Verification Continuum™ Verdi®
Python-Based NPI Coverage Model

Version V-2023.12-SP1, March 2024

SYNOPSYS®

Copyright and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader’s responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Free and Open-Source Licensing Notices

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

Contents

Customer Support 27

Synopsys Statement on Inclusivity and Diversity 28

1. **Introduction to Python Based NPI** 29

Package and Modules 29

Packages 29

Modules 29

Module Functions and Class Objects 30

L0 Module Functions 30

L1 Module Functions 30

Class Objects 30

User Interface and Use Flow 30

Environment and Library Setting 30

2. **Module npisys** 33

Overview 33

L0 APIs 33

npisys.init(*pyArgvList*) 33

npisys.load_design(*pyArgvList*) 33

npisys.end() 34

3. **Python-Based NPI Coverage Model** 35

Overview 35

Quick Start 35

Class Objects 37

4. **Class Objects** 39

Database 41

class cov.Database 41

test_handles() 41

instance_handles() 41

close() . . . 41
handle_by_name(***name***) . . . 41
test_by_name(***name***) . . . 42
type(***test=None, is_get_enum=False***) . . . 42
name() . . . 42
class cov.Test . . . 43
assert_metric_handle() . . . 43
testbench_metric_handle() . . . 43
test_info_handles() . . . 44
program_handles() . . . 44
power_data_handles() . . . 44
unload_test() . . . 44
save_test(***name***) . . . 45
load_exclude_file(***name***) . . . 45
save_exclude_file(***name, mode***) . . . 45
unload_exclusion() . . . 46
type(***test=None, is_get_enum=False***) . . . 46
name() . . . 46
class cov.TestInfo . . . 47
name() . . . 47

Test . . . 47

class cov.Test . . . 47

database_handle() . . . 47
testbench_metric_handle() . . . 48
power_data_handles() . . . 48
unload_test() . . . 48
save_test(***name***) . . . 48
load_exclude_file(***name***) . . . 49
save_exclude_file(***name, mode***) . . . 49
unload_exclusion() . . . 50
type(***test=None, is_get_enum=False***) . . . 50
line_no(***test=None***) . . . 50
size(***test=None***) . . . 51
coverable(***test=None***) . . . 51
covered(***test=None***) . . . 51
set_covered(***test, value***) . . . 51
count(***test=None***) . . . 52
count_goal(***test=None***) . . . 52
status(***test=None***) . . . 52
has_status_unreachable(***test***) . . . 53
set_status_unreachable(***test, value***) . . . 53
has_status_illegal(test) . . . 53
set_status_illegal(***test, value***) . . . 53
has_status_excluded_at_compile_time(***test***) . . . 54

set_status_excluded_at_compile_time(***test, value***) 54
has_status_covered(***test***) 54
set_status_covered(***test, value***) 54
has_status_proven(***test***) 55
set_status_proven(***test, value***) 55
has_status_excluded_at_report_time(***test***) 55
set_status_excluded_at_report_time(***test, value***) 56
has_status_attempted(***test***) 56
set_status_attempted(***test, value***) 56
has_status_excluded(***test***) 56
set_status_excluded(***test, value***) 57
has_status_partially_excluded(***test***) 57
set_status_partially_excluded(***test, value***) 57
has_status_partially_attempted(***test***) 58
set_status_partially_attempted(***test, value***) 58
set_status(***test, value***) 58
per_instance(***test=None***) 58
is_mda(***test=None***) 59
is_port(***test=None***) 59
is_event_condition(***test=None***) 59
severity(***test=None***) 60
category(***test=None***) 60
name() 60

Instance 60

class cov.Instance 60

database_handle() 60
instance_handles() 61
line_metric_handle() 61
toggle_metric_handle() 61
fsm_metric_handle() 62
condition_metric_handle() 62
branch_metric_handle() 62
assert_metric_handle() 62
scope_handle() 63
type(***test=None, is_get_enum=False***) 63
line_no(***test=None***) 63
status(***test=None***) 64
has_status_unreachable(***test***) 64
set_status_unreachable(***test, value***) 64
has_status_illegal(***test***) 64
set_status_illegal(***test, value***) 65
has_status_excluded_at_compile_time(***test***) 65
set_status_excluded_at_compile_time(***test, value***) 65
has_status_covered(***test***) 65

set_status_covered(***test, value***) 66
has_status_proven(***test***) 66
set_status_proven(***test, value***) 66
has_status_excluded_at_report_time(***test***) 67
set_status_excluded_at_report_time(***test, value***) 67
has_status_attempted(***test***) 67
set_status_attempted(***test, value***) 67
has_status_excluded(***test***) 68
set_status_excluded(***test, value***) 68
has_status_partially_excluded(***test***) 68
set_status_partially_excluded(***test, value***) 69
has_status_partially_attempted(***test***) 69
set_status_partially_attempted(***test, value***) 69
set_status(***test, value***) 69
name() 70
def_name() 70
def_info() 70
full_name() 70
file_name() 71

LineCoverage 71

class cov.LineMetric 71

instance_handle() 71
child_handles() 71
type(*test=None, is_get_enum=False*) 72
name() 72
full_name() 72

Block 73

class cov.Block 73

scope_handle() 73
child_handles() 73
type(***test=None, is_get_enum=False***) 73
line_no(***test=None***) 74
coverable(***test=None***) 74
covered(***test=None***) 74
status(***test=None***) 74
has_status_unreachable(***test***) 75
set_status_unreachable(***test, value***) 75
has_status_illegal(***test***) 75
set_status_illegal(***test, value***) 76
has_status_excluded_at_compile_time(***test***) 76
set_status_excluded_at_compile_time(***test, value***) 76
has_status_covered(***test***) 76
set_status_covered(test, value) 77

has_status_proven(***test***) . . . 77
set_status_proven(***test, value***) . . . 77
has_status_excluded_at_report_time(***test***) . . . 77
set_status_excluded_at_report_time(***test, value***) . . . 78
has_status_attempted(***test***) . . . 78
set_status_attempted(***test, value***) . . . 78
has_status_excluded(test) . . . 79
set_status_excluded(***test, value***) . . . 79
has_status_partially_excluded(***test***) . . . 79
set_status_partially_excluded(***test, value***) . . . 79
has_status_partially_attempted(***test***) . . . 80
set_status_partially_attempted(***test, value***) . . . 80
set_status(***test, value***) . . . 80
name() . . . 81
file_name() . . . 81

StmtBin . . . 81

class cov.StmtBin . . . 81
covered_test_handles(***test***) . . . 81
scope_handle() . . . 82
type(***test=None, is_get_enum=False***) . . . 82
line_no(***test=None***) . . . 82
coverable(***test=None***) . . . 82
covered(***test=None***) . . . 83
set_covered(***test, value***) . . . 83
status(***test=None***) . . . 83
has_status_unreachable(***test***) . . . 84
set_status_unreachable(***test, value***) . . . 84
has_status_illegal(***test***) . . . 84
set_status_illegal(***test, value***) . . . 85
has_status_excluded_at_compile_time(***test***) . . . 85
set_status_excluded_at_compile_time(*test, value*) . . . 85
has_status_covered(***test***) . . . 85
set_status_covered(***test, value***) . . . 86
has_status_proven(***test***) . . . 86
set_status_proven(***test, value***) . . . 86
has_status_excluded_at_report_time(***test***) . . . 86
set_status_excluded_at_report_time(***test, value***) . . . 87
has_status_attempted(***test***) . . . 87
set_status_attempted(***test, value***) . . . 87
has_status_excluded(***test***) . . . 88
set_status_excluded(***test, value***) . . . 88
has_status_partially_excluded(***test***) . . . 88
set_status_partially_excluded(***test, value***) . . . 88
has_status_partially_attempted(***test***) . . . 89

set_status_partially_attempted(***test, value***) 89
set_status(***test, value***) 89
name() 90
file_name() 90

Toggle Coverage 90

class cov.ToggleMetric 90
instance_handle() 90
child_handles() 90
type(***test=None, is_get_enum=False***) 91
name() 91
full_name() 91

Signal 92

class cov.Signal 92
scope_handle() 92
child_handles() 92
type(***test=None, is_get_enum=False***) 92
line_no(***test=None***) 93
size(***test=None***) 93
coverable(***test=None***) 93
covered(***test=None***) 93
status(***test=None***) 94
has_status_unreachable(***test***) 94
set_status_unreachable(***test, value***) 94
has_status_illegal(***test***) 95
set_status_illegal(***test, value***) 95
has_status_excluded_at_compile_time(***test***) 95
set_status_excluded_at_compile_time(***test, value***) 95
has_status_covered(***test***) 96
set_status_covered(***test, value***) 96
has_status_proven(***test***) 96
set_status_proven(***test, value***) 97
has_status_excluded_at_report_time(***test***) 97
set_status_excluded_at_report_time(***test, value***) 97
has_status_attempted(***test***) 97
set_status_attempted(***test, value***) 98
has_status_excluded(***test***) 98
set_status_excluded(***test, value***) 98
has_status_partially_excluded(***test***) 98
set_status_partially_excluded(***test, value***) 99
has_status_partially_attempted(***test***) 99
set_status_partially_attempted(***test, value***) 99
set_status(***test, value***) 100
is_mda(***test=None***) 100

is_port(***test=None***) 100
name() 101
file_name() 101

SignalBit 101

class cov.SignalBit 101

scope_handle() 101
child_handles() 101
type(***test=None, is_get_enum=False***) 102
line_no(***test=None***) 102
size(***test=None***) 102
coverable(***test=None***) 103
covered(***test=None***) 103
status(***test=None***) 103
has_status_unreachable(***test***) 103
set_status_unreachable(***test, value***) 104
has_status_illegal(***test***) 104
set_status_illegal(***test, value***) 104
has_status_excluded_at_compile_time(***test***) 105
set_status_excluded_at_compile_time(***test, value***) 105
has_status_covered(***test***) 105
set_status_covered(***test, value***) 105
has_status_proven(***test***) 106
set_status_proven(***test, value***) 106
has_status_excluded_at_report_time(***test***) 106
set_status_excluded_at_report_time(***test, value***) 106
has_status_attempted(***test***) 107
set_status_attempted(***test, value***) 107
has_status_excluded(***test***) 107
set_status_excluded(***test, value***) 108
has_status_partially_excluded(***test***) 108
set_status_partially_excluded(***test, value***) 108
has_status_partially_attempted(***test***) 108
set_status_partially_attempted(***test, value***) 109
set_status(***test, value***) 109
is_mda(***test=None***) 109
is_port(***test=None***) 110
name() 110
file_name() 110

