INDEX

A
assignment statements 40
audit trail of rule invocation 23
auditing RuleSets 52
auditObject *(Variable)* 34
auditVarName *(Variable)* 34

B
breaking and tracing RuleSets 50

C
caller *(Variable)* 33
colon-comma in a literal 38
comments 32
comparison with other rule languages 26
compiler options for breaking and tracing 50
computing selectors 39
control structures for selecting rules 18
converting from Buttress Rules 55
copying RuleSets 46
CopyRules *(Message)* 46
creating RuleSets 45

D
DefAVP *(Function)* 5.5
DefRSM *(Function)* 48
Do1 *(RuleSet Control Structure)* 18
DoAll *(RuleSet Control Structure)* 19
dollar notation to invoke RuleSets 42
double colon in a literal 38
double-dot syntax to invoke RuleSets 42
double-dot-star syntax to invoke RuleSets 43

E
EditAllDecls 46
editing RuleSets 45
EditMethod *(Message)* 45
EditRules *(Message)* 45
ER *(Message)* 45
exclamation sign to compute names 39
ExplicitFnActiveValue 4

F
factoring meta-level syntax 26
first/last rules 23
FOR1 *(RuleSet Control Structure)* 21
FORALL *(RuleSet Control Structure)* 21

I
if-then rules 15
infix operators 35
installing RuleSets
    as methods 48
    in active values 49
integrated programming environment 29
Interlisp
    constants 34
    functions 37
invoking RuleSets 42
items in release 1
iteration-condition in RuleSets 21

L
LHS syntax 31
ListRuleSets *(Function)* 47
literal 38
loading rules 52

M
message sending 37
meta-assignment statements 41
meta-control of RuleSets 42
multiple colons in a literal 40

O
one-shot rules 22

P
pop statement 42
PPR *(Message)* 47
PPRules *(Message)* 47
printing RuleSets 47
production rules 15
properties 38
push statement 42

Q
quoted constants 34

R
RE 51
reasons *(Variable)* 33
recursive compound literals 40
RHS syntax 32
rs *(Variable)* 33
RSGet *(Property)* 49
RSGetFn *(Function)* 49
RSPut *(Property)* 49
RSPutFn *(Function)* 49
rule Exec 51
rule-oriented programming 15
ruleApplied *(Variable)* 33
ruleLabel *(Variable)* 33
ruleNumber *(Variable)* 33
ruleObject *(Variable)* 33
rules 15
    basic concepts 16
    forms 34
    language 31
    loading 52
    major features 15
    using 45
    work space 15
RuleSets 15
    approaches to organizing 17
    auditing 52
    breaking and tracing 50
    control structures 28
    copying 46
    creating 45
    editing 45
    hierarchy 27
    installing as methods 48
    installing in active values 49
    invoking 15,42
    iteration condition 21
    meta-control 42
    printing 47
    protocols 47
    running from LOOPS 47

saving on Lisp files 47
running RuleSets from LOOPS 47
RunRS *(Function)* 48

S
saving RuleSets on Lisp files 47
self *(Variable)* 33
single colon in a literal 38
Stop *(RuleSet Statement)* 43
strings 34
system configuration 1

T
transfer calls 43

U
unary message sending 37
using rules 45

V
variable names 39
variables 32,38

W
While1 *(RuleSet Control Structure)* 19
WhileAll *(RuleSet Control Structure)* 20
work space for rules 15

←
←+ push statement 42
←- pop statement 42

~
~ *(Rule Unary Operator)* 36
~= *(Rule Infix Operator)* 36

!
! to compute names 39

$
$ to invoke RuleSets 42

***
***** *(Rule Infix Operator)* 35

+
+ *(Rule Infix Operator)* 35
++ *(Rule Infix Operator)* 35

-
- *(Rule Infix Operator)* 35
- *(Rule Unary Operator)* 36
-- *(Rule Infix Operator)* 35

.
.. to invoke RuleSets 42
..* to invoke RuleSets 43

/
/ *(Rule Infix Operator)* 35

:
: in a literal 38
:, in a literal 38
:: in a literal 38

<
< *(Rule Infix Operator)* 35
<< *(Rule Infix Operator)* 36
<= *(Rule Infix Operator)* 35

=
= *(Rule Infix Operator)* 36
== *(Rule Infix Operator)* 36

>
> *(Rule Infix Operator)* 35
>= *(Rule Infix Operator)* 35

[This page intentionally left blank]