columns in the example above were created by the following structure

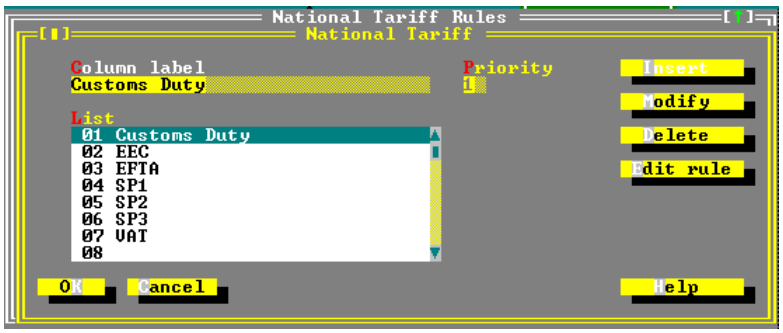


Figure A.3: MODCHQCF: Functions: Taxation Standards: National Tariff

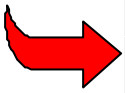
Taxation Columns

Taxation columns define taxes and rates of tax with general application, such as Customs Duty and VAT, and can also contain taxes that modify an existing tax, i.e. where tariff preferences are given.

For example, if a country is a member of a particular trading group, (such as Tax '02', 'EEC', used in this example), a lower rate of tax may be payable on imports from a country within that preference group. When a declaration is input 'Tariff Rules' recognise the country of origin, (from the declaration data), and the System is instructed to calculate the tax within the 'EEC' column, which replaces the normal rate of tax under the 'Customs Duty' column.

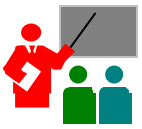
Creating Taxation Columns

- **Column Label** - is the column title. It contains the name of a tax. Insert column labels in the order in which they are to appear in the tariff on-screen display.



The maximum number of columns is 99. Columns 1 to 15 are set up in **MODCHQCF** and additional columns are created in **MODCHQ**. See Section 10 of this Reference Document for more details on creating additional columns.

- **Priority** - assigns a priority to the tax, according to the sequence in which the taxes within the columns are calculated. In setting priorities, values between 01 and 99 are acceptable.



Certain taxes include other taxes in their Tax Base. An example is VAT (Value Added Tax), which would include the Customs Duty in the VAT tax calculation. ASYCUDA ++ must calculate the Customs Duty first, so 'Customs Duty' is given a higher priority than VAT.

Important: To avoid damaging an existing tariff structure, any additional taxes must be added at the bottom of the list, after existing taxes. **Extra care must be taken when deleting columns**.

Taxation Rates

Taxation column boxes contain the rates of taxes applying to a Commodity Code. Rates are in a **numeric format** (e.g. 10), or as a **General Note** such as "VAT" or "EX", instead of a numeric tax rate. The use of codes to represent commonly used tax rates is explained below in **"Common Rates of Tax"**.

Tax rates are placed directly into the taxation columns through **MODCHQ.**

How the System Accesses Taxation Rates

Using Rules to modify or update existing taxation calculations means that ASYCUDA needs to know the original amount or rate that was calculated. Rates can be 'looked up' by a Rules Function such as **Rate(Commodity Code).**

When rates are looked up by the Function Rate(Commodity Code) or similar, the System uses the rate that is in the column to which the rate is attached, i.e. rate 01, 02, 03, 04 etc. Rates are aligned to the column order, or place in the National Tariff List. For 'recognising' rates, the column description or headings is of no relevance.

Taxation Rules

Rates input into taxation column boxes are activated by Taxation Rules. The linking of a Taxation Rule to each column defines

- When the rate in that column is applied (conditions such as for a certain CPC or group of CPC's, like "4" for imports CPC's);
- Mode of payment of the tax due or to be deferred;
- The tax base;
- The tax rate, against the Commodity Code in the tariff or a special rate written into the rule; and
- The action to be performed

  - The mode of calculation of the tax;
  - The modification of existing taxes (including calculation of loss in revenue); or
  - The deletion of an existing tax.

In the example below called Rule "Customs Duty", if the model of declaration is "4" (Entry for Home Use) an action called 'DoTax' is carried out. This action calculates the tax using a combination of data from the declaration and from the System Tariff files. The language of **'Rules'** is explained in the section **'Using Rules'** .

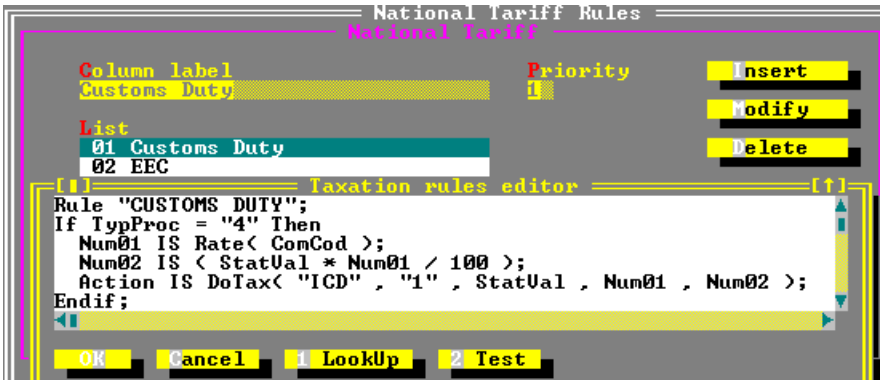
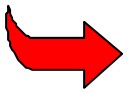


Figure A.4: MODCHQCF: Functions: Taxation Standards: National Tariff: Edit Rule

In this example the effect of Rule "Customs Duty" is to calculate tax, at the rate in the Customs Duty column against the Commodity Code, on the tax base **'StatVal'** (which is calculated from the declaration data through another Rule). This rule will then apply throughout the Tariff unless exceptions are inserted.

For instance, exemptions from Customs Duty can be linked to specific procedures, either in the general rules as above, or in specific rules set up at Procedure Code level.



See later in this Appendix a full list of Keywords, Functions and Variables used in Tariff Rules.

Valuation Note Rules

Calculations of Customs value and the apportionment of freight, insurance and other charges are managed by Valuation Note Rules. Data used in the calculation of Customs value or 'Tax base' is taken from both declaration input and from other reference tables, such as rates of currency exchange. System calculations such as currency conversion and apportioning charges over declaration item lines are handled by Rules.

Valuation Rules are input through **MODCHQCF.** The normal Customs Import and export valuation and apportionment rules are predefined within your system and so no modification of these Rules should be required. The Rules are fully user definable to take account of any national requirements.

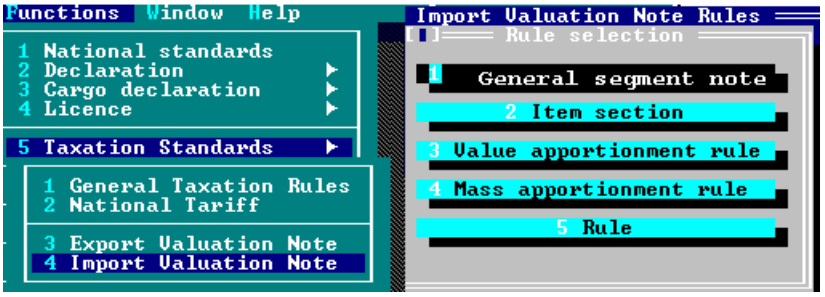


Figure A.5: MODCHQCF: Functions: Taxation Standards: Import Valuation Note

Rule selection options are available under both 'Export Valuation Note' and 'Import Valuation Note'.