Toggle Bin 110

class cov.ToggleBin 110

covered_test_handles(***test***) 110
scope_handle() 111
type(***test=None, is_get_enum=False***) 111

toggle_type(***test=None, is_get_enum=False***) 111
coverable(***test=None***) 112
covered(***test=None***) 112
set_covered(***test, value***) 112
status(***test=None***) 113
has_status_unreachable(***test***) 113
set_status_unreachable(***test, value***) 113
has_status_illegal(***test***) 114
set_status_illegal(***test, value***) 114
has_status_excluded_at_compile_time(***test***) 114
set_status_excluded_at_compile_time(***test, value***) 114
has_status_covered(***test***) 115
set_status_covered(***test, value***) 115
has_status_proven(***test***) 115
set_status_proven(***test, value***) 116
has_status_excluded_at_report_time(***test***) 116
set_status_excluded_at_report_time(***test, value***) 116
has_status_attempted(***test***) 116
set_status_attempted(***test, value***) 117
has_status_excluded(***test***) 117
set_status_excluded(*test, value*) 117
has_status_partially_excluded(***test***) 118
set_status_partially_excluded(***test, value***) 118
has_status_partially_attempted(***test***) 118
set_status_partially_attempted(***test, value***) 118
set_status(***test, value***) 119
name() 119

Fsm Coverage 119

class cov.FsmMetric 119

instance_handle() 119
child_handles() 120
type(***test=None, is_get_enum=False***) 120
name() 120
full_name() 121

Fsm 121

class cov.Fsm 121

scope_handle() 121
child_handles() 121
type(***test=None, is_get_enum=False***) 121
line_no(***test=None***) 122
name() 122
file_name() 122

States 123

class cov.States 123

scope_handle() 123

child_handles() 123

type(***test=None, is_get_enum=False***) 123

coverable(***test=None***) 124

covered(***test=None***) 124

status(***test=None***) 124

has_status_unreachable(***test***) 125

set_status_unreachable(***test, value***) 125

has_status_illegal(***test***) 125

set_status_illegal(***test, value***) 125

has_status_excluded_at_compile_time(***test***) 126

set_status_excluded_at_compile_time(***test, value***) 126

has_status_covered(***test***) 126

set_status_covered(***test, value***) 126

has_status_proven(***test***) 127

set_status_proven(***test, value***) 127

has_status_excluded_at_report_time(***test***) 127

set_status_excluded_at_report_time(***test, value***) 128

has_status_attempted(***test***) 128

set_status_attempted(***test, value***) 128

has_status_excluded(***test***) 128

set_status_excluded(***test, value***) 129

has_status_partially_excluded(***test***) 129

set_status_partially_excluded(***test, value***) 129

has_status_partially_attempted(***test***) 130

set_status_partially_attempted(***test, value***) 130

set_status(***test, value***) 130

name() 130

Transitions 131

class cov.Transitions 131

scope_handle() 131

child_handles() 131

type(***test=None, is_get_enum=False***) 131

coverable(***test=None***) 132

covered(***test=None***) 132

status(***test=None***) 132

has_status_unreachable(test) 133

set_status_unreachable(***test, value***) 133

has_status_illegal(***test***) 133

set_status_illegal(***test, value***) 133

has_status_excluded_at_compile_time(***test***) 134

set_status_excluded_at_compile_time(***test, value***) 134

has_status_covered(***test***) 134

set_status_covered(***test, value***) . . . 134
has_status_proven(***test***) . . . 135
set_status_proven(***test, value***) . . . 135
has_status_excluded_at_report_time(***test***) . . . 135
set_status_excluded_at_report_time(***test, value***) . . . 136
has_status_attempted(***test***) . . . 136
set_status_attempted(***test, value***) . . . 136
has_status_excluded(***test***) . . . 136
set_status_excluded(***test, value***) . . . 137
has_status_partially_excluded(***test***) . . . 137
set_status_partially_excluded(***test, value***) . . . 137
has_status_partially_attempted(***test***) . . . 138
set_status_partially_attempted(***test, value***) . . . 138
set_status(***test, value***) . . . 138
name() . . . 138

Sequences . . . 139

class cov.Sequences . . . 139

scope_handle() . . . 139
child_handles() . . . 139
type(***test=None, is_get_enum=False***) . . . 139
coverable(***test=None***) . . . 140
covered(***test=None***) . . . 140
status(***test=None***) . . . 140
has_status_unreachable(test) . . . 141
set_status_unreachable(***test, value***) . . . 141
has_status_illegal(***test***) . . . 141
set_status_illegal(***test, value***) . . . 141
has_status_excluded_at_compile_time(***test***) . . . 142
set_status_excluded_at_compile_time(***test, value***) . . . 142
has_status_covered(***test***) . . . 142
set_status_covered(***test, value***) . . . 143
has_status_proven(***test***) . . . 143
set_status_proven(***test, value***) . . . 143
has_status_excluded_at_report_time(***test***) . . . 143
set_status_excluded_at_report_time(***test, value***) . . . 144
has_status_attempted(***test***) . . . 144
set_status_attempted(***test, value***) . . . 144
has_status_excluded(***test***) . . . 144
set_status_excluded(***test, value***) . . . 145
has_status_partially_excluded(***test***) . . . 145
set_status_partially_excluded(***test, value***) . . . 145
has_status_partially_attempted(***test***) . . . 146
set_status_partially_attempted(***test, value***) . . . 146
set_status(***test, value***) . . . 146

name() 147

StateBin 147

class cov.StateBin 147

covered_test_handles(***test***) 147

scope_handle() 147

type(***test=None, is_get_enum=False***) 147

line_no(***test=None***) 148

coverable(***test=None***) 148

covered(***test=None***) 148

set_covered(***test, value***) 149

status(***test=None***) 149

has_status_unreachable(***test***) 149

set_status_unreachable(***test, value***) 150

has_status_illegal(***test***) 150

set_status_illegal(***test, value***) 150

has_status_excluded_at_compile_time(***test***) 150

set_status_excluded_at_compile_time(***test, value***) 151

has_status_covered(***test***) 151

set_status_covered(***test, value***) 151

has_status_proven(***test***) 152

set_status_proven(***test, value***) 152

has_status_excluded_at_report_time(***test***) 152

set_status_excluded_at_report_time(***test, value***) 152

has_status_attempted(***test***) 153

set_status_attempted(***test, value***) 153

has_status_excluded(***test***) 153

set_status_excluded(***test, value***) 153

has_status_partially_excluded(***test***) 154

set_status_partially_excluded(***test, value***) 154

has_status_partially_attempted(***test***) 154

set_status_partially_attempted(*test, value*) 155

set_status(***test, value***) 155

name() 155

file_name() 155

TransBin 156

class cov.TransBin 156

state_bin_handles() 156

covered_test_handles(***test***) 156

scope_handle() 156

type(***test=None, is_get_enum=False***) 157

line_no(***test=None***) 157

coverable(***test=None***) 157

covered(***test=None***) 158

set_covered(***test, value***) . . . 158
status(***test=None***) . . . 158
has_status_unreachable(***test***) . . . 159
set_status_unreachable(***test, value***) . . . 159
has_status_illegal(***test***) . . . 159
set_status_illegal(***test, value***) . . . 159
has_status_excluded_at_compile_time(***test***) . . . 160
set_status_excluded_at_compile_time(***test, value***) . . . 160
has_status_covered(***test***) . . . 160
set_status_covered(***test, value***) . . . 160
has_status_proven(***test***) . . . 161
set_status_proven(***test, value***) . . . 161
has_status_excluded_at_report_time(***test***) . . . 161
set_status_excluded_at_report_time(***test, value***) . . . 162
has_status_attempted(***test***) . . . 162
set_status_attempted(***test, value***) . . . 162
has_status_excluded(***test***) . . . 162
set_status_excluded(***test, value***) . . . 163
has_status_partially_excluded(***test***) . . . 163
set_status_partially_excluded(***test, value***) . . . 163
has_status_partially_attempted(***test***) . . . 164
set_status_partially_attempted(***test, value***) . . . 164
set_status(***test, value***) . . . 164
name() . . . 164
file_name() . . . 165

