

Scripts for testing the record package (assuming currently in **Common Lisp User** package)

Record type - Record

type:

```
(il:record record-test-name (alpha bravo gamma) (il:synonym alpha a) (il:type? (oddp (length il:datum))))  
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some string"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with three numbered elements 1 (a b c), 2 "some string", 3 nil

```
((a b c) some string  
1 (a b c)  
2 "some string"  
3 nil
```

type:

```
(inspect record-test-record)
```

choose "as a record"

choose "record-test-name"

--should produce a window with four elements where a and alpha have the same value.

```
((a b c) some string nil)  
a (a b c)  
gamma nil  
bravo "some string"  
alpha (a b c)
```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window identical to the one in the previous step.

type:

```
(ed 'record-test-name 'records)
```

delete gamma from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements where a and alpha have the same value and there is no gamma.

return to the edit window and type gamma back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value and gamma is returned to the list

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
((a b c) some string nil) Inspector
user::a      (user::a user::b user::c)
user::gamma  nil
user::bravo  "some string"
user::alpha  (user::a user::b user::c)
```

type:

```
(cl:in-package 'user)
```

type:

```
(il:replace (record-test-name gamma) il:of record-test-record il:with '(a (b (c (d (e (f
(g)))))))
(inspect record-test-record)
```

select inspect

-should get:

```
((a b c) some string (
1 (a b c)
2 "some string"
3 (a (b #))
```

type:

```
(setq il:inspectprintlevel '(5 . 5))
(inspect record-test-record)
```

select inspect

-should get:

```
((a b c) some string (
1 (a b c)
2 "some string"
3 (a (b (c (d (
```

type:

```
(setq il:inspectprintlevel '(5 . 1))
(inspect record-test-record)
```

select inspect

-should get:

```
((a b c) some string (
1 (a ...)
2 "some string"
3 (a ...)
```

type:

```
(setq il:inspectprintlevel '(2 . 5))
```

Record type - Typerecord

type:

```
(il:typerecord record-test-name (alpha bravo gamma) (il:synonym alpha a) )
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some
```

```
string"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with four numbered elements 1 record-test-name 2 (a b c), 3 "some string", 4 nil

```
(record-test-name (a b c)
 1 record-test-name
 2 (a b c)
 3 "some string"
 4 nil)
```

type:

```
(setq il:maxinspectcdrlevel 2)
(inspect record-test-record)
```

choose "inspect"

--should produce a window with two numbered elements similar to the previous step and the elements number 3 and 4 in a list labeled il:&&

type:

```
(setq il:maxinspectcdrlevel 50)
```

type:

```
(inspect record-test-record)
```

choose "as a record"

choose "record-test-name"

--should produce a window with four elements where a and alpha have the same value.

```
(record-test-name (a b c)
 a (a b c)
 gamma nil
 bravo "some string"
 alpha (a b c))
```

type:

```
(inspect record-test-record)
```

choose "as record-test-name"

--should produce a window identical to the one in the previous step.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window identical to the one in the previous step.

type:

```
(ed 'record-test-name 'records)
```

delete gamma from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements where a and alpha have the same value and there is no gamma.

return to the edit window and type gamma back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value and gamma is returned to the list

close the edit window**type:**

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
(record-test-name (a b c) some string nil) Ins
user::a      (user::a user::b user::c)
user::gamma  nil
user::bravo  "some string"
user::alpha  (user::a user::b user::c)
```

type:

```
(cl:in-package 'user)
```

Record type - Proprecord

type:

```
(il:proprecord record-test-name (alpha bravo gamma) (il:synonym alpha a) (il:type? (evenp (length
il:datum))))
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some
string"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with four numbered elements 1 alpha, 2 (a b c), 3 bravo, 4 "some string".

```
(alpha (a b c) bravo some string) Ins
1  alpha
2  (a b c)
3  bravo
4  "some string"
```

type:

```
(inspect record-test-record)
```

choose "as a PLIST"

--should produce a window with two elements, alpha and bravo with their associated values.

```
(alpha (a b c) bravo some string) Ins
alpha (a b c)
bravo "some string"
```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value.