The Rule selection window gives you the option to define

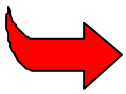
- General segment or declaration header valuation rule.
- Item section rule.
- Value apportionment rule.
- Mass apportionment rule.
- Rule (any other method of apportionment).

Selection of Apportionment Method

After apportionment Rules have been defined, the method of apportionment to be used on a particular declaration can be chosen at the data input stage when the operator is in the Valuation Note screen. Choice of method is available under **<F9> Local Menu** options.

General Segment Note

Under the General Segment Note, a Rule (similar to the predefined Rule below) takes the declaration data that has been input and calculates the invoice value, freight, insurance, other charges and deductions by converting into national currency. The Rule is structured as a series of statements that assign values to variables, and those variables are in turn used for other processing by the system.



A list of variables used by Valuation Note Rules is shown later in this Appendix.

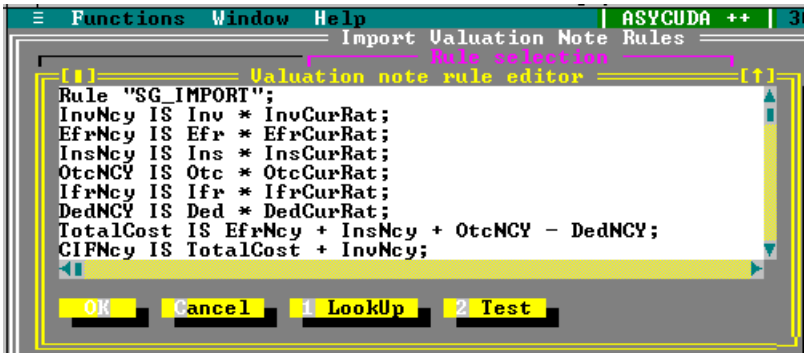


Figure A.6: MODCHQCF: Finctions: Taxtion Standard: Import Valuation Note

Line 1 is the Keyword 'Rule' and Rule name: "SG_IMPORT" and Lines 2 to 7 take declaration data:

Inv	Invoice amount in Foreign currency
Efr	External freight amount in Foreign currency
Ins	Insurance amount in Foreign currency
Otc	Other cost amount in Foreign currency
Ifr	Internal freight amount in Foreign currency
Ded	Deductions amount in Foreign currency

..and multiplies by the currency conversion rates:

InvCurRat	Invoice Currency Exchange Rate
EfrCurRat	External freight Currency Exchange Rate
InsCurRat	Insurance Currency Exchange Rate
OtcCurRat	Other cost Currency Exchange Rate
IfrCurRat	Internal freight Currency Exchange Rate
DedCurRat	Deductions Currency Exchange Rate

... to give an amount in National Currency, represented by the variables:

InvNcy	Invoice amount in National Currency
EfrNcy	External freight amount in National Currency
InsNcy	Insurance amount in National Currency
OtcNcy	Other cost amount in National Currency
IfrNcy	Internal freight amount in National Currency
DedNcy	Deductions amount in National Currency
DedCur	Deductions currency code

The variable **TotalCost** is used to store the value of freight, insurance, other charges, less deductions, after conversion to the National Currency. Invoice amount and total other costs (**InvNcy + Total Cost)** are added and stored to **CIFNcy** (CIF value in National Currency).

Item Section

The purpose of the **'Item Section'** Valuation Note Rule is to calculate, using declaration data and reference tables, the invoice amounts, freight, insurance, other charges and deductions at the declaration item level. Amounts input from the declaration are converted to national currency, added, and the Customs value (taxbase) is stored to the variable called **ItmStatVal**.

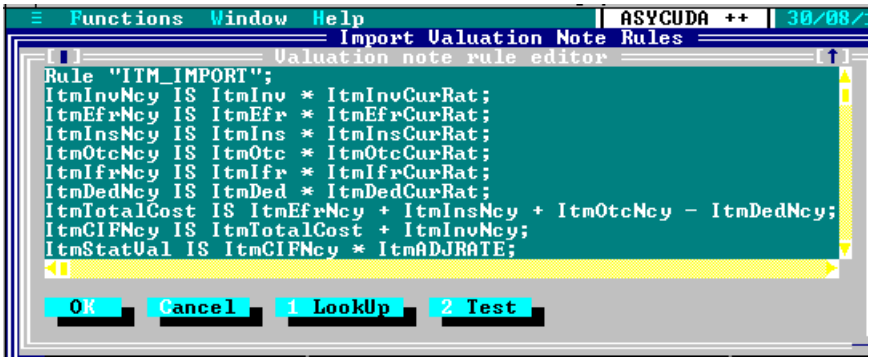


Figure A.7: MODCHQCF: Functions: Import Valuation Note: Item Selection

ItmStatVal (Item statistical value) passes its value to **StatVal** - a variable that represents item statistical value when used in Taxation Rules.

An 'action' under a Taxation Rule, such as

```
DoTax("ICD","1", StatVal, Rate(ComCod),(StatVal * Rate(ComCod)/100));
```

uses the contents of the variable **StatVal** as the TaxBase for tax calculation purposes.

By using this Rule, invoice item amounts are converted into national currency. Any additional taxable value components are converted and added, and the total amount is used as the basis on which the taxation for that item on the declaration is calculated.

Apportionment Rules

When a declaration covers more than one item, ASYCUDA ++ allows for different methods of apportioning taxable amounts such as freight and insurance charges to the Customs value (TaxBase) for each item on the declaration. Rules are predefined to apportion charges either on the basis of value or of mass.

The method of apportionment to be used on a particular declaration can be chosen at the data input stage when the operator is in the Valuation Note screen of the declaration.

Value Apportionment Rule

Valuation Note - (Valuation Apportionment) - Rule "PRV_IMPORT":

The following screen illustrates a typical Valuation Apportionment Rule - in this case, to initial 'standard' Rule as supplied with ASYCUDA++. It can be simply modified to allow for any National variations in Customs Valuation law.

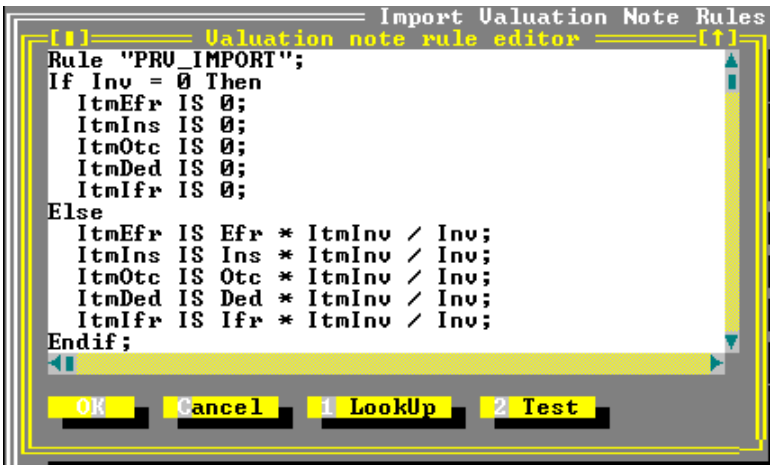


Figure A.8: MODCHQCF: Functions: Import Valuation Note: Value Apportionment Rule

This Rule checks if the Invoice amount is zero, if this is the case a value of zero is given to the Freight, Insurance and Other Charges and to any Deductions. This prevents the system attempting to divide by zero. If the Invoice amount is greater than zero, the apportionment of the Freight, Insurance and Other Charges and any Deductions is done by taking the item invoice amount as a percentage of the total invoice amount (**ItmInv / Inv**).

