
HighQSoft



Manual

Version 1.2

Ascoba

Karst Schaap 2007/04/07

Contents

1	Ascoba Manual	1
1.1	Ascoba Manual	1
2	Modification Histroy	11

Chapter 1

Ascoba Manual

1.1 Ascoba Manual

1.1.1 General Information

This manual should give the reader a short overview of the Ascoba software and its features.

- **The Purpose of Ascoba**(p. 1), The Purpose of Ascoba, explains what Ascoba is and what it does.
- **The ini-files**(p. 2), The ini-files, should give a survey of the use of ini-files.
- **The Ascoba Syntax**(p. 3), The Ascoba Syntax, describes the syntax of the various functions.
- **Ascoba Examples**(p. 8), Examples, shows some examples how Ascoba works.
- **Known Problems and Restrictions**(p. 9), Known Problems and Restrictions.

For the purpose of clarity of the text, filenames are shown in *italic letters*; commands and programme examples are shown in the following font

`Courier`

1.1.2 The Purpose of Ascoba

With the development of the ASAM standard many people have in the meantime experienced the problem - when working with ASAM conform data - to transport the information from one database to any other, eventually on a different computer or even in a different company.

Generally speaking Ascoba is an import and export tool for ASAM conform data. The programme offers the possibility to export, import, delete, or list any ASAM conform data or parts thereof.

Dynamic driver linking at runtime is supported.

1.1.3 The Start of Ascoba

Ascoba has a variety of minimum system requirements depending on which computer it will be installed. The programme is available at the moment for Windows NT, HP-UX, VAX (under VMS

6.2 and UCX 4.1).

The commands given below are only valid for the VAX.

- ascoba.exe	is the executable file of Ascoba
- setupascoba.com	the last two lines define the logical name and symbol and is the absolute minimum requirement: <pre>\$ ascoba := \$'dsknam'['dirnam']ascoba.exe \$ define ASCOBA_INI 'dsknam'['dirnam']ascoba_demo.ini</pre>
- ascoba_demo.ini	holds the initial statements and definitions for Ascoba

When you are ready to start Ascoba simply type

```
@setupascoba
```

.

1.1.4 The ini-files

For a proper and efficient use of Ascoba various adjustments have to be determined. This prevents that certain recurrent definitions such as e.g. RPC numbers, services etc. have to be specified again and again.

These adjustments are listed in an ini-file. The corresponding current ini-file has to be specified somewhere for Ascoba. This is made by defining an environment variable - here ascoba_ini (see picture below). On VAX machines the variable is defined with the file setupascoba.com (see chapter 3, (The Start of Ascoba)).

For the demo version this ini-file is called ascoba_demo.ini. The following shows an example of such an ini-file:

```
[ATHOS]                                ; Athos system definitions.
BASE_MODEL_URL    = ao_base.htm
MAX_RUN_INST      = 10
[SERVICE "ASAM ODS Protocol Level Version 3"]
RPCNUMBER = 553652250
NODENAME   = Atlas
ENVNAME    = Your Choice
[SERVICE "ASAM Transport Format (ATF)"]
FILENAME   = ascoba.atf
```

The line "BASE_MODEL_URL ..." defines the path to the file "ao_base.htm" which contains the Athos system definitions.

When defining the service name please note that the names have to be unique. The attributes of the last service in the ini-file will overwrite the attributes of the previous entry.

The node name and RPC number has to be changed according to your environment. The environment name "ENVNAME" is optional and will be overwritten when the environment is successfully opened.

A special Ascoba definition in the ini-file is

```
MAX_RUN_INST=10;
```



This value gives the maximum number of selected instances for each run and is checked before the references are loaded. Please note: It has only influence when the command is started with the "with reference" - clause. Depending on the structure of the server data it can take a long time to load the references:

ascoba select "Versuch" where {"*Id" LT "100"} with reference using "wk.atf" This would cause that for the 100 instances the references are loaded. With the given variable MAX_RUN_INST the command is split into 10 different commands. The performance difference between the total command and the split command can be enormous. If the command is split the ATF-filename is automatically generated for a unique filename, the new filename will get a continuous number append to the filename.

```
e.g. wk.atf ->wk.atf.001
wk.atf.002
wk.atf.003 etc.
```

1.1.5 The Ascoba Syntax

The functions used in Ascoba have to follow a certain syntax. This chapter should give the user an overview which syntax has to be adhered to.

