

# Repoz apc commands

Version : 1.0

Date : 19/08/2006

Author : Patrick Germain Placidoux

Copyright (c) 2007-2008, Patrick Germain Placidoux

All rights reserved.

## SUMMARY

1	<a href="#">1 Objective</a>	3
2	<a href="#">2 Introduction</a>	4
3	<a href="#">3 Getting Help</a>	5
4	<a href="#">4 The apc processor</a>	6
5	<a href="#">5 The apath mode</a>	7
5.1	<a href="#">5.1 The List command</a>	8
6	<a href="#">6 The aql mode</a>	9
6.1	<a href="#">6.1 The select operation</a>	10
6.2	<a href="#">6.2 The Update operation</a>	12
6.3	<a href="#">6.3 The Delete operation</a>	13

## **1 OBJECTIVE**

---

This document describes the xpc processor commands

## 2 INTRODUCTION

---

The set of available commands for a processor depends on the processor and the current mode.

The samples below use the files into the directory: <repoz\_installation\_dir>/samples

Note about the sample directory:

To run the Repoz samples of this documentation from a Kikonf installation , replace the directory : <repoz\_intallation\_dir>/samples by: **<kikonf\_intsallation\_dir>/samples/repoz.**

### 3 GETTING HELP

---

The following help commands may be type in any mode.

**help** shows this help: a long help for the commands available for the mounted processor for the current mode.

**h (or help) <command>** shows help for this commands available for the mounted processor for the current mode.

**h** show a short summary of the commands available for the mounted processor for the current mode.

**H** shows a short summary of the globally available commands (for all mode, all processor).

**HELP** shows a long help for the globally available commands.

**H (or HELP) <command>** shows help for this globally available commands.

## 4 THE APC PROCESSOR

---

The xpc processor is used to work with properties files (aka attributes) files.  
An apc processor is mounted using the apc command like this :

```
:-> xpc -F <repoz_intallation_dir>/samples/test.xml
```

**e.g.:**

```
:->apc -F <repoz_intallation_dir>samples/test.attrs -D <repoz_intallation_dir>samples/test.attrs.desc  
-A -C
```

*New processor with alias:test, created and mounted.*

## 5 THE APATH MODE

---

The apath implementation supports one command: ls

**To switch to the apath mode :**

- Switch to the apath mode  
e.g. (from the python : ? mode):  
**?test:/>:** (or type "mode xpath")  
:test:/>
- From any mode Create and Mount an xpc processors  
e.g.:  
:test:/>**apc -F** <repoz\_intallation\_dir>samples/**test.attrs -D**  
<repoz\_intallation\_dir>samples/**test.attrs.desc -A -C -a mypc**  
*New processor with alias:mypc, created. (use mount mypc, to mount it)*  
:test:/>
- Mount your new processor (if not already mounted)  
e.g.:  
**:test:/:mount mypc**  
:mypc:/>

## 5.1 THE LIST COMMAND

---

The ls command list one or more Attributes.

Syntax:

```
ls <attr> [<attr>]
```

e.g.:

```
:>apc -F <repoz_installation_dir>samples/test.attrs -D  
<repoz_installation_dir>samples/test.attrs.desc -A -C  
New processor with alias:test, created and mounted.
```

```
:test:/>ls field1 field3
```

```
field1:value1 field3:{ccc:caa,ddd:daa}  
:test:/>
```

The ls command returned values are stored into the Repoz reserved Variable: ro.

```
:test:/>var ro
```

```
{'field3': {'ccc': 'caa', 'ddd': 'daa'}, 'field1': 'value1'}  
:test:/>
```

## 6 THE AQL MODE

---

The aql implementation supports 3 commands: **select**, **update**, **delete**

**To switch to the aql mode :**

- Switch to the aql mode  
e.g. (from the python : ? mode):  
**?test:/>%** (or type "mode ql")  
 %test:/>
- From any mode Create and Mount an apc processors  
e.g.:  
**%test:/>apc -F <repoz\_intallation\_dir>samples/test.attrs -D**  
**<repoz\_intallation\_dir>samples/test.attrs.desc -A -C -a mypc**  
*New processor with alias:mypc, created. (use mount mypc, to mount it)*  
 %test:/>
- Mount your new processor (if not already mounted)  
e.g.:  
**%test:/:mount mypc**  
 %mypc:/>

## 6.1 THE SELECT OPERATION

The select operation, selects one or more Attributes according an optional where clause.