Mass Apportionment Rule

Valuation Note - (Valuation Apportionment) - Rule "PRV_IMPORT":

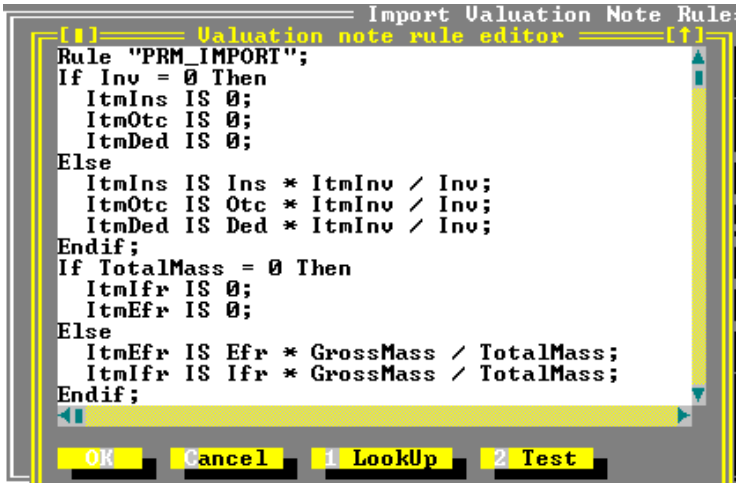


Figure A.9: MODCHQCF: Functions: Import Valuation Note: Mass Apportionment Rule

This Rule apportions External and Internal Freight according to Gross Mass, and apportions the other value elements, Insurance, Other Charges and any Deductions, according to value

This Rule checks if the Invoice amount is zero and if this is the case a value of zero is given to the Insurance and Other Charges and to any Deductions. This prevents the system attempting to divide by zero. If the Invoice amount is greater than zero, the apportionment of the Insurance and Other Charges and any Deductions is done by taking the item invoice amount as a percentage of the total invoice amount (**ItmInv / Inv**).

For External and Internal Freight, the Rule checks if the Gross Mass is zero, and if this is the case a value of zero is given to the External and Internal Freight charges. This prevents the system attempting to divide by zero. If the Gross Mass is greater than zero, the apportionment of the Export and Import Charges is done by taking the Item Gross Mass as a percentage of the Total Gross Mass for the declaration.

Global and Item Rules

The Global and Item Rules define taxes to be applied at the highest level, usually taxes independent of any declaration Procedure Code or Commodity Code classification. Global and Item Rules are set up through **MODCHQCF**. They are accessed through menu choices '**Functions', 'Taxation Standards', 'Global'** or **'Item Rules', 'Edit Rule'.**

Example 1:

```
Rule "FEE";
Action is DoTax("FEE" , "1", ItmTotal, 2, ItmTotal*2) ;
```

In this example an Administration has imposed a computer-processing fee on all declarations processed through the ASYCUDA++ System. The Rate for the fee has been set at a flat fee of two National Currency Units (NCU – francs, dollars, etc) for each item line on the declaration processed.

The number of items **ItmTotal** (General Segment: Total number of items) is multiplied by 2 NCU. This is output on the declaration as an additional amount payable.

Example 2:

```
Rule “Global” ;
If TypProc < “7” then
Action Is DoTax("CST","1",InvNcy,0.5,InvNcy * 0.5/100);
Endif ;
```

This is an example of a similar fee, but instead of a fixed amount it is based on the value of the goods. The Rule is to collect a general tax on all declarations processed, and is set at a rate of 0.5% of the invoice amount of the declared goods.

Explanation:

Line 1 starts with the Keyword 'Rule' and the name of the Rule : "GLOBAL"

Line 2 sets the scope of the tax. It is a Customs Stamp Tax that applies to all declarations with a Type of Declaration less than 7.

Line 3 specifies the action to be carried out in the processing. It uses the Rules Function DoTax()

```
Action Is DoTax( "CST", "1" ,InvNcy, 0.5, InvNcy * 0.5/100
```

The function **DoTax("TaxCod", "MP", TaxBase, Rate, Amount)** carries out the following action

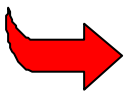
It recognises **"CST" - "TaxCod"** (Tax Code) - as Customs Stamp Tax.

It recognises from **"1" -** the Method of Payment (**MP**) - that the tax is to be paid, rather than "0", which means that it is not payable.

InvNcy is a variable that represents the **TaxBase** calculated (by a Valuation Rule) from declaration data.

0.5 is the **Rate** of tax and is input directly into the Rule.

InvNcy *0.5/100 is the calculated **Amount** of tax, using the TaxBase and the Rate of tax.



See the section Control Structure in Rules for a further explanation of the Rules language.

In the above examples, for both Rules "Computer Fee" and "GLOBAL", a low Priority is set. By giving these Rules the priority number "90" it means that most other tax calculations are carried out before these two General Application Rules are processed.

Customs Headquarters

Customs Headquarters Module **MODCHQ** is used to enter tax rates into the taxation structure at Commodity Code level and for updating rates. It is also used to define the rules for specific taxation cases and to define General Notes representing rates of tax.

General Notes

General Notes or 'Common Rates of Tax' are used to create a code that is inserted into the Taxation column box, in place of a numeric value. It is used when a certain common rate of tax applies to a wide range of commodities, and this common rate may be periodically changed.

Example:

A Value Added Tax of 5% may apply to most commodities in the tariff. The 5% rate could be entered against each Commodity Code under the VAT column. If the rate is increased to 6%, each Commodity Code line with "VAT" "5%" must be updated.

If instead a code is used to represent the numeric value of the VAT rate, and this code is inserted into each Commodity Code line under the "VAT" column, then changing the numeric value of the code from "5" to "6" updates all of the Commodity Code lines containing the code.

Similarly, when a Commodity is exempt from a particular tax, the code is given the value of 0.00.

The example below shows code "ST" for Stamp Tax with a rate of 3%.

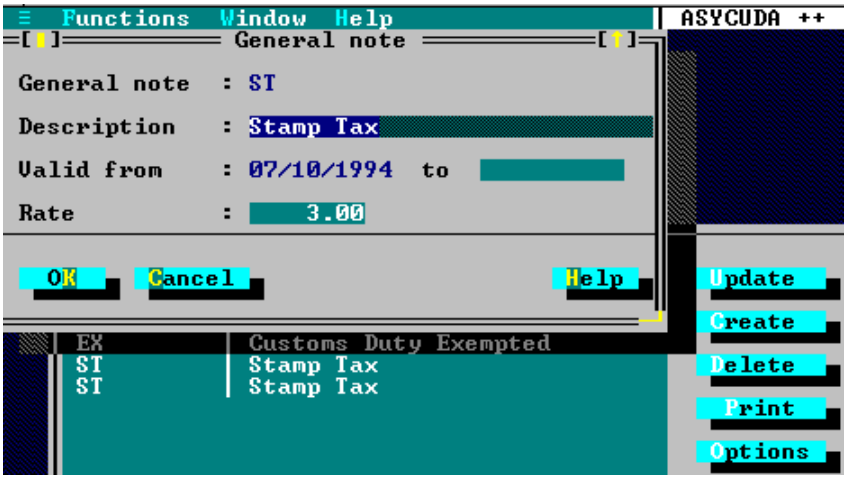


Figure A.10: MODCHQ: Functions: Datbase Management: Customs Tariff: Tariff Components: General Note

Taxation Processing

General segment level

To calculate taxes on a declaration the taxation process retrieves and performs the applicable rules from the national configuration. These include 'General Application Rules' such as taxes to be applied at a high level, independent of any declaration procedure or commodity, and 'Valuation Rules', where the tax base is calculated from declaration data.