SeqBin . . . 165

class cov.SeqBin . . . 165
state_bin_handles() . . . 165
scope_handle() . . . 165
type(***test=None, is_get_enum=False***) . . . 166
coverable(***test=None***) . . . 166
covered(***test=None***) . . . 166
status(***test=None***) . . . 167
has_status_unreachable(***test***) . . . 167
set_status_unreachable(***test, value***) . . . 167
has_status_illegal(***test***) . . . 167
set_status_illegal(***test, value***) . . . 168
has_status_excluded_at_compile_time(***test***) . . . 168
set_status_excluded_at_compile_time(***test, value***) . . . 168
has_status_covered(***test***) . . . 168
set_status_covered(***test, value***) . . . 169
has_status_proven(***test***) . . . 169
set_status_proven(***test, value***) . . . 169
has_status_excluded_at_report_time(***test***) . . . 170

set_status_excluded_at_report_time(***test, value***) 170
has_status_attempted(***test***) 170
set_status_attempted(***test, value***) 170
has_status_excluded(***test***) 171
set_status_excluded(***test, value***) 171
has_status_partially_excluded(***test***) 171
set_status_partially_excluded(***test, value***) 172
has_status_partially_attempted(***test***) 172
set_status_partially_attempted(***test, value***) 172
set_status(***test, value***) 172
name() 173
file_name() 173

ConditionMetric 173

class cov.ConditionMetric 173

instance_handle() 173
child_handles() 174
type(***test=None, is_get_enum=False***) 174
name() 174
full_name() 175

Condition 175

class cov.Condition 175

scope_handle() 175
child_handles() 175
type(***test=None, is_get_enum=False***) 175
line_no(***test=None***) 176
coverable(***test=None***) 176
covered(***test=None***) 176
status(***test=None***) 177
has_status_unreachable(***test***) 177
set_status_unreachable(***test, value***) 177
has_status_illegal(***test***) 178
set_status_illegal(***test, value***) 178
has_status_excluded_at_compile_time(***test***) 178
set_status_excluded_at_compile_time(***test, value***) 178
has_status_covered(***test***) 179
set_status_covered(***test, value***) 179
has_status_proven(***test***) 179
set_status_proven(***test, value***) 179
has_status_excluded_at_report_time(***test***) 180
set_status_excluded_at_report_time(***test, value***) 180
has_status_attempted(***test***) 180
set_status_attempted(***test, value***) 181
has_status_excluded(***test***) 181

set_status_excluded(***test, value***) 181
has_status_partially_excluded(***test***) 181
set_status_partially_excluded(***test, value***) 182
has_status_partially_attempted(***test***) 182
set_status_partially_attempted(***test, value***) 182
set_status(***test, value***) 183
is_event_condition(***test=None***) 183
name() 183
file_name() 183

ConditionBin 184

class cov. ConditionBin 184

condition_term_handles() 184

ConditionTerm 192

class cov.ConditionTerm 192

type(***test=None, is_get_enum=False***) 192
name() 193

Branch Metric 193

class cov.BranchMetric 193

instance_handle() 193
child_handles() 193
type(***test=None, is_get_enum=False***) 194
name() 194
full_name() 194

Branch 195

class cov.Branch 195

scope_handle() 195
child_handles() 195
type(***test=None, is_get_enum=False***) 195
line_no(***test=None***) 196
coverable(***test=None***) 196
covered(***test=None***) 196
status(***test=None***) 196
has_status_unreachable(***test***) 197
set_status_unreachable(***test, value***) 197
has_status_illegal(***test***) 197
set_status_illegal(***test, value***) 198
has_status_excluded_at_compile_time(***test***) 198
set_status_excluded_at_compile_time(***test, value***) 198
has_status_covered(***test***) 198
set_status_covered(***test, value***) 199
has_status_proven(***test***) 199
set_status_proven(***test, value***) 199

has_status_excluded_at_report_time(***test***) 199
set_status_excluded_at_report_time(***test, value***) 200
has_status_attempted(***test***) 200
set_status_attempted(***test, value***) 200
has_status_excluded(***test***) 201
set_status_excluded(***test, value***) 201
has_status_partially_excluded(***test***) 201
set_status_partially_excluded(***test, value***) 201
has_status_partially_attempted(***test***) 202
set_status_partially_attempted(***test, value***) 202
set_status(***test, value***) 202
name() 203
file_name() 203

BranchBin 203

class cov.BranchBin 203

branch_term_handles() 203
covered_test_handles(***test***) 203
scope_handle() 204
type(***test=None, is_get_enum=False***) 204
line_no(***test=None***) 204
coverable(***test=None***) 205
covered(***test=None***) 205
set_covered(***test, value***) 205
status(***test=None***) 206
has_status_unreachable(***test***) 206
set_status_unreachable(***test, value***) 206
has_status_illegal(***test***) 206
set_status_illegal(***test, value***) 207
has_status_excluded_at_compile_time(***test***) 207
set_status_excluded_at_compile_time(***test, value***) 207
has_status_covered(***test***) 208
set_status_covered(***test, value***) 208
has_status_proven(***test***) 208
set_status_proven(***test, value***) 208
has_status_excluded_at_report_time(***test***) 209
set_status_excluded_at_report_time(***test, value***) 209
has_status_attempted(***test***) 209
set_status_attempted(***test, value***) 209
has_status_excluded(***test***) 210
set_status_excluded(***test, value***) 210
has_status_partially_excluded(***test***) 210
set_status_partially_excluded(***test, value***) 211
has_status_partially_attempted(***test***) 211
set_status_partially_attempted(***test, value***) 211

set_status(***test, value***) 211
name() 212
file_name() 212

BranchTerm 212

class cov.BranchTerm 212

type(***test=None, is_get_enum=False***) 212
value(***test=None, is_get_enum=False***) 213
line_no(***test=None***) 213
name() 214
file_name() 214

AssertMetric 214

class cov.AssertMetric 214

instance_handle() 214
child_handles() 214
type(***test=None, is_get_enum=False***) 215
name() 215
full_name() 215

Assert 216

class cov .Assert 216

scope_handle() 216
child_handles() 216
type(***test=None, is_get_enum=False***) 216
line_no(***test=None***) 217
coverable(***test=None***) 217
covered(***test=None***) 217
status(***test=None***) 217
has_status_unreachable(***test***) 218
set_status_unreachable(***test, value***) 218
has_status_illegal(***test***) 218
set_status_illegal(***test, value***) 219
has_status_excluded_at_compile_time(***test***) 219
set_status_excluded_at_compile_time(***test, value***) 219
has_status_covered(***test***) 219
set_status_covered(***test, value***) 220
has_status_proven(***test***) 220
set_status_proven(***test, value***) 220
has_status_excluded_at_report_time(***test***) 220
set_status_excluded_at_report_time(***test, value***) 221
has_status_attempted(***test***) 221
set_status_attempted(***test, value***) 221
has_status_excluded(***test***) 222
set_status_excluded(***test, value***) 222
has_status_partially_excluded(***test***) 222

set_status_partially_excluded(***test, value***) 222
has_status_partially_attempted(***test***) 223
set_status_partially_attempted(***test, value***) 223
set_status(***test, value***) 223
severity(***test=None***) 224
category(***test=None***) 224
name() 224
file_name() 224

CoverProperty 225

class cov.CoverProperty 225

scope_handle() 225
child_handles() 225
type(***test=None, is_get_enum=False***) 225
line_no(***test=None***) 226
coverable(***test=None***) 226
covered(***test=None***) 226
status(***test=None***) 227
has_status_unreachable(*test*) 227
set_status_unreachable(***test, value***) 227
has_status_illegal(***test***) 227
set_status_illegal(***test, value***) 228
has_status_excluded_at_compile_time(***test***) 228
set_status_excluded_at_compile_time(***test, value***) 228
has_status_covered(***test***) 228
set_status_covered(***test, value***) 229
has_status_proven(***test***) 229
set_status_proven(***test, value***) 229
has_status_excluded_at_report_time(***test***) 230
set_status_excluded_at_report_time(***test, value***) 230
has_status_attempted(***test***) 230
set_status_attempted(***test, value***) 230
has_status_excluded(***test***) 231
set_status_excluded(***test, value***) 231
has_status_partially_excluded(***test***) 231
set_status_partially_excluded(*test, value*) 232
has_status_partially_attempted(*test*) 232
set_status_partially_attempted(*test, value*) 232
set_status(*test, value*) 232
severity(***test=None***) 233
category(***test=None***) 233
name() 233
file_name() 234

CoverSequence 234

class cov.CoverSequence 234
scope_handle() 234
child_handles() 234
type(***test=None, is_get_enum=False***) 234
line_no(***test=None***) 235
coverable(***test=None***) 235
covered(***test=None***) 235
status(***test=None***) 236
has_status_unreachable(***test***) 236
set_status_unreachable(***test, value***) 236
has_status_illegal(***test***) 237
set_status_illegal(***test, value***) 237
has_status_excluded_at_compile_time(***test***) 237
set_status_excluded_at_compile_time(***test, value***) 237
has_status_covered(***test***) 238
set_status_covered(***test, value***) 238
has_status_proven(***test***) 238
set_status_proven(***test, value***) 238
has_status_excluded_at_report_time(***test***) 239
set_status_excluded_at_report_time(***test, value***) 239
has_status_attempted(***test***) 239
set_status_attempted(***test, value***) 240
has_status_excluded(***test***) 240
set_status_excluded(***test, value***) 240
has_status_partially_excluded(***test***) 240
set_status_partially_excluded(***test, value***) 241
has_status_partially_attempted(***test***) 241
set_status_partially_attempted(***test, value***) 241
set_status(***test, value***) 242
severity(***test=None***) 242
category(***test=None***) 242
name() 242
file_name() 243
SuccessBin 243
class cov. SuccessBin 243
covered_test_handles(***test***) 243
scope_handle() 243
type(***test=None, is_get_enum=False***) 244
count(***test=None***) 244
name() 244
AttemptBin 245
class cov. AttemptBin 245
covered_test_handles(***test***) 245

scope_handle() 245
type(***test=None, is_get_enum=False***) 245
count(***test=None***) 246
name() 246