These paragraphs describe the currently valid commands and their syntax.

```
ascoba VERBOSE FUNCTION "Application element" WHERE {Condition} WITH REFERENCE
FROM "SERVICE1" TO "SERVICE2" USING [Filename | Modifier]
AT "Instance element" IGNORE
```

The verbose keyword is an optional keyword and must not necessarily be included in each command. In case it is included the processing information is shown when executing the command.

FUNCTIONS

select selects data from first service of ini-file to second service

insert inserts data from second service of ini-file to first service

modify modify an attribute of an instance element of first service

delete delete data information from first service

list list data information from first service

help shows help

Select Selects the elements from the first service of the ini-file and save the elements in the second service. The first and second service are the order of the services in the INI-file and can be overwritten by the FROM and TO clause. The FROM service is the source service the TO service is the tagret service (ATF). The application and instance elements are saved. If there is only one service specified in the ini-file the Select command cannot be used.

The name of the atf-file from the second service will be overwritten with the filename specified in the USING clause.

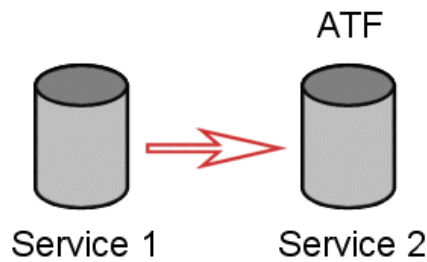


Figure 1.1: Select

There are some INI-file variables which can be defined at the second service to modify the number of selected application elements and instances. An INI-file variable (ASCOBA_-COMPLETE_MODEL) forces ascoba to append all the application elements from the first service to the list with selected objects, which will be stored in the second service. Another INI-file variable (ASCOBA_COPY_ALL_INSTANCES_OF) forces ascoba to append the application element and all the instances of the application element to the list with selected objects. This last INI-file variable overwrites the select of the instances in the select clause. When a select is given for the application element this last INI-file variable select all instances of that element.

Insert Inserts the selected instance elements from the second service into the first service at the specified application element. The first and second service are the two services of the INI-file and can be overwritten by the FROM and TO clause. The FROM service is the source service (ATF) the TO service is the target service. The name of the atf-file from the second service will be overwritten with the filename specified in the using clause. If there is only one service specified in the ini-file the Insert command cannot be used. The Id's of the instances will be given new at the TO service. The references will be updated to the Id's of the TO service. The referenced instances will be search by the ASAM Path. ASCOBA first checks if the instance (ASAM Path) exist already in the TO service before the instance will be inserted.

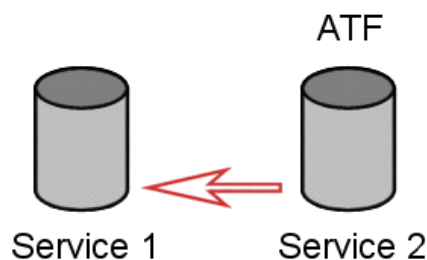


Figure 1.2: Insert

Modify Modifies an attribute of the selected instance element of the first service. The first service is the first service in the INI-File and can be overwritten with the FROM clause. The name of the attribute and the new value are specified in the using clause.

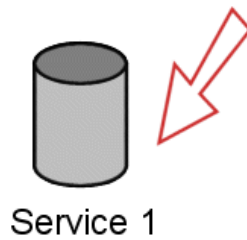


Figure 1.3: Modify

Delete Deletes the selected instance elements from the first service. The name of the atf-file from the first service will be overwritten with the filename specified in the USING clause. The first service is the first service in the INI-file and can be overwritten by the FROM clause.

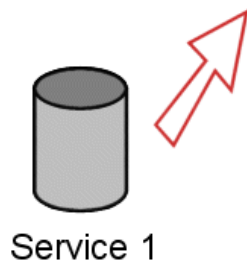


Figure 1.4: Delete

List Lists the names of the selected application elements and the selected instance element to the standard output from the first service. The first service is the first service in the INI-file and can be overwritten by the FROM clause. In case there is no "WHERE"-clause specified only the application elements are listed.

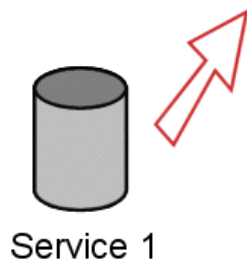


Figure 1.5: List

Help Shows a short help information on Ascoba.

1.1.6 Application Element

"Application element" The name of an application element is needed. Wildcard characters may be used to specify the name (see description WILDCARD).