```
(alpha (a b c) bravo some string)
a      (a b c)
alpha  (a b c)
bravo  "some string"
gamma  nil
```

type:

```
(ed 'record-test-name 'records)
```

delete gamma from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements where a and alpha have the same value and there is no gamma.

return to the edit window and type gamma back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value and gamma is returned to the list

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
(alpha (a b c) bravo some string) Inspector
user::a      (user::a user::b user::c)
user::alpha  (user::a user::b user::c)
user::bravo  "some string"
user::gamma  nil
```

type:

```
(cl:in-package 'user)
```

Record type - Datatype

type:

```
(il:datatype record-test-name (alpha bravo gamma) (il:synonym alpha a))
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some string"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with four elements where a and alpha have the same value.

```
# <record-test-name @
a      (a b c)
alpha  (a b c)
bravo  "some string"
gamma  nil
```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window identical to the one in the previous step.

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
# <record-test-name @ 375,74770> Inspect
user::a      (user::a user::b user::c)
user::alpha  (user::a user::b user::c)
user::bravo  "some string"
user::gamma  nil
```

type:

```
(cl:in-package 'user)
```

Record type - Arrayrecord

type:

```
(il:arrayrecord record-test-name (alpha bravo gamma) (il:synonym alpha a) (il:type? (cond
(il:datum t))))
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some
string"))
```

type:

```
(inspect record-test-record)
```

--should produce a window with three numbered elements 1 (a b c), 2 "some string", 3 nil

```
# <arrayp @ 377,11
1  (a b c)
2  "some string"
3  nil
```

type:

```
(setq il:maxinspectarraylevel 2)
(inspect record-test-record)
```

--should produce a window similar to the previous one but with only two elements showing.

type:

```
(setq il:maxinspectarraylevel 300)
```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value.

```
# <arrayp @ 377,11514>
a      (a b c)
gamma  nil
bravo  "some string"
alpha  (a b c)
```

type:

```
(ed 'record-test-name 'records)
```

delete gamma from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements where a and alpha have the same value and there is no gamma.

return to the edit window and type gamma back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value and gamma is returned to the list

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
# <arrayp @ 377,11514> Inspector
user::a      (user::a user::b user::c)
user::gamma  nil
user::bravo  "some string"
user::alpha  (user::a user::b user::c)
```

type:

```
(cl:in-package 'user)
```

Record type - Assocrecord

type:

```
(il:assocrecord record-test-name (alpha bravo gamma) (il:synonym alpha a) (il:type? (not (atom (car il:datum))))))
(setq record-test-record (il:create record-test-name alpha il:_ '(a b c) bravo il:_ "some string"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with two numbered elements 1 (alpha a b c), 2 (bravo . "some string")

```
((alpha a b c) (bravo . some string))
1 (alpha a b c)
2 (bravo . "some string")
```

type:

```
(inspect record-test-record)
```

choose "as an ALIST"

--should produce a window with two elements, alpha and bravo and their associated values.

```
((alpha a b c) (bravo . some string))
alpha (a b c)
bravo "some string"
```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements and a has the same value as alpha.

```
((alpha a b c) (bravo . some string))
a (a b c)
alpha (a b c)
bravo "some string"
gamma nil
```

type:

```
(ed 'record-test-name 'records)
```

delete gamma from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements where a and alpha have the same value and there is no gamma.

return to the edit window and type gamma back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window with four elements where a and alpha have the same value and gamma is returned to the list

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with four elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```
((alpha a b c) (bravo . some string)) Inspecto
user::a (user::a user::b user::c)
user::alpha (user::a user::b user::c)
user::bravo "some string"
user::gamma nil
```



```
type:
(cl:in-package 'user)
```

Record type - Accessfns

```
type:
(IL:ACCESSFNS RECORD-TEST-NAME
  ((ALPHA (CAR IL:DATUM)
    (SETQ IL:DATUM (CONS IL:NEWVALUE
      (CDR IL:DATUM))))
    (BRAVO (CADR IL:DATUM)
      (SETQ IL:DATUM (CONS (CAR IL:DATUM)
        (CONS IL:NEWVALUE
          (CDDR IL:DATUM))))))
    (GAMMA (CADDR IL:DATUM)
      (SETQ IL:DATUM (LIST (CAR IL:DATUM)
        (CADR IL:DATUM)
          IL:NEWVALUE))))
    (IL:CREATE (LIST '(a b c) "some string" NIL))
    (IL:TYPE? (ODDP (LENGTH IL:DATUM))))
  (setq record-test-record (il:create record-test-name))
```

```
type:
(inspect record-test-record)
```

choose "inspect"

--should produce a window with three numbered elements 1 (a b c), 2 "some string". 3 ,nil

```
((a b c) some string nil)
1 (a b c)
2 "some string"
3 nil
```

```
type:
(inspect record-test-record 'record-test-name)
```

--should produce a window with three elements: alpha, bravo and gamma with their associated values.