FailureBin 246

class cov. FailureBin 246
covered_test_handles(***test***) 246
scope_handle() 247
type(***test=None, is_get_enum=False***) 247
count(***test=None***) 247
name() 248

VacuousBin 248

class cov.VacuousBin 248
covered_test_handles(***test***) 248
scope_handle() 248
type(***test=None, is_get_enum=False***) 248
count(***test=None***) 249
name() 249

IncompleteBin 250

class cov.IncompleteBin 250
covered_test_handles(***test***) 250
scope_handle() 250
type(***test=None, is_get_enum=False***) 250
count(***test=None***) 251
name() 251

FirstmatchBin 251

class cov.FirstmatchBin 251
covered_test_handles(***test***) 251
scope_handle() 252
type(***test=None, is_get_enum=False***) 252
count(***test=None***) 252
name() 253

TestbenchMetric 253

***class* cov.TestbenchMetric** 253
test_handle() 253
child_handles() 253
type(***test=None, is_get_enum=False***) 253
name() 254

Covergroup 254

class cov.Covergroup 254

cover_instance_handles() 254
scope_handle() 255
child_handles() 255
type(***test=None, is_get_enum=False***) 255
line_no(***test=None***) 255
coverable(***test=None***) 256
covered(***test=None***) 256
status(***test=None***) 256
has_status_unreachable(***test***) 257
set_status_unreachable(***test, value***) 257
has_status_illegal(***test***) 257
set_status_illegal(***test, value***) 257
has_status_excluded_at_compile_time(***test***) 258
set_status_excluded_at_compile_time(***test, value***) 258
has_status_covered(***test***) 258
set_status_covered(***test, value***) 258
has_status_proven(***test***) 259
set_status_proven(***test, value***) 259
has_status_excluded_at_report_time(***test***) 259
set_status_excluded_at_report_time(***test, value***) 260
has_status_attempted(***test***) 260
set_status_attempted(***test, value***) 260
has_status_excluded(***test***) 260
set_status_excluded(***test, value***) 261
has_status_partially_excluded(***test***) 261
set_status_partially_excluded(***test, value***) 261
has_status_partially_attempted(***test***) 262
set_status_partially_attempted(***test, value***) 262
set_status(***test, value***) 262
per_instance(***test=None***) 262
weight(***test=None***) 263
name() 263
file_name() 263

CoverInstance 264

class cov.CoverInstance 264

scope_handle() 264
child_handles() 264
type(***test=None, is_get_enum=False***) 264
line_no(***test=None***) 265
coverable(***test=None***) 265
covered(***test=None***) 265
status(***test=None***) 265
has_status_unreachable(***test***) 266
set_status_unreachable(***test, value***) 266

has_status_illegal(***test***) 266
set_status_illegal(***test, value***) 267
has_status_excluded_at_compile_time(***test***) 267
set_status_excluded_at_compile_time(***test, value***) 267
has_status_covered(***test***) 267
set_status_covered(***test, value***) 268
has_status_proven(***test***) 268
set_status_proven(***test, value***) 268
has_status_excluded_at_report_time(***test***) 268
set_status_excluded_at_report_time(***test, value***) 269
has_status_attempted(***test***) 269
set_status_attempted(***test, value***) 269
has_status_excluded(***test***) 270
set_status_excluded(***test, value***) 270
has_status_partially_excluded(***test***) 270
set_status_partially_excluded(***test, value***) 270
has_status_partially_attempted(***test***) 271
set_status_partially_attempted(***test, value***) 271
set_status(***test, value***) 271
weight(***test=None***) 272
goal(***test=None***) 272
at_least(***test=None***) 272
auto_bin_max(***test=None***) 272
cross_num_print_missing(***test=None***) 273
detect_overlap(***test=None***) 273
name() 273
file_name() 274
comment() 274

Cross 274

class cov.Cross 274

scope_handle() 274
child_handles() 274
component_handles() 275
type(***test=None, is_get_enum=False***) 275
line_no(***test=None***) 275
coverable(***test=None***) 276
covered(***test=None***) 276
status(***test=None***) 276
has_status_unreachable(***test***) 276
set_status_unreachable(***test, value***) 277
has_status_illegal(***test***) 277
set_status_illegal(***test, value***) 277
has_status_excluded_at_compile_time(***test***) 277
set_status_excluded_at_compile_time(***test, value***) 278

has_status_covered(***test***) 278
set_status_covered(***test, value***) 278
has_status_proven(***test***) 279
set_status_proven(***test, value***) 279
has_status_excluded_at_report_time(***test***) 279
set_status_excluded_at_report_time(***test, value***) 279
has_status_attempted(***test***) 280
set_status_attempted(***test, value***) 280
has_status_excluded(***test***) 280
set_status_excluded(***test, value***) 281
has_status_partially_excluded(***test***) 281
set_status_partially_excluded(***test, value***) 281
has_status_partially_attempted(***test***) 281
set_status_partially_attempted(***test, value***) 282
set_status(***test, value***) 282
weight(***test=None***) 282
goal(***test=None***) 283
at_least(***test=None***) 283
cross_num_print_missing(***test=None***) 283
name() 283
file_name() 284
`comment`() 284

Coverpoint 284

class cov.Coverpoint 284

scope_handle() 284
child_handles() 284
type(***test=None, is_get_enum=False***) 285
line_no(***test=None***) 285
coverable(***test=None***) 285
covered(***test=None***) 286
status(***test=None***) 286
has_status_unreachable(***test***) 286
set_status_unreachable(***test, value***) 287
has_status_illegal(***test***) 287
set_status_illegal(***test, value***) 287
has_status_excluded_at_compile_time(***test***) 287
set_status_excluded_at_compile_time(***test, value***) 288
has_status_covered(***test***) 288
set_status_covered(***test, value***) 288
has_status_proven(***test***) 288
set_status_proven(***test, value***) 289
has_status_excluded_at_report_time(***test***) 289
set_status_excluded_at_report_time(***test, value***) 289
has_status_attempted(***test***) 290

set_status_attempted(***test, value***) 290
has_status_excluded(***test***) 290
set_status_excluded(***test, value***) 290
has_status_partially_excluded(***test***) 291
set_status_partially_excluded(***test, value***) 291
has_status_partially_attempted(***test***) 291
set_status_partially_attempted(***test, value***) 292
set_status(***test, value***) 292
weight(***test=None***) 292
goal(***test=None***) 292
at_least(***test=None***) 293
auto_bin_max(***test=None***) 293
detect_overlap(***test=None***) 293
name() 294
file_name() 294
`comment`() 294

CoverBin 294

class cov.CoverBin 294

covered_test_handles(***test***) 294
scope_handle() 295
type(***test=None, is_get_enum=False***) 295
coverable(***test=None***) 295
covered(***test=None***) 296
count(***test=None***) 296
count_goal(***test=None***) 296
status(***test=None***) 296
has_status_unreachable(***test***) 297
set_status_unreachable(***test, value***) 297
has_status_illegal(***test***) 297
set_status_illegal(***test, value***) 298
has_status_excluded_at_compile_time(***test***) 298
set_status_excluded_at_compile_time(***test, value***) 298
has_status_covered(***test***) 298
set_status_covered(***test, value***) 299
has_status_proven(***test***) 299
set_status_proven(***test, value***) 299
has_status_excluded_at_report_time(***test***) 300
set_status_excluded_at_report_time(***test, value***) 300
has_status_attempted(***test***) 300
set_status_attempted(***test, value***) 300
has_status_excluded(***test***) 301
set_status_excluded(***test, value***) 301
has_status_partially_excluded(***test***) 301
set_status_partially_excluded(***test, value***) 301

has_status_partially_attempted(***test***) 302
set_status_partially_attempted(***test, value***) 302
set_status(***test, value***) 302
name() 303

PowerDomain 303

class cov.PowerDomain 303
test_handle() 303
power_domain_handles() 303
power_measure handle() 304
scope_handle() 304
type(***test=None, is_get_enum=False***) 304
line_no(***test=None***) 304
name() 305
file_name() 305

PowerMeasure 305

class cov.PowerMeasure 305
power_metric_handle() 305
scope_handle() 306
type(***test=None, is_get_enum=False***) 306
name() 306

PowerMetric 307

class cov . PowerMetric 307
test_handle() 307
power_measure_handle() 307
child_handles() 307
type(***test=None, is_get_enum=False***) 307
name() 308

Covered Test 308

test_handle() 308
type(***test=None, is_get_enum=False***) 309
count(***test=None***) 309
name() 309

5. L0 APIs 310

cov.open(vdb_dir) 310

cov.merge_test(***dst_test, src_test, map_file=None***) 310

Preface

The Python-Based NPI Coverage Model User Guide provides information to let you access coverage database generated by verification tools such as VCS simulator.

Customer Support

For any online access to the self-help resources, you can refer to the documentation and searchable knowledge base available in SolvNetPlus.

To obtain support for your Verdi product, choose one of the following:

- Open a case through SolvNetPlus.