Declaration item level

The taxation process retrieves the list of applicable rules from

- The national configuration for every tax column in the tariff;
- The table of General notes;
- The table of Taxation Rules; and
- Sorts the retrieved rules according to priority, and performs them in that order.

Rates of Taxation

Rates of taxation are input directly into the taxation columns for each Commodity Code. The Commodity Codes are accessed through **MODCHQ**, menu options

MODCHQ, 'Functions', 'Database Management', 'Customs Tariff', ' Commodity Code'.

Inserting Rates of Taxation

To add rates of taxation or alter existing rates, select the Commodity Code that you want to change. The Commodity Code can be selected by either scrolling through the codes on screen, or by using the **'Options'** button to more quickly access particular Commodity Codes. Select the **'Update'** button to view the Commodity Code details.

This is an example screen for Commodity Code 902710.

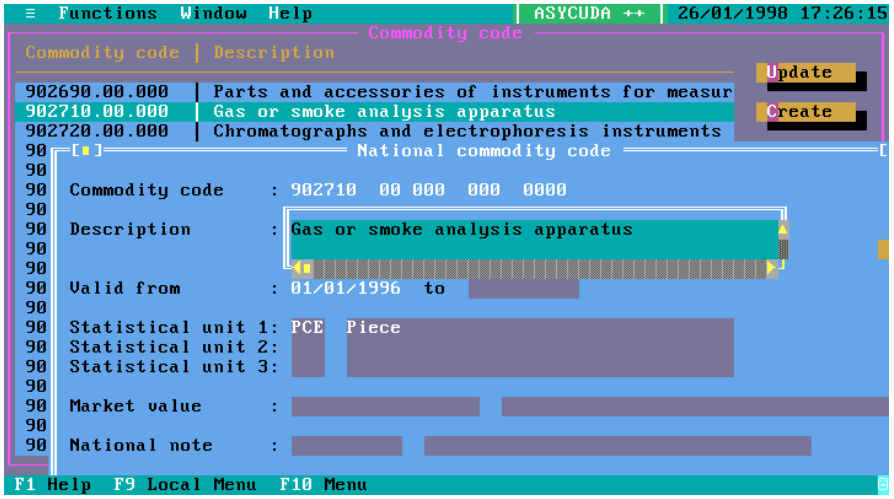


Figure A.11: MODCHQ: Functions: Database Management: Customs Tariff: Commodity Code

Insert Validity Dates "from .... to" in the format dd/mm/yyyy.

Important: See the Section on 'Validity Dates'.

The bottom part of the Commodity Code screen appears as follows

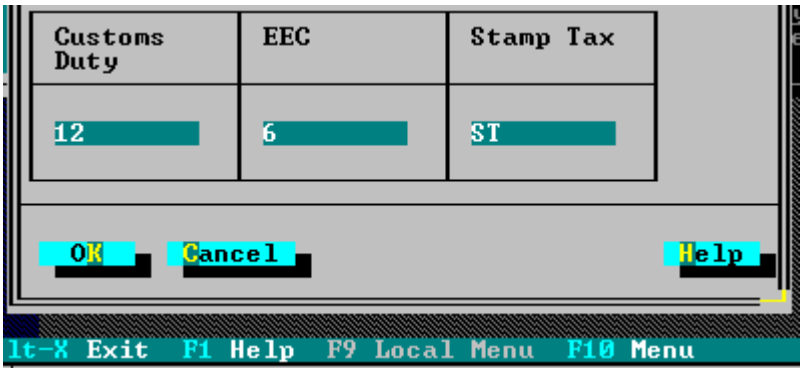


Figure A.12: MODCHQ: Functions: Database Management: Customs Tariff: Commodity Code

You can alter existing rates or insert new rates directly in the tax column. Each change or addition of rate adds a new Commodity Code taxation line to the table. Taxation lines added in this manner can only be deleted if the taxation line is not valid and has never been valid, i.e. the taxation line record has never become 'current' as regards the start validity date. See 'Validity Dates' below.

Tip: for development purposes, 'Validity Dates' for trial changes to taxation lines should be sufficiently in the future to not become current.

Validity Dates

For database integrity and to maintain a historical record of changes in taxation rates the ASYCUDA++ System requires that any new rates or changes must be from at least the next day.

The system will not allow changes to, or deletion of, records that have a current validity date, or were current records for a period in the past.

Changes of rates have the effect of inserting an additional Commodity Code line into the Tariff table, which become active next day or at a chosen date in the future. The alteration 'closes off' the existing Commodity Code taxation line by inserting a closing validity date.

Simplified Data Management

Default Taxation Rate Values

To simplify data capture and the display of taxation rates, only the applicable rates are present and displayed. As an example, a country may have one basic tax rate, General, for Customs Duty and a number of specific rates, limited to determined Commodity Codes for specific Tariff preference.

In this case, only the applicable rate need appear, i.e. if a specific preference rate does not exist, the basic rate remains applicable.

Example:

Rates as follows -

Note: General note "EX" corresponds to a duty exemption (rate = 0)

Commodity Code	Customs Duty (Basic Rate)	Preference Rate 1	Preference Rate 2	Preference Rate 3
010201	10 %	5 %	Basic rate	Basic rate
010202	10 %	Basic rate	Basic rate	Basic rate
010203	15 %	10 %	10 %	8 %
010204	10 %	EX	Basic rate	Basic rate
010205	10 %	Basic rate	EX	EX
010206	20 %	12 %	Basic rate	Basic rate

can be input (and will be displayed) as

Commodity Code	Customs Duty (Basic Rate)	Preference Rate 1	Preference Rate 2	Preference Rate 3
010201	10 %	5 %		
010202	10 %			
010203	15 %	10 %	10 %	8 %
010204	10 %	EX		
010205	10 %		EX	EX
010206	20 %	12 %		

Customs Tariff Management

To add or change rates of taxation (see **'Inserting Rates of Taxation'**) means individually accessing each relevant Commodity Code. This is a time consuming process, particularly if you need to insert or alter similar rates across a broad range of Commodity Codes.

To change the details for a number of Commodity Codes using common data, such as a rate of 20% for Import Customs Duty across an entire Tariff Chapter, you can select and update the rate in that Chapter in one operation, as a block. Other data elements input through the Commodity Code screen can be updated as blocks, in a similar fashion.

To add or change taxation over a range of Commodity Codes select:-

MODCHQ, 'Functions', 'Database Management', 'Simplified Management', 'Customs Tariff' which displays this screen

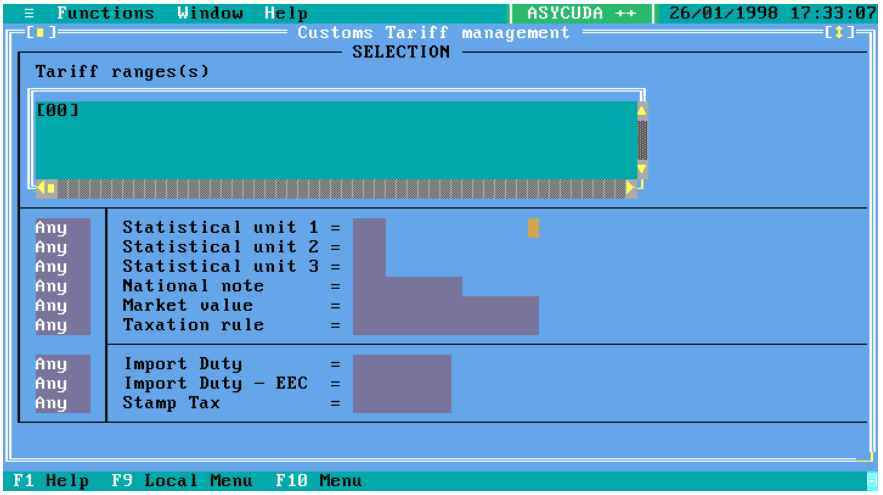


Figure A.12: MODCHQ: Functions: Database Management: Simplified Management: Customs Tariff

Defining The Tariff Range

To make any update by inserting or altering data in a block of Commodity Codes, first specify the Commodity Code range. The syntax of the "Tariff ranges<s>" editing box is simply explained as follows - The Tariff Code must be enclosed in square brackets, for example -

[8701] - specifies all Commodity Codes within heading 8701.