{CONDITION} Ascoba can use a variety of conditions and condition operators which are listed below. Conditions always have to be parenthized with curved brackets. A condition with a condition operator would have to look like:

```
{"Attribute name" CONDITION OPERATOR "Attribute value"}
```

A condition with condition and logical operator would have to look like

```
{"Attribute name" CONDITION OPERATOR "Attribute value" LOGICAL OPERATOR
  "Attribute name" CONDITION OPERATOR "Attribute value"}
CONDITION OPERATOR = EQ equal
                    NE not equal
                    LT less than
                    GT greater than
                    LE less or equal
                    GE greater or equal
LOGICAL OPERATION = AND
                  OR
                  NOT
                  XOR exclusive or
```

"Attribute name" Name of the application attribute. The name can be specified with wildcard character (see description WILDCARD). If there is no application attribute with the specified name, ASCOBA try to find an application attribute with a base attribute which name corresponds the given attribute name. The search for the base attribute name is case blind, but no wildcards are allowed.

"Attribute value" Value of the attribute. The value can be specified with wildcard character (see description WILDCARD).

WILDCARD Ascoba offers the possibility to use wildcard characters with application elements and attribute values. When using wildcard characters the name or value has to be put in doublequotes. It is recommended to always use the doublequotes - although only necessary with wildcard characters - thus bypassing a typical syntax error. There are two different kinds of 'wildcarding'.

```
WILDCARD = * to substitute several characters
WILDCARD_ONE = ? to substitute just one character
```

Filename The name of the atf-file. Using the functions select or insert the filename at the second service listed in the ini-file is changed. With the other functions the name of the first service is changed. The services are defined in the ini-file. With the USING clause it is allowed to use wildcard characters, e.g. "wk.atf.*" (see description WILDCARD). When using wildcards all matching filenames will be handled in separate commands of Ascoba. These commands will be generated automatically.

Modifier The name of an attribute and its new value (attribute="value"). Using the function modify the attribute of the selected instance is set to the new value.

WITH REFERENCE using the select or list function this means that the elements referenced by and referring to the application or instance elements are also included.

When using the function insert it means all the instance elements referring to the selected instance element are also inserted. When using the function delete it means that it is the server's obligation to delete the references.

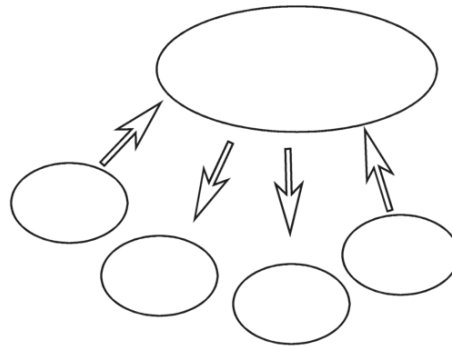


Figure 1.6: With reference

1.1.7 Instance Element

Instance element

The name of an instance element is needed. Wildcard characters are not allowed.

AT

means the father instance element of the first service specified by an ASAM-path at which the inserted instance is connected.

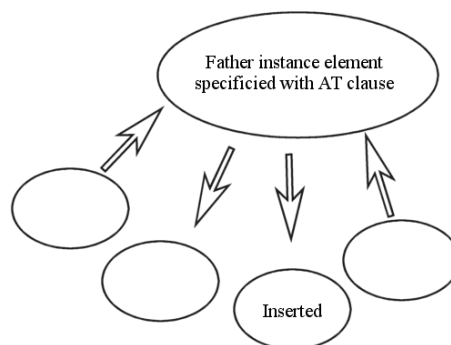


Figure 1.7: At clause

1.1.8 Mandatory clauses

FROM

This clause is mandatory. The defined service name is the source service. In case no service is given see **The Ascoba Syntax**(p. 3), Functions, which explains which services are opened.

TO

This clause is mandatory. The defined service name is the target service. In case no service is given see **The Ascoba Syntax**(p. 3), Functions, which explains which services are opened.

IGNORE

This clause is mandatory and only use with the function 'insert' (see **The Ascoba Syntax**(p. 3)). ASCOBA will stop the operation if an instance element exist already in the target service. If this clause is set ASCOBA will continue the operation with the next instance and report only a messages to the user in VERBOSE mode (**The Ascoba Syntax**(p. 3)).

1.1.9 Ascoba Examples

The following examples show various Ascoba commands with their correct syntax and a description of what they should do. As previously pointed out in this manual these examples use for the adjustments the file athos.ini.