  Go to https://solvnetplus.synopsys.com/s/contactsupport and provide the requested information, including:

  - Product L1 as Verdi
  - Case Type

  Fill in the remaining fields according to your environment and issue.

- Send an e-mail message to verdi_support@synopsys.com.

  Include product name (L1), sub-product name/technology (L2), and product version in your e-mail, so it can be routed correctly.

  Your e-mail will be acknowledged by automatic reply and assigned a Case number along with Case reference ID in the subject (ref:_...:ref).

  For any further communication on this Case via e-mail, send e-mail to verdi_support@synopsys.com and ensure to have the same Case ref ID in the subject header or else it will open duplicate cases.

- You can call for support at:

  https://www.synopsys.com/support/global-support-centers.html

Note:

In general, we need to be able to reproduce the problem in order to fix it, so a simple model demonstrating the error is the most effective way for us to identify the bug. If that is not possible, then provide a detailed explanation of the problem along with complete error and corresponding code, if any/permissible.

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

1

Introduction to Python Based NPI

Python-Based NPI APIs support six models:

- Coverage
- Language
- Netlist
- Text
- Waveform
- Waveform Writer

Each model have their own APIs to let you be able to traverse data objects and obtain objects' properties like the existing C-Based or Tcl-Based NPI APIs.

In this guide, the environment setting for using **Python-Based NPI APIs for Coverage** is demonstrated.

Package and Modules

Packages

The Python-based NPI package name is “pynpi”, and it is placed at `$VERDI_HOME/share/NPI/python`.

Modules

There are seven modules inside the “pynpi” package: npisys, lang, netlist, text, cov waveform and waveformw. The first module, npisys, is the system model for initialization, loading design and exit. The other modules represent language model, netlist model, text model, coverage model, wave model and waveform writer model respectively

Module Functions and Class Objects

L0 Module Functions

Every module provides some L0 (level 0) functions to let you get the class objects. These functions return a class object or a list of class objects, and they follow the specification of the existing L0 APIs provided in C or Tcl.

L1 Module Functions

Similar to L0 module functions, every module also provides some L1 (level 1) functions to let you get advanced information based on the results obtained by L0 module functions. These functions follows the specification of the existing L1 APIs provided in C or Tcl.

Class Objects

The class object is similar to the so-called handle in NPI C APIs. The most difference is that some basic L0 APIs in C and Tcl will become class method function. These L0 APIs are usually to get integer value, string value, 1-to-1 method to get a handle, and 1-to-many method to get handle iterator.

User Interface and Use Flow

Environment and Library Setting

The python library setting flow of using Python-Based NPI APIs contains four parts:

1. Check your Python's version:

   Python-Based NPI APIs need the Python version greater than 3.6.0.

2. Environment setting for "VERDI_HOME" is required for Python-based NPI. Ensure that you set it up before running program.

3. Add python library path into your python code before loading Python-Based NPI by the following commands:

   ```
   rel_lib_path = os.environ['VERDI_HOME'] + '/share/NPI/python'
   sys.path.append(os.path.abspath(rel_lib_path))
   ```

4. Import module "npisys" for using the function of NPI initialization and exit from pynpi package.

   ```
   from pynpi import npisys
   ```

5. Import the module you need from pynpi package. For example, if you want to use Coverage model, you can import module like the following:

   ```
   from pynpi import cov
   ```

6. Note that initialization function `npisys.init()` must be called before writing your code by using any other modules. Also, you must call `npisys.end()` after finishing your code. Following is a simple example to demonstrate how to use coverage model by Python-Based NPI APIs.

   Python program to use NPI models: (`demo.py`)

   ```
   #!/global/freeware/Linux/2.X/python-3.6.0/bin/python

   import sys, os
   rel_lib_path = os.environ["VERDI_HOME"] + "/share/NPI/python"
   sys.path.append(os.path.abspath(rel_lib_path))
   from pynpi import npisys
   from pynpi import cov

   # Initialize NPI
   if not npisys.init(sys.argv):
   print("Error: Fail to initialize NPI")
   assert 0

   # Load design (if needed, depends on models)
   if not npisys.load_design(sys.argv):
   print("Error: Fail to load design")
   assert 0

   # Beginning of your code here --------------------
   #
   # Example code can be found in later chapters
   #
   # End of your code -------------------------------

   # End NPI
   npisys.end()
   ```

   C shell script example to setup environment and execute Python program: (`run_demo`)

   ```
   #!/bin/csh -f
   # Setup your $VERDI_HOME here
   setenv VERDI_HOME [YOUR_VERDI_HOME_PATH]
   # run the python program
   # - Input arguments depend on your program design
   # - If loading design is required, you can pass the options like
   ./demo.py -sv demo.v
   ```

To run the files, put the above files in the same directory and execute the `run_demo` C shell script.

```
./run_demo
```

2

Module npisys

This chapter includes the following topics:

- Overview
- L0 APIs

Overview

Module npisys is for setting Python-based NPI. You must call `npisys.init()` before using any other NPI modules and call `npisys.end()` after using any other NPI modules.

L0 APIs

Following are the public L0 APIs for system module:

npisys.init(*pyArgvList*)

System initialization for Python-Based NPI.

Parameters: **pyArgList (str list)** – input argument list, for example, sys.argv

Returns: Return 1 if successful. Otherwise, return 0.

Return type: int

Example

```
>>>npisys.init(sys.argv)
```

npisys.load_design(*pyArgvList*)

Load design for Python-Based NPI.

Parameters: **pyArgList (str list)** – input argument list. For example, sys.argv

Returns: Return 1 if successful. Otherwise, return 0.

Return type: int

Example

```
>>>npisys.load_design(sys.argv)
```

npisys.end()

Clean NPI-related settings and data.

Parameters: none

Returns: Return 1 if successful. Otherwise, return 0.

Return type: int

Example

```
>>>npisys.end()
```

3

Python-Based NPI Coverage Model

This chapter includes the following topics:

- Overview
- Quick Start
- Class Objects

Overview

The NPI coverage model lets you to access coverage database generated by verification tools such as VCS simulator. With coverage model, you can create custom reports and tools for displaying and analyzing coverage data.

Quick Start

Example Code:

```
import sys
import os
rel_lib_path = os.environ['VERDI_HOME'] + '/share/NPI/python'
sys.path.append(os.path.abspath(rel_lib_path))
from pynpi import npisys
from pynpi import cov

# -- traverse coverage toggle info -------------------------------------
def trv_toggle(hdl, indentCnt, test, fp_w):
    childList = hdl.child_handles()
    for child in childList:
        for i in range(indentCnt):
            fp_w.write(' ')
        fp_w.write('# '+child.type()+': '+child.name()+'
cov:'+str(child.covered(test)))
        fp_w.write('/'+str(child.coverable(test))+'\n')
        trv_toggle(child, indentCnt+1, test, fp_w)
        cov.release_handle(child)

# -- traverse coverage instances ---------------------------------------
def trv_cov_inst(hdl, indentCnt, test, fp_w):
    instList = hdl.instance_handles()
    for inst in instList:
        for i in range(indentCnt):
            fp_w.write(' ')
        hdlType = inst.type()
        fp_w.write(hdlType+': ')
        if hdlType == 'npiCovDatabase':
            fp_w.write(inst.name()+'\n')
        else:
            fp_w.write(inst.full_name()+'\n')
        tm = inst.toggle_metric_handle()
        if (tm):
            trv_toggle(tm, indentCnt, test, fp_w)
        trv_cov_inst(inst, indentCnt+1, test, fp_w)
        cov.release_handle(inst)

# -- init --------------------------------------------------------------
res = npisys.init(sys.argv)
if res != 1:
    print('[Error] Failed to do npi init')
    sys.exit(1)

# -- open log to write -------------------------------------------------
fp_w = open('test.log', 'w')
vdbName = 'merged.vdb'

# -- traverse all the toggle bins and write thier coverage info --------
covDb = cov.open(vdbName)
if (covDb):
    fp_w.write('load vdb: '+covDb.name()+'\n')

    # -- merge tests ---------------------------------------------------
    mergedTest = None
    testList = covDb.test_handles()
    for test in testList:
        fp_w.write(test.name()+'\n')
        if (mergedTest == None):
            mergedTest = test
        else:
            mergedTest = cov.merge_test(mergedTest, test)
    fp_w.write('mergedTest: '+mergedTest.name()+'\n')
    trv_cov_inst(covDb, 1, mergedTest, fp_w)
    covDb.close()

# -- close log and end -------------------------------------------------
fp_w.close()
npisys.end()
```

Execution Result

```
open vdb: merged.vdb
mergedTest: merged.vdb/test
npiCovInstance: test_jukebox
```

```
npiCovInstance: test_jukebox.st0
  # npiCovSignal: trki cov:4/4
    # npiCovSignalBit: trki[0] cov:2/2
      # npiCovToggleBin: 0 -> 1 cov:1/1
      # npiCovToggleBin: 1 -> 0 cov:1/1
    # npiCovSignalBit: trki[1] cov:2/2
      # npiCovToggleBin: 0 -> 1 cov:1/1
      # npiCovToggleBin: 1 -> 0 cov:1/1
 # npiCovSignal: dski cov:2/4
    # npiCovSignalBit: dski[0] cov:1/2
      # npiCovToggleBin: 0 -> 1 cov:0/1
      # npiCovToggleBin: 1 -> 0 cov:1/1
    # npiCovSignalBit: dski[1] cov:1/2
      # npiCovToggleBin: 0 -> 1 cov:1/1
      # npiCovToggleBin: 1 -> 0 cov:0/1
```