The Commodity Code coverage is from a two digit level to eleven digits; for example, at Chapter level, like **[87]**, or as specific as the lowest level national subdivision, such as **[87011010000]**.

Tariff ranges can be used but the syntax requires that the ranges be defined in pairs, using a comma as a separator, for example -

[01, 05] - selects all ComCodes in Chapters 1 to 5.

[5501, 550110] - selects all ComCodes for Heading 5501 up to and including Subheading 550110.

Multiple selections can be made, either of ranges or by combining single selections, for example –

[01] + [03] + [040210] - selects chapters 1 and 3 and sub-heading 040210.

[8501, 8506] + [91] selects all ComCodes for heading 8501 to heading 8506, plus all of chapter 91.

Excluding part of the range initially selected can further refine range selections. The following example illustrates this point, for example -

[55] - [5503] selects all the ComCodes for Chapter 55 except ComCodes of heading 5503.

Additional Selection

To further refine or restrict the range of Commodity Codes upon which to carry out the action choose from the Additional Selections. This permits a more precise definition within the chosen Tariff range, using a number of criteria apart from the Commodity Code.

Update 'Actions'

After making any **'Additional Selection'**, specify the **'Action'** to be carried out on the selected range. **'Actions'** allow the addition, deletion or replacement of Tariff related components. These are the Tariff components i.e. Statistical Unit, Market Value, Rule, National Note, Taxation/Duty rates, that are directly inserted or amended through the Commodity Code screen.

After planning the extent of the update, an appropriate **'Action'** is selected by using the space bar to cycle through the available options.

Note: 'Planning' is stressed, because of the potential difficulty in reversing an ill-considered and incorrect update.

Taxation Rules for Special Circumstances

Taxation is normally calculated by the System referencing the Taxation Rule linked to the appropriate Taxation column in the Tariff.

These Rules and links are set up in **MODCHQCF** and use functions like DoTax( ) and RelTax( ) to perform 'standard' calculations. For more detail see **'Taxation Rules'** and **'Using Rules'**.

In certain circumstances, either for specific Commodities or a number of other reasons, you may require taxation to be calculated by another method, such as using the number of pieces or the net mass as the TaxBase, or you may want to apply a special relief to imports from a specified country.

You can implement these exceptions to 'standard' taxation by several methods by writing your Taxation Rule in **MODCHQCF**, linked to the taxation column, using more complex statements, i.e. "If ...Then ... Else ... Endif", "and", "not", etc.

Special Taxation Rules in **MODCHQ** can be used to calculate tax according to specific national requirements. This can be done by inserting these Rules, in the Tariff at Commodity Code level, by linking them to CPC, through the CPC **'Additional Code'**, in an Agreement code, or by linking them to a Country or to a Preference Code. The **'Market Value'** screen can also accept a Rule.

Writing Special Taxation Rules

Special Taxation Rules are written in module **MODCHQ.** See the section on **"Using Rules"** and Section 10 of this Reference Document.

Tax Rules in Commodity Code

Rules may be written to cover special taxation situations where a National policy is to vary tax calculation within a specific tariff classification, or within limited ranges of classifications. For example – if the taxation rule linked to the relevant taxation column normally calculates tax at an ad valorem rate, specific Commodities may be taxed at a fixed rate, (e.g. currency per unit weight), with the 'normal' method of calculation replaced for that specific Commodity.

Note: The extent of use depends on the taxation policies of the country implementing ASYCUDA++. In designing the taxation rules, careful consideration needs to be given to the efficiency and maintenance that the different approaches give. If the taxation for the majority of Commodities is based on a value tax base, it is simpler to have the taxation column rule calculate on a percentage rate and then negate this calculation by attaching specific tax rules to each Commodity Code.

The Rule is written in **MODCHQ**, under **'Functions', 'Database Management', 'Tariff', 'Rule'** and saved. The rule is **'Inserted'** by adding the rule name in the **'Rule'** field on the Commodity Code Update screen.

Tax Rules in Customs Procedure Codes

In a similar manner to Rules attached to the Commodity Code, you can write Rules to modify the means of calculating taxation or the amounts of taxation according to specific CPCs.

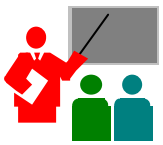
Note: This is the normal method of implementing reduced rates of taxation on goods processed under concessional CPCs.

The Rule is written in **MODCHQ**, under '**Functions', 'Database Management', 'Tariff', 'Rule'** and saved. The rule is **'Inserted'** by adding the rule name in the **'Rule'** field on the CPC Update screen.

Tax Rules used with Agreement codes

Agreements can be used in two different ways to achieve different types of relief.

General - If the relief is available to certain Companies irrespective of the CPC, an appropriate rule can be inserted in the Agreement for the relief to be granted.



An example of this would be a VAT relief that is available only to registered Companies. If a rule exempting VAT were inserted in to the Agreement then no matter which Procedure code was used, a Company and Commodity included in the Agreement would be relieved of VAT. This is because no link to procedures is defined and the system checks the Agreement table for all declarations to check if the importer is eligible for some sort of relief.

Specific - If a relief under a CPC Additional Code is to be limited to certain Companies or Commodities then the Agreement record is set up in a different way. No rule is included in the Agreement but the Agreement itself is linked to the Additional Code that contains the rule giving the relief.

The Additional Code record has an Agreement box **'Yes/No'** box. The Agreement record has a box to insert the name of the appropriate Additional Code. Only if the importer requests the relief by declaring the Additional Code does the system check that the importer or goods are eligible for that relief. If they are not, the system will not accept the Additional code declared and gives an error message that the Company or Commodity are not authorised to receive the relief.

Tax Rules in Countries and Preference Codes

In a similar manner to Rules attached to Commodity Code you can write Rules to modify the means of calculating taxation or the amounts of taxation according to specific Country or Preference Codes. The Rule is written in **MODCHQ**, under '**Functions', 'Database Management', 'Tariff', 'Rule'** and saved. The rule is **'Inserted'** by adding the rule name in the **'Rule'** field on the Preference Code Update screen.

The normal method for dealing with preferences is to set up a tariff column in **MODCHQCF**. These options in Country or preference code should be used only in special circumstances were a very simple preference regime is involved.

Tax Rules in Market Values

The Market Value update screen has provision for including a rule if the normal functions fail to provide sufficient flexibility. For example if different values are required depending on the country of origin. (See Section 10 of this Reference Document for the **'Market Value'** function).

The Rule is written in **MODCHQ**, under **'Functions', 'Database Management', 'Tariff', 'Rule'** and saved. The rule is **'Inserted'** by adding the rule name in the **'Rule'** field on the Market Value Update screen.

Summary:

The Tariff structure is built and calculation of taxes is managed by **MODCHQCF** and **MODCHQ** by

- Creating a taxation column for each tax to be linked to Tariff Commodity Codes;
- Writing Taxation Rules to link to taxation columns, giving instructions for when and how to apply the tax; and
- Inserting tax rates into the Taxation Columns.