1.1.10 Creating an atf-file using a condition operator

Create an atf-file with the name ascoba.atf. The instance from the application element Messungen with an attribute named *Id and a value equal 100 will be written to the atf-file. The references from or to the selected application and instance element are also stored in the atf-file. ascoba select "Messungen" where {"*Id" EQ "100"} with reference using ascoba.atf

1.1.11 Creating an atf-file using a condition and a logical operator

Create an atf-file with the name ascoba.atf. The instance from the application element Messungen with an attribute named *Id and a value greater than 100 and less than 110 will be written to the atf-file. ascoba select "Messungen" where {"*Id" GT "100" and "*Id" LT "110"} using ascoba.atf

1.1.12 List application elements

List all application elements with the attributes from the service. ascoba list ""

1.1.13 List instances which fulfill a certain condition

List all instances from the application element Messungen with an attribute named *Id and a value greater than 100. ascoba list "Messungen" where "*Id" GT "100"

1.1.14 List instances which fulfill a certain condition using wildcard characters

List all instances from the application element Messungen with an attribute named *Name and a value equal E34*. ascoba list "Messungen" where "*Name" EQ "E34*"

1.1.15 Delete instances which fulfill a certain condition

Delete all instances from the application element Messungen with an attribute named *Id and a value greater than 100. The references to the instances are also deleted, but not the referencing instances. ascoba delete "Messungen" where "*Id" GT "100"

1.1.16 Delete instances without a reference

Delete all instances from the application element Messungsgrößen without a reference to a Messung. ascoba delete "Messungsgr*" where "Messung" EQ 0

1.1.17 Copy instances from one service to another service

Copy all instances of "Versuch" with "ID" greater than 100 from service "Rad" to "ATF-Target". In case there are more instances of "Versuch" than in the variable MAX_RUN_INST defined the atf-files ascoba.atf.001, ascoba.atf.002 etc. will be created automatically. ascoba select "Versuch" where "ID" GT 100 with reference from "Rad" to "ATF-Target" using ascoba.atf

1.1.18 Modify an instance attribute

Set the attribute "Status" of the Instance of Test with Id = 100 to "finished". ascoba modify "Test" where Id EQ 100 from "Rad" using Status="finished"

1.1.19 Insert all instances from an ATF-File in a service

Insert all the instances given in an ATF-File into a service. The service of the ATF-File is called 'ATF-Source', the target service is called 'Motor'. The ATF-File is called 'Motor1234.ATF'

```
ascoba insert "*" where {"*" EQ "*"} from "ATF-Source" to Motor using Motor1234.ATF
```

1.1.20 Known Problems and Restrictions

- The AVL protocol level server version 2 does not support the datatype 'Date'. Therefore, date and time based selections are not supported for this server type.
- The AVL protocol level server version 3 reports datatype 'String' instead of 'Date'. Therefore, date and time based selections are not supported for this server type.
- Ascoba uses only one term of Select-conditions for server side selects. Therefore, e.g. range checking is not yet done completely in the server.



Chapter 2

Modification Histroy

2001/08/05 Karst Schaap

Create the document on HighQSoft format. Add modification history. Attribute name can be the name of the base attribute.

2001/09/05 Markus Schuldlos

Documentation of "modify" - function

2001/10/16 Karst Schaap

Add example for insert function. More detailed description of the used services.

2002/05/21 Karst Schaap

Add the clause IGNORE. Extend the explanation of the insert function.

2005/07/13 Karst Schaap

Cleanup text for new generation.

2007/04/07 Karst Schaap

Add INI-file variable for the select.

Index

Ascoba

- examples, [8](#)
- general information, [1](#)
- INI-files, [2](#)
- problems, [9](#)
- purpose of Ascoba, [1](#)
- restrictions, [9](#)
- start of Ascoba, [1](#)
- syntax, *see* Ascoba Manual, syntax
- Usage, [1](#)

Ascoba Manual, [1](#)

- examples, [8](#)
- general information, [1](#)
- INI-files, [2](#)
- problems, [9](#)
- purpose of Ascoba, [1](#)
- restrictions, [9](#)
- start of Ascoba, [1](#)
- syntax, [3](#)
 - application element, [6](#)
 - instance element, [7](#)
 - mandatory clause, [8](#)

FROM, [8](#)

IGNORE, [8](#)

TO, [8](#)

Usage

Ascoba

- examples, [8](#)
- general information, [1](#)
- INI-files, [2](#)
- problems, [9](#)
- purpose of Ascoba, [1](#)
- restrictions, [9](#)
- start of Ascoba, [1](#)
- syntax, *see* Ascoba Manual, syntax