Class Objects

In coverage module, you can use the object diagrams in the existing C/C++-Based NPI object diagrams to see the usage of Python-Based NPI class objects by the following rules:

1. Every int or string property can be called as a class function whose function name is that int or string property after removing prefix “npiCov” and making the rest part become snake case style. Here is the table to show the mapping:

C/C++-based NPI Usage	Python-based NPI Usage
npi_cov_get_str(npiCovName, hdl)	hdl.name()
npi_cov_get(npiCovLineNo, hdl)	hdl.line_no()

2. Every 1-to-1 method can be called as a class function whose function name is that method plus “_method” after removing prefix “npiCov” and making the rest part become snake case style. Similar to 1-to-1 method mapping, 1-to-many method mapping only needs to change from “_method” to “_methods”. Note that the return object of 1-to-many method is an iterator in C/C++, but a list in Python. Here is the table to show the mapping:

C/C++-based NPI Usage	Python-based NPI Usage
npi_cov_handle(npiCovLineMetric, hdl)	hdl.line_metric_handle()
npi_cov_handles(npiCovChild, hdl)	hdl.child_handles()

Following are all the class object diagrams, relationships and properties:

Object Category	Class Objects
Coverage Database	Database, Test, TestInfo
Instance	Instance
Line Coverage	LineCoverage, Block, StmtBin
Toggle Coverage	Toggle Coverage, Signal, SignalBit, Toggle Bin
Finite State Machine Coverage	Fsm Coverage, Fsm, States, Transitions, Sequences, StateBin, TransBin, SeqBin
Condition Coverage	ConditionMetric, Condition, ConditionBin, ConditionTerm
Branch Coverage	Branch Metric, Branch, BranchBin, BranchTerm
Assert Coverage	AssertMetric, AssertAssertCoverProperty, CoverSequence, SuccessBin, AttemptBin, FailureBin, VacuousBin, IncompleteBin, FirstmatchBin
Testbench Coverage	TestbenchMetric, Covergroup, CoverInstance, Cross, Coverpoint, CoverBin
Power Coverage	PowerDomain, PowerMeasure, PowerMetric, Covergroup
Covered Test	Covered Test

For more information about Coverage Object Category, see the “NPI Coverage Model” chapter in the *VC Apps Native Programming Interface (NPI) Guide*.

4

Class Objects

This chapter includes the following topics:

- Database
- Test
- Instance
- LineCoverage
- Block
- StmtBin
- Toggle Coverage
- Signal
- SignalBit
- Toggle Bin
- Fsm Coverage
- Fsm
- States
- Transitions
- Sequences
- StateBin
- TransBin
- SeqBin
- ConditionMetric
- Condition
- ConditionBin

- ConditionTerm
- Branch Metric
- Branch
- BranchBin
- BranchTerm
- AssertMetric
- Assert
- CoverProperty
- CoverSequence
- SuccessBin
- AttemptBin
- FailureBin
- VacuousBin
- IncompleteBin
- FirstmatchBin
- TestbenchMetric
- Covergroup
- CoverInstance
- Cross
- Coverpoint
- CoverBin
- PowerDomain
- PowerMeasure
- PowerMetric
- Covered Test

Database

class cov.Database

test_handles()

Get test handle list.

Returns: Return the handle list obtained by 1-many method: test

Return type: list of Test

Example

```
>>> hdlList = ref.test_handles()
>>> hdlList[0].type()
'npiCovTest'
```

instance_handles()

Get instance handle list.

Returns: Return the handle list obtained by 1-many method: instance

Return type: list of cov.Instance

Example

```
>>> hdlList = ref.instance_handles()
>>> hdlList[0].type()
'npiCovInstance'
```

close()

close database handle.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> dbHdl.close()
1
```

handle_by_name(*name*)

Get instance handle by fullname.

Parameters: **name** (str) – The fullname of instance handle

Returns: Return the handle according to the fullname

Return type: Instance

Example

```
>>> instHdl = dbHdl.handle_by_name('top.a.b')
>>> instHdl.type()
'npiCovInstance'
```

test_by_name(*name*)

Get test handle by name.

Parameters: **name** (str) – The name of test handle

Returns: Return the handle according to the name

Return type:Test

Example

```
>>> testHdl = dbHdl.test_by_name('test03')
>>> testHdl.type()
'npiCovTest'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle.
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns:

Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovDatabase'
>>> hdl.type(test, True)
0
```

name()

Get name

Returns: Return the name string by property:name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

class cov.Test

database_handle()

Get database handle

Returns: Return the handle obtained by 1-1 method: database

Return type: cov.Database

Example

```
>>> hdl = ref.database_handle()
>>> hdl.type()
'npiCovDatabase'
```

assert_metric_handle()

Get assert_metric handle.

Returns: Return the handle obtained by 1-1 method: assert_metric

Return type: cov.AssertMetric

Example

```
>>> hdl = ref.assert_metric_handle()
>>> hdl.type()
'npiCovAssertMetric'
```

testbench_metric_handle()

Get testbench_metric handle.

Returns: Return the handle obtained by 1-1 method: testbench_metric

Return type: TestbenchMetric

Example

```
>>> hdl = ref.testbench_metric_handle()
>>> hdl.type()
'npiCovTestbenchMetric'
```

test_info_handles()

Get test_info handle list

Returns: Return the handle obtained by 1-many method: test_info

Return type: list of cov.TestInfo

Example

```
>>> hdlList = ref.test_info_handles()
>>> hdlList[0].type()
'npiCovTestInfo'
```

program_handles()

Get program handle list

Returns: Return the handle list obtained by 1-many method: program

Return type: list of cov.Program

Example

```
>>> hdlList = ref.program_handles()
>>> hdlList[0].type()
'npiCovProgram'
```

power_data_handles()

Get power_data handle list.

Returns: Return the handle list obtained by 1-many method: power_data

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.power_data_handles() >>> hdlList[0].line_no()
'45'
```

unload_test()

Unload test handle.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.unload_test()
1
```

save_test(*name*)

Save test handle.

Parameters: **name** (str) – The test name to save

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> saveTest = testHdl.save_test('save_test')
>>> saveTest.type()
'npiCovTest'
```

load_exclude_file(*name*)

Load exclude file.

Parameters: **name** (str) – The exclude file name to load

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.load_exclude_file('fifo.el')
1
```

save_exclude_file(*name, mode*)

Save exclude file.

Parameters:

- **name** (str) – The exclude file name to load
- **mode** (str)

‘w’ - Write: The file will be overwritten.

’a’ - Append: The excluded objects will be appended to the end of the

file if the file already exists

’ws’ - Write with strict mode: Similar to ‘w’, but covered objects will not

be saved to exclude file even if they have been set excluded

’as’ - Append with strict mode. Similar to ‘a’, but covered objects will not be saved to exclude file even if they have been set excluded

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.save_exclude_file('fifo.el', 'ws')
1
```

unload_exclusion()

Unload exclusion.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.unload_exclusion()
1
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test**(Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovTest'
>>> hdl.type(test, True)
1
```

name()

Get name

Returns: Return the name string by property:name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

class cov.TestInfo

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovTestInfo'
>>> hdl.type(test, True)
48
```

name()

Get name

Returns: Return the name string by property:name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Test

class cov.Test

database_handle()

Get database handle.

Returns: Return the handle obtained by 1-1 method: database

Return type: Database

Example

```
>>> hdl = ref.database_handle()
>>> hdl.type()
'npiCovDatabase'
```

testbench_metric_handle()

Get testbench_metric handle.

Returns: Return the handle obtained by 1-1 method: testbench_metric

Return type: TestbenchMetric

Example

```
>>> hdl = ref.testbench_metric_handle()
>>> hdl.type()
'npiCovTestbenchMetric'
```

power_data_handles()

Get power_data handle list.

Returns: Return the handle list obtained by 1-many method: power_data

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.power_data_handles() >>> hdlList[0].line_no()
'45'
```

unload_test()

Unload test handle.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.unload_test()
1
```

save_test(*name*)

Save test handle.

Parameters: **name** (str) – The test name to save

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> saveTest = testHdl.save_test('save_test')
>>> saveTest.type()
'npiCovTest'
```

load_exclude_file(*name*)

Load exclude file.

Parameters: **name** (str) – The exclude file name to load

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.load_exclude_file('fifo.el')
1
```

save_exclude_file(*name, mode*)

Save exclude file.

Parameters:

- **name** (str) – The exclude file name to load
- **mode** (str)

‘w’ - Write: The file will be overwritten.