Taxation Management - Table Structures

Taxation Rules Editor - Syntax

Rule Functions

Functions used for mathematical calculation

Function	Logic
RoundInf(x)	If N x < N + 1 then N
RoundSup(x)	If N < x N + 1 then N + 1
Round(x)	If N x < N + 0.5 then N else N + 1
Min(x, y)	If x < y then x else y
Max(x, y)	If x > y then x else y
Sqr(x)	Square x * x
Abs(x)	Absolute value of x
JulianDate(s)	Convert date to julian number

RoundInf(x)

Logic:	if N x < N + 1 then N
Syntax:	
Purpose:	Round a decimal value to the lowest whole number
Example :	If the tax amount is 45.55 the RoundInf function will store and display 45.

RoundSup(x)

Logic:	if N < x N + 1 then N + 1
Syntax:	
Purpose:	Round a decimal value to the highest whole number
Example :	if the tax amount is 45.55 the roundsup function will store and display 46 if the tax amount is 45.45 the roundsup function will store and display 46

Round(x)

Logic:	if N x < N + 0.5 then N else N + 1
Syntax:	
Purpose:	Round a decimal value to the nearest giving a whole number.
Example :	if the tax amount is 45.55 the round function will store and display 46. if the tax amount is 45.45 the round function will store and display 45.

Min(x, y)

Logic:	if x < y then x else y
Syntax:	
Purpose:	to select the smallest of two quantities
Example :	

Max(x, y)

Logic:	if x > y then x else y
Syntax:	
Purpose:	to select the largest of two quantities
Example :	

Sqr(x)

Logic:	square x * x
Syntax:	
Purpose:	to multiply a value by itself
Example :	

Abs(x)

Logic	absolute x
Syntax:	
Purpose:	absolute value of x
Example :	

JulianDate(s)

Logic	Convert date to julian number
Syntax:	JulianDate(YYYYMMDD)
Purpose:	To transform a date into an integer which can be used to count the number of days between two dates.
Example :	Num01 is Juliandate(20000112) - Juliandate(19991123). This functions stores in Num01 the number of days between 23 November 1999 and 12 January 2000.

Functions used for tax rate retrieval

Function	Logic
Rate(s)	Rate for commodity s
Ratecol(s,x)	Rate for commodity s and column x

Rate(s)

Logic	Rate for commodity s
Use:	This function must be written in a Customs Tariff rule in **MODCHQCF**. In such a case the line number of the rule gives the number of the column where the appropriate rate will be found.
Purpose:	Retrieve a tax rate directly from the Customs Tariff.
Example :	Path **MODCHQCF/Functions/Taxation standards/National Tariff:** 1 Customs duty 1/3 Country 2 Customs duty preference #1 3 Customs duty preference #2 Rate(Comcod) means retrieve from the Customs Tariff the rate for the Commodity Code. In the rule for line number 2 this function retrieves the rate in the first column of the Tariff. In the rule for line number 3 this function retrieves the rate in the second column of the Tariff.

Ratecol(s)

Logic	Rate for commodity s and column x
Use:	Can be used in any rule, as the number of the column is specified in the rule and it permits the appropriate rate to be found.
Purpose:	Retrieve a tax rate directly from the customs tariff.
Example :	Ratecol(Comcod, 3) means retrieve from the Customs Tariff the rate in the third column for the Commodity Code.

Functions used for tax calculations

These functions perform the tax calculation and in all cases are preceded by the Keyword **'Action'**.

Function	Logic
AskTax(...)	Input tax data
DoTax(...)	Create ("TaxCod", "MP", TaxBase, Rate, Amount
UpdTax(...)	Update ("TaxCod", "MP", TaxBase, Rate, Amount)
RelTax(...)	Relieve("TaxCod", "Relcode", TaxBs, Rate, Amount)
RelTaxPart(...)	Relieve Part ("TaxCod","Relcode",TaxBase, Rate, Amount)
DelTax(...)	Delete ("TaxCod")
DelAllTax(0)	Delete all taxes belonging to a declaration

AskTax(..)

Logic	User input of some or all of the Tax data.
Syntax:	Action is AskTax("Tax code","MP",Tax Base, Tax Rate,Tax). **Note**: use of **'-1'** in place of base, rate or tax allows user input.
Purpose:	This function may be used in **MODCHQ** or **MODCHQCF** to allow the details of a complex tax to be manually input. The AskTax function is used at the Item level or for Global taxes. Up to 3 manual taxes can be created for Global taxes.
Example:	AskTax("TXC", "1", -1 , -1 , -1). This action will open a window for the manual input of the tax base, rate and calculated tax. **Note:** To input just the base and/or rate and have the system do the calculation, follow the AskTax with an UpdTax statement, described below, to recalculate the tax amount using the details from the AskTax.

This is the input window generated by an AskTax statement

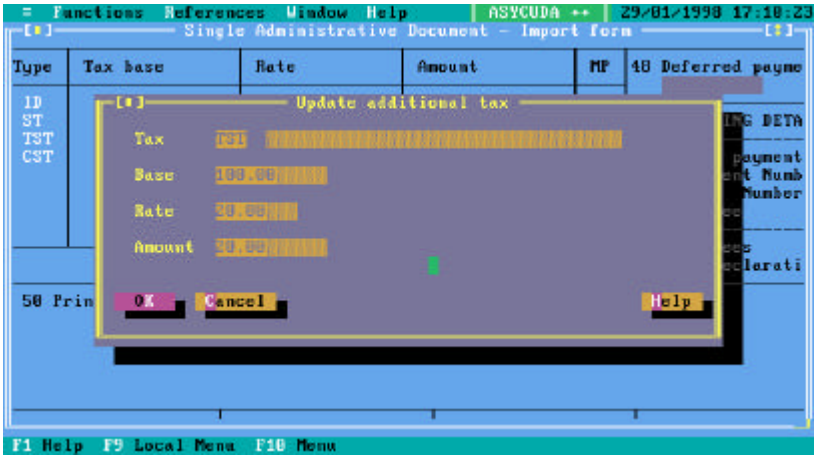


Figure A.13: MODCBR: Declaration input

Dotax(..)

Logic	To calculate tax.
Syntax:	Action is DoTax("Tax code","MOP",Tax Base, Tax Rate,Tax)
Purpose:	Calculates, charges and displays the tax.
Example:	DoTax("ICD", "1", Statval , 10 , Statval * 10/100). This action will calculate and display the tax ICD, to be paid (MP 1), tax basis is StatVal, rate is 10%, tax is 10% of StatVal. **Note:** If a tax has already been calculated and a "Dotax" is requested for the same tax code, the Dotax function recalculates it.

Updtax(..)

Logic	Update a previously calculated tax.
Syntax:	Action is UpdTax(“Tax code”,”MP”,Tax Base, Tax Rate,Tax).
Purpose:	Used to modify the calculation of an existing tax and display the new results.
Example :	UpdTax("ICD", "1", Statval , 5 , Statval * 5/100); This action will modify the tax “ICD”, which has to be paid (MP "1"), the tax basis is StatVal, rate is 5%, tax is 5% of StatVal. **Note:** Nothing is done if this action is requested on a non-existent tax.

Reltax(..)