```
((a b c) some string nil)
alpha (a b c)
bravo "some string"
gamma nil
```

```
type:
(ed 'record-test-name 'records)
```

delete the gamma entry from the element list.

type cntl-x to save edit.

```
type:
(inspect record-test-record 'record-test-name)
--should produce a window with two elements where there is no gamma.
```

return to the edit window and type the gamma entry back in where it belongs.

type cntl-x to save again.

```
type:
(inspect record-test-record 'record-test-name)
--should produce a window with three elements where gamma is returned to the list
```

close the edit window

```
type:
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name)
```

--should produce a window with three elements where user::a and user::alpha have the same value. All the values in the list (for a and alpha) are qualified as user.

```

((a b c) some string nil) Inspector
user::alpha (user::a user::b user::c)
user::bravo "some string"
user::gamma nil

```

type:

```
(cl:in-package 'user)
```

Record type - Blockrecord

type:

```

(IL:DATATYPE RECORD-TEST-NAME
  ((ALPHA IL:POINTER) alpha il:_ '(a b c))
(IL:BLOCKRECORD RECORD-TEST-NAME1
  ((BRAVO IL:WORD) (GAMMA IL:WORD)))
(SETQ RECORD-TEST-RECORD (IL:CREATE RECORD-TEST-NAME))

```

type:

```
(inspect record-test-record)
```

--should produce a window with alpha and the list (a b c)

```

#<record-test-r
alpha (a b c)

```

type:

```
(inspect record-test-record 'record-test-name)
```

--should produce a window identical to the one in the previous step.

type:

```
(inspect record-test-record 'record-test-name1)
```

--should produce a window with bravo and gamma, each with two numbers

```

#<record-tes
bravo 254
gamma 39654

```

type:

```
(ed 'record-test-name1 'records)
```

delete the gamma entry from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name1)
```

--should produce a window with one element where there is no gamma.

return to the edit window and type the gamma entry back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name1)
```

--should produce a window with two elements where gamma is returned to the list

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name1)
```

--should produce a window with two elements where .All the values in the list are qualified as user.

```
# <record-test-name
user::bravo 254
user::gamma 39654
```

type:

```
(cl:in-package 'user)
```

Record type - Subrecords

type:

```
(il:record record-test-name (alpha bravo gamma) (il:synonym alpha a) (il:type? (oddp (length
il:datum))))
(il:record record-test-name1 (xray zebra record-test-name)
(il:subrecord record-test-name))
(setq record-test-record (il:create record-test-name1 alpha il:_ '(a b c) bravo il:_ "some
string" zebra il:_ "hi"))
```

type:

```
(inspect record-test-record)
```

choose "inspect"

--should produce a window with three numbered elements 1 nil 2 "hi" and 3 the list ((a b c) "some string" nil)

```
(nil hi ((a b c) some string nil)) Inspect
1 nil
2 "hi"
3 ((a b c) "some string" nil)
```

type:

```
(inspect record-test-record)
```

choose "as a record"

choose "record-test-name1"

--should produce a window with three elements the first one of which is a list of a list, a string and nil .

```
(nil hi ((a b c) some string nil)) Inspector
record-test-name ((a b c) "some string" nil)
zebra           "hi"
xray            nil
```

type:

```
(inspect record-test-record 'record-test-name1)
```

--should produce a window identical to the one in the previous step.

type:

```
(ed 'record-test-name1 'records)
```

delete zebra from the element list.

type cntl-x to save edit.

type:

```
(inspect record-test-record 'record-test-name1)
--should produce a window with no zebra and the value that used to be zebra is now record-test-name.
```

return to the edit window and type zebra back in where it belongs.

type cntl-x to save again.

type:

```
(inspect record-test-record 'record-test-name1)
--should produce a window with three elements the first one of which is a list of a list, a string and nil and
zebra is returned to the list
```

close the edit window

type:

```
(in-package 'interlisp)
```

type:

```
(inspect user::record-test-record 'user::record-test-name1)
--should produce a window with three elements where all the values not in quotes are qualified as user.
```

```
(nil hi ((a b c) some string nil)) Inspector
user::record-test-name ((user::a user::b user::c)
user::zebra           "hi"
user::xray             nil)
```

type:

```
(cl:in-package 'user)
```