’a’ - Append: The excluded objects will be appended to the end of the

file if the file already exists

’ws’ - Write with strict mode: Similar to ‘w’, but covered objects will not

be saved to exclude file even if they have been set excluded

’as’ - Append with strict mode. Similar to ‘a’, but covered objects will not

be saved to exclude file even if they have been set excluded

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.save_exclude_file('fifo.el', 'ws')
1
```

unload_exclusion()

Unload exclusion.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> testHdl.unload_exclusion()
1
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
11
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

size(*test=None*)

Get size.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.size(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.count(test)
1
```

count_goal(*test=None*)

Get count goal.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.count_goal(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(test)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

Feedback

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters:: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type int

Example

```
>>> hdl.set_status(test, 0)
1
```

per_instance(*test=None*)

Get per instance.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.per_instance(test)
1
```

is_mda(*test=None*)

Get is mda.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.is_mda(test)
1
```

is_port(*test=None*)

Get is port.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.is_port(test)
1
```

is_event_condition(*test=None*)

Get is event condition.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.is_event_condition(test)
1
```

severity(*test=None*)

Get severity.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.severity(test)
1
```

category(*test=None*)

Get category.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.category(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Instance

class cov.Instance

database_handle()

Get database handle.

Returns: Return the handle obtained by 1-1 method: database

Return type: Database

Example

```
>>> hdl = ref.database_handle()
>>> hdl.type()
'npiCovDatabase'
```

instance_handles()

Get instance handle list.

Returns: Return the handle list obtained by 1-many method: instance

Return type: list of cov.Instance

Example

```
>>> hdlList = ref.instance_handles()
>>> hdlList[0].type()
'npiCovInstance'
```

line_metric_handle()

Get line_metric handle.

Returns: Return the handle obtained by 1-1 method: line_metric

Return type: cov.LineMetric

Example

```
>>> hdl = ref.line_metric_handle()
>>> hdl.type()
'npiCovLineMetric'
```

toggle_metric_handle()

Get toggle_metric handle.

Returns: Return the handle obtained by 1-1 method: toggle_metric

Return type: cov.ToggleMetric

Example

```
>>> hdl = ref.toggle_metric_handle()
>>> hdl.type()
'npiCovToggleMetric'
```

fsm_metric_handle()

Get fsm_metric handle.

Returns: Return the handle obtained by 1-1 method: fsm_metric

Return type: cov.FsmMetric

Example

```
>>> hdl = ref.fsm_metric_handle()
>>> hdl.type()
'npiCovFsmMetric'
```

condition_metric_handle()

Get condition_metric handle.

Returns: Return the handle obtained by 1-1 method: condition_metric

Return type: cov.ConditionMetric

Example

```
>>> hdl = ref.condition_metric_handle()
>>> hdl.type()
'npiCovConditionMetric'
```

branch_metric_handle()

Get branch_metric handle.

Returns: Return the handle obtained by 1-1 method: branch_metric

Return type: cov.BranchMetric

Example

```
>>> hdl = ref.branch_metric_handle()
>>> hdl.type()
'npiCovBranchMetric'
```

assert_metric_handle()

Get assert_metric handle.

Returns: Return the handle obtained by 1-1 method: assert_metric

Return type: cov.AssertMetric

Example

```
>>> hdl = ref.assert_metric_handle()
>>> hdl.type()
'npiCovAssertMetric'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
2
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

def_name()

Get def name.

Returns: Return the name string by property: def_name

Return type: str

Example

```
>>> hdl.def_name()
'mm'
```

def_info()

Get def info.

Returns: Return the name string by property: def_info

Return type: str

Example

```
>>> hdl.def_info()
'mm ( parameter mp=1 )'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

LineCoverage

class cov.LineMetric

instance_handle()

Get instance handle.

Returns: Return the handle obtained by 1-1 method: instance.

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle.
- **is_get_enum (bool)** – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
4
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Block

class cov.Block

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
```

```
>>> hdl.type(test, True)
11
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(test, value)

Set status covered to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(test)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

StmtBin

class cov.StmtBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle.

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
12
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

>>> hdl.set_covered(test, 0)

1

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

Toggle Coverage

class cov.ToggleMetric

instance_handle()

Get instance handle.

Returns: Return the handle obtained by 1-1 method: instance.

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
5
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Signal

class cov.Signal

scope_handle()

Get scope handle

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns:Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
13
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

size(*test=None*)

Get size.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.size(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test)) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

is_mda(*test=None*)

Get is mda.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.is_mda(test)
1
```

is_port(*test=None*)

Get is port.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.is_port(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

SignalBit

class cov.SignalBit

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns:Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
14
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

size(*test=None*)

Get size.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.size(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

is_mda(*test=None*)

Get is mda.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.is_mda(test)
1
```

is_port(*test=None*)

Get is port.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.is_port(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

Toggle Bin

class cov.ToggleBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle.

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value.

Returns:Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
15
```

toggle_type(*test=None, is_get_enum=False*)

Get toggle type

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.toggle_type(test)
'npiCovToggle10'
>>> hdl.toggle_type(test, True)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Fsm Coverage

class cov.FsmMetric

instance_handle()

Get instance handle

Returns: Return the handle obtained by 1-1 method: instance

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns:Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
6
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Fsm

class cov.Fsm

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test**(Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
16
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

States

class cov.States

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns:Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
17
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle.

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

Feedback

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle.

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Transitions

class cov.Transitions

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
18
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(test)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Sequences

class cov.Sequences

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
'
```

child_handles()

Get child handle list

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- ***test*** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
19
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(test)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

StateBin

class cov.StateBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum (bool)** – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
20
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

Feedback

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

TransBin

class cov.TransBin

state_bin_handles()

Get state_bin handle list.

Returns: Return the handle list obtained by 1-many method: state_bin.

Return type: list of cov.StateBin

Example

```
>>> hdlList = ref.state_bin_handles()
>>> hdlList[0].type()
'npiCovStateBin'
```

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
21
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Return type: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

Feedback

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns:Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

SeqBin

class cov.SeqBin

state_bin_handles()

Get state_bin handle list

Returns: Return the handle list obtained by 1-many method: state_bin

Return type: list of cov.StateBin

Example

```
>>> hdlList = ref.state_bin_handles()
>>> hdlList[0].type()
'npiCovStateBin'
```

scope_handle()

Get scope handle

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
22
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- ***test*** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

ConditionMetric

class cov.ConditionMetric

instance_handle()

Get instance handle.

Returns: Return the handle obtained by 1-1 method: instance.

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovLineMetric'
>>> hdl.type(test, True)
7
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Condition

class cov.Condition

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle.

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
23
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Feedback

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

is_event_condition(*test=None*)

Get is event condition

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.is_event_condition(test)
1
```

name()

Get name

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

ConditionBin

class cov. ConditionBin

condition_term_handles()

Get condition_term handle list.

Returns: Return the handle list obtained by 1-many method: condition_term.

Return type: list of cov.ConditionTerm.

Example

```
>>> hdlList = ref.condition_term_handles()
>>> hdlList[0].type()
'npiCovConditionTerm'
```

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle.

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns:

Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovStmtBin'
>>> hdl.type(test, True)
24
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set.

Returns:Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set.

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

ConditionTerm

class cov.ConditionTerm

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovStmtBin'
>>> hdl.type(test, True)
25
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Branch Metric

class cov.BranchMetric

instance_handle()

Get instance handle.

Returns: Return the handle obtained by 1-1 method: instance.

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
8
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Branch

class cov.Branch

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
```

```
>>> hdl.type(test, True)
26
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

BranchBin

class cov.BranchBin

branch_term_handles()

Get branch_term handle list.

Returns: Return the handle list obtained by 1-many method: branch_term.

Return type: list of cov.BranchTerm

Example

```
>>> hdlList = ref.branch_term_handles()
>>> hdlList[0].type()
'npiCovBranchTerm'
```

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
27
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

set_covered(*test, value*)

Set covered to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_covered(test, 0)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

BranchTerm

class cov.BranchTerm

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
28
```

value(*test=None, is_get_enum=False*)

Get value

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.value(test)
'1'
>>> hdl.value(test, True)
1
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

AssertMetric

class cov.AssertMetric

instance_handle()

Get instance handle.

Returns: Return the handle obtained by 1-1 method: instance.

Return type: cov.Instance

Example

```
>>> hdl = ref.instance_handle()
>>> hdl.type()
'npiCovInstance'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
9
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

full_name()

Get full name.

Returns: Return the name string by property: full_name

Return type: str

Example

```
>>> hdl.full_name()
'top.a.b.c'
```

Assert

class cov .Assert

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
```

```
>>> hdl.type(test, True)
29
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to objec

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

severity(*test=None*)

Get severity.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.severity(test)
1
```

category(*test=None*)

Get category.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.category(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: full_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

CoverProperty

class cov.CoverProperty

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
44
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Feedback

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

severity(*test=None*)

Get severity.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.severity(test)
1
```

category(*test=None*)

Get category

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.category(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

CoverSequence

class cov.CoverSequence

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
45
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set.

Returns: Return 1 if success; Otherwise, return 0.

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

severity(*test=None*)

Get severity.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.severity(test)
1
```

category(*test=None*)

Get category.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.category(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name.

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

SuccessBin

class cov. SuccessBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
30
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

AttemptBin

class cov. AttemptBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
31
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

FailureBin

class cov. FailureBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
32
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

VacuousBin

class cov.VacuousBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
33
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

IncompleteBin

class cov.IncompleteBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
34
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

FirstmatchBin

class cov.FirstmatchBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
46
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

TestbenchMetric

class cov.TestbenchMetric

test_handle()

Get test handle.

Returns: Return the handle obtained by 1-1 method: test

Return type: Test

Example

```
>>> hdl = ref.test_handle()
>>> hdl.type()
'npiCovTest'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
10
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Covergroup

class cov.Covergroup

cover_instance_handles()

Get cover_instance handle list.

Returns: Return the handle list obtained by 1-many method: cover_instance.

Return type: list of cov.CoverInstance.