Logic	Relieve an existing tax.
Syntax:	Action is RelTax(“Tax code”,”Relief code”,Tax Base, Tax Rate,Tax).
Purpose:	Reduces the calculation of an existing tax, displays the modified results and stores the difference in the database (Loss of revenue).
Example :	RelTax("ICD", "PR1", Statval , 5 , Statval * 5/100); The calculation of tax “ICD” is modified, MP is “1” and tax basis is StatVal, rate is 5%, tax is 5% of StatVal. This function is different from Updtax because it stores the reduction of tax amount (loss of revenue) under the preference code PR1. The loss of revenue can then be retrieved for any purpose (fiscal or statistical). In the example the loss of revenue is stored under the preference code "PR1". **Note**: Nothing is done if this action is requested on a non-existent tax.

RelTaxPart(...)

Logic	Partially relieve a previously calculated tax.
Syntax:	Action is RelTaxPart(“Tax code”,”Relief code”,Tax Base, Tax Rate,Tax).
Purpose:	This function is similar to RelTax. It permits two different losses of revenue to be calculated and combined. E.g. goods are imported from a country that is a member of a preference group. They benefit from a reduced rate of duty. The consignee may also be eligible to a reduced rate under a specific Customs regime. The two losses of revenue are applicable and can be stored separately.
Example :	RelpartTax("ICD", "PR1", Statval , 5 , Statval * 5/100); RelpartTax("ICD", "IVP", Statval , 2.5 , Statval * 2.5/100); where PR1 is the relief code for the preference and IVP the relief code for the Customs regime. Note: Nothing is done if this action is requested on a non-existent tax.

DelTax(...)

Logic	to delete a calculated tax
Syntax:	Action is Deltax("Tax Code")
Purpose:	This function will physically delete the tax from the taxation database, it will neither be stored nor displayed.
Example :	Action is Deltax("ICD"). The tax ICD will disappear from the declaration and from the database. **Note:** The same result can be achieved with the following line: UpdTax("ICD", "PR1", Statval , 0 , 0), and this line is displayed on the declaration and the officer will know that a tax has been deleted.

DelAllTax(0)

Logic	Sets all taxes on declaration to 0.
Syntax:	Action is DelAlltax(0).
Purpose:	When used in **Global** rules this function will set all calculated Duties and Taxes on the declaration to 0. Must have highest priority. The Zero is mandatory.
Example :	Action is DelAlltax(0). All taxes will be displayed with the normal base and rate shown but the tax amount will be zero.

Retrieving previous tax details

These functions are used to retrieve the elements of taxes previously calculated and to be used in the current tax, **the first element in the brackets must be a Tax Code**. In the case of a tax update the system always retrieves the latest value (i.e. after update the value is the new value).

Function	Logic
TaxMP(s, t)	Get ("TaxCod", "MP")
TaxBasisVal(x)	Get ("TaxCod")
TaxRate(x)	Get ("TaxCod")
TaxAmount(x)	Get ("TaxCod")
CurRate(s)	Exchange Rate for currency s

TaxMP(s, t)

Logic	Get the Means of Payment Code for the Tax Rule
Syntax:	
Purpose:	This permits the mode of payment of an existing tax to be retrieved (egg for selection purposes);
Example:	TaxMP("VAT", "1")

TaxBasisVal(x)

Logic	Get the Tax Base as calculated by the Tax Rule
Syntax:	
Purpose:	Retrieves the tax basis of an existing tax (e.g. to be included in the basis of another tax):
Example :	Num01 is TaxBasisVal("EXC"); Num02 is STATVAL + NUM01.....

TaxRate(x)

Logic	Get the Tax Rate as calculated by the Tax Rule
Syntax:	
Purpose:	Retrieves the tax rate of an existing tax (e.g. recalculate the tax or compare to any other):
Example :	Num01 is TaxRate("ICD");

TaxAmount(x)

Logic	Get the Tax Amount as calculated by the Tax Rule
Syntax:	
Purpose:	Retrieves the amount of an existing tax (e.g. to be included in the basis of another tax).
Example:	Num01 is TaxAmount("ICD"); **Note:** A variable called TaxItmAmount (Tax Item Amount) is the total of duty and taxes for the item.

Currency rate

CurRate(s)	Exchange Rate for currency s
Logic	
Syntax:	
Purpose:	Retrieves the exchange rate of a Foreign Currency from the control table. Used to calculate taxes in National Currency even if the rule states that a tax amount is in Foreign Currency.
Example:	Cigarettes are dutiable at a rate of 5 USD per ten boxes.

Attached document management

These functions manage the documents requested for the particular taxation case.

Function	Logic
AddAttDoc(x)	Add("Attached document code")
DelAttDoc(x)	Delete("Attached document code")

AddAttDoc(x)

Logic	
Syntax:	
Purpose:	This function displays on the declaration the document code requested for the specific taxation case.
Example:	In case of a preferential trade agreement the reduced taxation requires a Certificate of origin If prefercod = "EU" then Action is Updtax... Action is AddAttDoc("861");

DelAttDoc(x)

Logic	
Syntax:	
Purpose:	This function cancels the need to attach a document for the particular taxation case.
Example:	In the case of preferential trade agreement the reduced taxation requires a Certificate of origin except for values less than 1000 NCU. If value < "1001" then Action is DelAttDoc("861");

Listing Commodity Codes for rules

Function	Logic
InListTar(x)	**In list Tariff (“List name”)**

InListTar

Logic	
Syntax:	If InListTar(“List Name”) = 0 Then
Purpose:	Used to simplify rule management in MODCHQ by allowing lists of commodities to be maintained outside the rule in a separate list.
Example :	If only a limited number of commodities are eligible for a particular relief then you could include them in the rule with repeated If ComCod = “ x ” statements. This is difficult to maintain and it is much easier if the Commodity Codes are held in a list and the rule merely uses one Inlist statement.

Changing the working date

Function	Logic
SetWrkDate(x)	**Sets working date (x julian date)**

SetWrkDate(x)

Logic	Uses a date other than the current system working date
Syntax:	Action is SetWrkDate (Date)
Purpose:	Used to change the date for calculation purposes.
Example:	If a tax has to be calculated using the tax and currency rates applicable on a date other than the current date then the working date can be set using this function.

Displaying Screen Messages

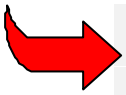
Function	Logic
DisplayErr(s)	Display the string on the screen

DisplayErr(s)

Logic	Displays a specified string on the screen
Syntax:	Action is DisplayErr(" Message")
Purpose:	This function will display the text sting on screen for the user.
Example :	If in the case of a special Car tax each car must be on a separate SAD form then the following line in the appropriate rule would cause the message to be displayed Action is DisplayErr ("Only one Car per Declaration").

Rule Variables

The following list is available for **MODCHQ** rules. This is the most comprehensive list. Other rules have access to a more limited list. For example, in **MODCHQCF** Global rules there is no access to Item variables. The purpose of Global rules is for the whole declaration and not for item level.



See the full up-to-date listing for each type of rule variables by using the local menu look up within the rule-editing window concerned.