### Syntax:

select O\_WHAT at F\_TAGS if F\_ATTRS

- Simple select

### e.g.:

```
>apc -F <repoz_installation_dir>samples/test.attrs -D
```

```
<repoz_installation_dir>samples/test.attrs.desc -A -C
```

*New processor with alias:test, created and mounted.*

```
:test:/>ls
```

```
field1:value1 field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:  
{ccc:caa,ddd:daa}
```

```
:test:/>%
```

```
%test:/>select field1,field3
```

```
field1 field3  
value1 {ccc:caa,ddd:daa}
```

- Select with a where clause

### e.g.:

```
%test:/>select field1,field3 where field1=value1
```

```
field1 field3  
value1 {ccc:caa,ddd:daa}
```

```
%test:/>select field1,field3 where field1<>value1
```

```
field1 field3
```

- Select with an imbricated where clause

### e.g.:

```
%test:/>select field1,field3 where field1=value1 and ((field1 *in [value2,value1] or field1=value1)  
and field1=value1)
```

```
field1 field3  
value1 {ccc:caa,ddd:daa}
```

```
%test:/>select field1,field3 where field1=value1 and ((field1 *in [value2,value1] or field1=value1)  
and field1<>value1)
```

```
field1 field3
```

Note: There is no limit for parenthesis imbrication.

- Select using complex type and a Repoz Variable

e.g. 1:

```
:=>apc -F <repoz_installation_dir>samples/test.attrs -D
```

```
<repoz_installation_dir>samples/test.attrs.desc -A -C
```

*New processor with alias:test, created and mounted.*

```
:test:/>ls
```

```
field1:value1 field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:  
{ccc:caa,ddd:daa}
```

```
%test:/>%
```

```
%test:/>select * where field3={ccc:caa,ddd:daa}
```

```
field1          field2                                     field3  
value1         [{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}]   {ccc:caa,ddd:daa}
```

e.g. 2:

```
%test:/>var v3={ccc:caa,ddd:daa}
```

```
%test:/>var v3
```

```
{'ccc': 'caa', 'ddd': 'daa'}
```

```
%test:/>select * where field3=$v3
```

Var replacement: v3 to: {ccc:caa,ddd:daa}

Var replacement: new line is:select \* where field3={ccc:caa,ddd:daa}

```
field1          field2                                     field3  
value1         [{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}]   {ccc:caa,ddd:daa}
```

## 6.2 THE UPDATE OPERATION

The Update operation, updates a set of Attributes with a set of pair Attribute/Values.

Syntax:

update set O\_SET

Simple update

e.g.:

```
>apc -F <repoz_installation_dir>samples/test.attrs -D
```

```
<repoz_installation_dir>samples/test.attrs.desc -A -C
```

*New processor with alias:test, created and mounted.*

```
:test:/>ls
```

```
field1:value1 field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:
{ccc:caa,ddd:daa}
```

```
:test:/>%
```

```
%test:/>update set field1=aaa
```

*Updating attr:field1*

```
%test:/>:
```

```
:test:/>ls
```

```
field1:aaa field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:
{ccc:caa,ddd:daa}
```

```
:test:/>%
```

- Update with by complexe type

e.g.:

```
%test:/>update set field3={ccc:c,ddd:d}
```

*Updating attr:field3*

```
%test:/>:
```

```
:test:/>ls
```

```
field1:aaa field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:{ccc:c,ddd:d}
```

### 6.3 THE DELETE OPERATION

---

The Delete operation, deletes one or more Attribute(s)

Syntax:

delete F\_ATTRS

e.g.:

```
>apc -F <repoz_installation_dir>samples/test.attrs -D  
<repoz_installation_dir>samples/test.attrs.desc -A -C  
New processor with alias:test, created and mounted.
```

```
:test:/>ls
```

```
field1:value1 field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}] field3:  
{ccc:caa,ddd:daa}
```

```
:test:/>%
```

```
%test:/>delete field1, field3
```

```
%test:/>:
```

```
:test:/>ls
```

```
field2:[{AAA:ccccA2,BBB:ccccB2},{AAA:bbbbA1,BBB:bbbbB1}]
```