Example

```
>>> hdlList = ref.cover_instance_handles()
>>> hdlList[0].type()
'npiCovCoverInstance'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle.

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
35
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
```

1

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

per_instance(*test=None*)

Get per instance.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.per_instance(test)
1
```

weight(*test=None*)

Get weight

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.weight(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

CoverInstance

class cov.CoverInstance

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
```

```
>>> hdl.type(test, True)
40
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

weight(*test=None*)

Get weight

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.weight(test)
1
```

goal(*test=None*)

Get goal

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.goal(test)
1
```

at_least(*test=None*)

Get at least

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.at_least(test)
1
```

auto_bin_max(*test=None*)

Get auto bin max

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.auto_bin_max(test)
1
```

cross_num_print_missing(*test=None*)

Get cross num print missing

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.cross_num_print_missing(test)
1
```

detect_overlap(*test=None*)

Get detect overlap

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.detect_overlap(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

comment()

Get comment

Returns: Return the name string by property: comment

Return type: str

Example

```
>>> hdl.comment()
'obj'
```

Cross

class cov.Cross

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45''
```

component_handles()

Get component handle list.

Returns: Return the handle list obtained by 1-many method: component.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.component_handles() >>> hdlList[0].line_no()
'45''
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
37
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

weight(*test=None*)

Get weight

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.weight(test)
1
```

Feedback

goal(*test=None*)

Get goal

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.goal(test)
1
```

at_least(*test=None*)

Get at least

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.at_least(test)
1
```

cross_num_print_missing(*test=None*)

Get cross num print missing

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.cross_num_print_missing(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

Feedback

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

comment()

Get comment

Returns: Return the name string by property: comment

Return type: str

Example

```
>>> hdl.comment()
'obj'
```

Coverpoint

class cov.Coverpoint

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child.

Return type: list of cov.Handle.

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
36
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

status(*test=None*)

Get status.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

>>> binHdl.status_excluded_at_compile_time(testHdl)

1

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

weight(*test=None*)

Get weight

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.weight(test)
1
```

goal(*test=None*)

Get goal

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.goal(test)
1
```

at_least(*test=None*)

Get at least

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.at_least(test)
1
```

auto_bin_max(*test=None*)

Get auto bin max

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.auto_bin_max(test)
1
```

detect_overlap(*test=None*)

Get detect overlap

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.detect_overlap(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

comment()

Get comment

Returns: Return the name string by property: comment

Return type: str

Example

```
>>> hdl.comment()
'obj'
```

CoverBin

class cov.CoverBin

covered_test_handles(*test*)

Get covered_test handle list.

Parameters: **test** (Test) – The test handle

Returns: Return the handle list obtained by 1-many method: covered_test.

Return type: list of cov.CoveredTest.

Example

```
>>> hdlList = ref.covered_test_handles(test)
>>> hdlList[0].type()
'npiCovCoveredTest'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope.

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
38
```

coverable(*test=None*)

Get coverable.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.coverable(test)
1
```

covered(*test=None*)

Get covered.

Parameters: **test** (Test) – The test handle

Returns: Return the int property.

Return type: int

Example

```
>>> hdl.covered(test)
1
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.count(test)
1
```

count_goal(*test=None*)

Get count goal.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.count_goal(test)
1
```

status(*test=None*)

Get status.

Feedback

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.status(test)
1
```

has_status_unreachable(*test*)

Check if status unreachable is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

set_status_unreachable(*test, value*)

Set status unreachable to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_unreachable(testHdl)
1
```

has_status_illegal(*test*)

Check if status illegal is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

set_status_illegal(*test, value*)

Set status illegal to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_illegal(testHdl)
1
```

has_status_excluded_at_compile_time(*test*)

Check if status excluded at compile time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

set_status_excluded_at_compile_time(*test, value*)

Set status excluded at compile time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_compile_time(testHdl)
1
```

has_status_covered(*test*)

Check if status covered is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

set_status_covered(*test, value*)

Set status covered to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_covered(testHdl)
1
```

has_status_proven(*test*)

Check if status proven is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

set_status_proven(*test, value*)

Set status proven to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_proven(testHdl)
1
```

has_status_excluded_at_report_time(*test*)

Check if status excluded at report time is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

set_status_excluded_at_report_time(*test, value*)

Set status excluded at report time to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded_at_report_time(testHdl)
1
```

has_status_attempted(*test*)

Check if status attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

set_status_attempted(*test, value*)

Set status attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_attempted(testHdl)
1
```

has_status_excluded(*test*)

Check if status excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

set_status_excluded(*test, value*)

Set status excluded to object

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_excluded(testHdl)
1
```

has_status_partially_excluded(*test*)

Check if status partially excluded is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

set_status_partially_excluded(*test, value*)

Set status partially excluded to object.

Feedback

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_excluded(testHdl)
1
```

has_status_partially_attempted(*test*)

Check if status partially attempted is set.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if the object does not have the status; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status_partially_attempted(*test, value*)

Set status partially attempted to object.

Parameters: **test** (Test) – The test handle

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> binHdl.status_partially_attempted(testHdl)
1
```

set_status(*test, value*)

Set status to object.

Parameters:

- **test** (Test) – The test handle
- **value** (int) – The value to set

Returns: Return 1 if success; Otherwise, return 0

Return type: int

Example

```
>>> hdl.set_status(test, 0)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

PowerDomain

class cov.PowerDomain

test_handle()

Get test handle.

Returns: Return the handle obtained by 1-1 method: test

Return type: Test

Example

```
>>> hdl = ref.test_handle()
>>> hdl.type()
'npiCovTest'
```

power_domain_handles()

Get power_domain handle list.

Returns: Return the handle list obtained by 1-many method: power_domain

Return type: list of cov.PowerDomain

Example

```
>>> hdlList = ref.power_domain_handles()
>>> hdlList[0].type()
'npiCovPowerDomain'
```

Feedback

power_measure handle()

Get power_measure handle list.

Returns: Return the handle list obtained by 1-many method: power_measure.

Return type: list of cov.PowerMeasure

Example

```
>>> hdlList = ref.power_measure_handles()
>>> hdlList[0].type()
'npiCovPowerMeasure'
```

scope_handle()

Get scope handle.

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
42
```

line_no(*test=None*)

Get line no.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.line_no(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

file_name()

Get file name.

Returns: Return the name string by property: file_name

Return type: str

Example

```
>>> hdl.file_name()
'/u/john_doe/test/test.v'
```

PowerMeasure

class cov.PowerMeasure

power_metric_handle()

Get power_metric handle

Returns: Return the handle obtained by 1-1 method: power_metric

Return type: cov.PowerMetric

Example

Feedback

```
>>> hdl = ref.power_metric_handle()
>>> hdl.type()
'npiCovPowerMetric'
```

scope_handle()

Get scope handle

Returns: Return the handle obtained by 1-1 method: scope

Return type: cov.Handle

Example

```
>>> hdl = ref.scope_handle()
>>> hdl.line_no()
'35'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
43
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

PowerMetric

class cov . PowerMetric

test_handle()

Get test handle.

Returns: Return the handle obtained by 1-1 method: test

Return type: *Test*

Example

```
>>> hdl = ref.test_handle()
>>> hdl.type()
'npiCovTest'
```

power_measure_handle()

Get power_measure handle.

Returns: Return the handle obtained by 1-1 method: power_measure

Return type: cov.PowerMeasure

Example

```
>>> hdl = ref.power_measure_handle()
>>> hdl.type()
'npiCovPowerMeasure'
```

child_handles()

Get child handle list.

Returns: Return the handle list obtained by 1-many method: child

Return type: list of cov.Handle

Example

```
>>> hdlList = ref.child_handles() >>> hdlList[0].line_no()
'45'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovLineMetric'
>>> hdl.type(test, True)
41
```

name()

Get name.

Returns: Return the name string by property: name.

Return type: str

Example

```
>>> hdl.name()
'obj'
```

Covered Test

***class* cov.CoveredTest**

test_handle()

Get test handle.

Returns: Return the handle obtained by 1-1 method: test

Return type: Test

Example

```
>>> hdl = ref.test_handle()
>>> hdl.type()
'npiCovTest'
```

type(*test=None, is_get_enum=False*)

Get type.

Parameters:

- **test** (Test) – The test handle
- **is_get_enum** (bool) – Flag for the data type of return value.

Returns: Return the string property if is_get_enum is False; Otherwise, return the int property if is_get_enum is True.

Return type: str or int

Example

```
>>> hdl.type(test)
'npiCovBlock'
>>> hdl.type(test, True)
47
```

count(*test=None*)

Get count.

Parameters: **test** (Test) – The test handle

Returns: Return the int property

Return type: int

Example

```
>>> hdl.count(test)
1
```

name()

Get name.

Returns: Return the name string by property: name

Return type: str

Example

```
>>> hdl.name()
'obj'
```

5

L0 APIs

Following are the L0 APIs in Coverage module:

API	Description
cov.open(vdb_dir)	Open coverage vdb to get database handle.
cov.merge_test(dst_test, src_test, map_file=None)	Merge test handles.

cov.open(vdb_dir)

open coverage vdb to get database handle.

Parameters: **vbb_dir** (str) – the vdb name. For example, simv.vdb.

Returns: Return the database handle according to the vdb name.

Return type: Database

Example

```
>>> db = cov.open('simv.vdb')
>>> db.name()
'simv.vdb'
```

cov.merge_test(*dst_test, src_test, map_file=None*)

Merge test handles.

Parameters:

- **dst_test**(Test) – the destination test.
- **src_test**(Test) – the source test.
- **map_file** (**str**) – the mapping file name for merging tests (optional).

Returns: Return the merged test.

Return type: Test

Example

```
>>> merged_test = cov.merge_test(dst_test, src_test, 'mapping.file')
>>> merged_test.type()
'npiCovTest'
```