Variable	Section	Description
InvNcy	Valuation note	Invoice amount in National Currency
EfrNcy	Valuation note	External Freight amount in National Currency
InsNcy	Valuation note	Insurance amount in National Currency
OtcNcy	Valuation note	Other cost amount in National Currency
IfrNcy	Valuation note	Internal freight amount in National Currency
DedNcy	Valuation note	Deductions amount in National Currency
TotalCost	Valuation note	Total insurance, freight etc costs
CIFNcy	Valuation note	CIF Value in National Currency
ItemInv	Valuation note	Item Invoice amount in Foreign currency
ItemInvNcy	Valuation note	Item Invoice amount in National Currency
ItemEfrNcy	Valuation note	Item external freight amount in National Currency
ItemInsNcy	Valuation note	Item insurance amount in National Currency
ItemOtcNcy	Valuation note	Item other cost amount in National Currency
ItemIfrNcy	Valuation note	Item internal freight amount in National Currency
ItemDedNcy	Valuation note	Item deductions amount in National Currency
ItemTotalCost	Valuation note	Item total insurance, freight etc costs
ItemCIFNcy	Valuation note	Item CIF Value in National Currency
ValMethod	Valuation note	Valuation method field from box 43 of the SAD
CuoCod	General Segment	Customs Clearance Office code
Exporter	General Segment	Exporter code
TypDec	General Segment	Type of declaration
TypProc	General Segment	Declaration General Procedure code
ItemTotal	General Segment	Total number of items
PackTotal	General Segment	Total number of packages
Consignee	General Segment	Consignee code
Financial	General Segment	Person responsible for financial settlement code
ValDetails	General Segment	Value details
CAPRef	General Segment	Common Agricultural Policy reference
Declarant	General Segment	Declarant code
CtyDestCod	General Segment	Country of Destination code
NatMotDptArr	General Segment	Nationality of means of transport at depart or arrival
CtnrFlag	General Segment	Container flag
TodCod	General Segment	Terms of delivery
NatMotBorder	General Segment	Nationality of means of transport at the border
TotInvoice	General Segment	Total amount Invoiced (Foreign currency)
TraCod1	General Segment	Nature of transaction code 1
TraCod2	General Segment	Nature of transaction code 2
BankCode	General Segment	Bank code
MoTBorder	General Segment	Mode of transport crossing the border
MoTInland	General Segment	Inland Mode of Transport
CuoBord	General Segment	Border Customs Office code
LocGoods	General Segment	Location of goods
BnkBranchCod	General Segment	Bank – Branch code
BnkFileNber	General Segment	Bank – File Reference Number
DeferedPay	General Segment	Deferred payment reference
WhsCod	General Segment	Warehouse Identification Code

Variable	Section	Description
WhsDelay	General Segment	Warehouse Delay time (Days)
SealsNber	General Segment	Seals Affixed – Number
TrstTimeLimit	General Segment	Transit time limit (date)
ItmNber	Item Level	Item Number
ComCod	Item Level	HSTariff Commodity Code + National Precision (8 + 3)
HSPrec2	Item Level	Commodity Code (National Precision # 2)
HSPrec3	Item Level	Commodity Code (National Precision # 3)
HSPrec4	Item Level	Commodity Code (National Precision # 4)
HSSpecif	Item Level	Commodity Code Specification
PackNber	Item Level	Number of packages for item
PackKindCod	Item Level	Kind of packages code
CtyOrigCod	Item Level	Country of Origin code
CtyOrigReg	Item Level	Country of Origin region code
GrossMass	Item Level	Gross mass
PreferCod	Item Level	Preference code
ExtdProc	Item Level	Extended Customs Procedure Code
NatProc	Item Level	National Customs Procedure Code
NetMass	Item Level	Net mass
Quota	Item Level	Quota reference
GoodDesc	Item Level	Description of goods
SuppUnits	Item Level	Supplementary Unit Of Measurement - 1
SupValue1	Item Level	Supplementary Unit Of Measurement - 2
SupValue2	Item Level	Supplementary Unit Of Measurement - 3
ItemPrice	Item Level	Item price
LicNber	Item Level	Licence reference number
DValAmount	Item Level	Value deducted from licence
DQty	Item Level	Quantity deducted from licence
AdjRate	Item Level	Rate of adjustment
StatVal	Item Level	Statistical Value
ItemTotMoP	Item Level	Mode of payment for item total amount
Action	Working Number	Free use
Result	Working Number	Free use
Num01	Working Number	Free use
Num02	Working Number	Free use
Num03	Working Number	Free use
Num04	Working Number	Free use
Num05	Working Number	Free use
Num06	Working Number	Free use
Num07	Working Number	Free use
Num08	Working Number	Free use
Num09	Working Number	Free use
Num10	Working Number	Free use
Num11	Working Number	Free use
Num12	Working Number	Free use
Num13	Working Number	Free use
Num14	Working Number	Free use
Num15	Working Number	Free use
Num16	Working Number	Free use
Str01	Working String	Free use
Str02	Working String	Free use
Str03	Working String	Free use
Str04	Working String	Free use
Str05	Working String	Free use
Str06	Working String	Free use
Str07	Working String	Free use
Str08	Working String	Free use

Variable	Section	Description
Str09	Working String	Free use
Str10	Working String	Free use
Str11	Working String	Free use
Str12	Working String	Free use
Str13	Working String	Free use
Str14	Working String	Free use
Str15	Working String	Free use
ItemTotAmount	Working Number	Total amount of Duties and Taxes for the item
CustValue	Working Number	Customs Value (Statistical or Market Value)
TotalStat	Working Number	Total Statistical Values
CurDate	Working Number	Today's date (julian date)
RegDate	Working Number	Registration's date (julian date)
PrvRegDate	Working Number	Previous declaration registration date (julian date)
ItmGarAmount	Working Number	Total guaranteed duties and taxes
ItmPayAmount	Working Number	Total unguaranteed duties and taxes
HSCod	Working String	HS tariff code at 6 digit level
Chapter	Working String	Tariff chapter at 2 digit level
Heading	Working String	Tariff heading at 4 digit level
NatProc1	Working String	National Procedure Code – Digit 1
NatProc2	Working String	National Procedure Code – Digit 2
NatProc3	Working String	National Procedure Code – Digit 3
NoError	Error	No Error
Err-TaxCdNF	Error	Tax code not found
Err-TaxCdAE	Error	Tax code already exists
Err-8Taxes	Error	Eight taxes already exists
Err-HsCdNF	Error	National Commodity Code not found
Err-TxNtNF	Error	Tax note not found
Err-NoTcRt	Error	No tax rate defined
Err-TxRt_0	Error	Tax rate is lower than 0
Err-TxBs_0	Error	Tax basis is lower than 0
Err-TAmt_0	Error	Tax amount is lower than 0
Err-AtDCAE	Error	Attached document code already exists
Err-2MnyAD	Error	Too many attached document codes already exist
Err-AtDcNF	Error	Attached document code not found
Err-Date	Error	Incorrect date format
Err-RulLk	Error	Rule not linked to a tariff column
Err-LocalDB	Error	Paradox access error

Keywords.

Here is a list of the keywords that are used for writing rules.

Keyword	Usage
If	If condition Then statements [Else statements] Endif
then	If condition Then statements [Else statements] Endif
else	If condition Then statements [Else statements] Endif
Endif	If condition Then statements [Else statements] Endif
and	condition 1 and condition 2
or	condition 1 or condition 2
not	not (condition 1)
(	open parenthesis
)	close parenthesis
IS	data element IS expression; (assignment statement)
:=	data element := expression; (assignment statement)
=	expression 1 = expression 2
EQ	expression 1 EQ expression 2
<	expression 1 < expression 2
LT	expression 1 LT expression 2
<=	expression 1 <= expression 2
LE	expression 1 LE expression 2
>	expression 1 > expression 2
GT	expression 1 GT expression 2
>=	expression 1 >= expression 2
GE	expression 1 GE expression 2
<>	expression 1 <> expression 2
NE	expression 1 NE expression 2
;	end of statement
+	expression 1 + expression 2
Plus	expression 1 Plus expression 2
-	expression 1 - expression 2
Minus	expression 1 Minus expression 2
*****	expression 1 * expression 2
Mul	expression 1 Mul expression 2
/	expression 1 / expression 2
Div	expression 1 Div expression 2
,	function(expression 1, expression 2, ...)