

# Contents

<b>1</b>	<b>Shallot Scripting Documentation</b>	<b>1</b>
1.1	Scripting Manual	1
1.1.1	General Notes	1
1.1.1.1	Plugin Management Assistant	1
1.1.1.2	General Plugin Design	2
1.1.1.3	Implement & Register	2
1.1.1.4	Position Indexes	2
1.1.1.5	The '_ApiProxy' Class	3
1.1.1.6	Exceptions	3
1.1.1.7	Localization	3
1.1.1.8	Deployment	3
1.1.2	Feature Overview	3
1.1.2.1	Actions	4
1.1.2.2	User Feedback	5
1.1.2.3	Configuration Values	5
1.1.2.4	Settings	6
1.1.2.5	Main Window	6
1.1.2.6	EURLs	6
1.1.2.7	Filesystem	7
1.1.2.8	Detail Columns	7
1.1.2.9	Operations	8
1.1.2.10	Panel Details	9
1.1.2.11	File Property Dialog	9

1.1.2.12	File Searches	10
1.1.2.13	Thumbnails	10
1.1.2.14	Bookmarks	11
1.1.2.15	Logging	11
1.1.2.16	Utilities	11
1.2	Packages	11
1.3	Class Hierarchy	12
1.4	Class List	13
1.5	shallot Namespace Reference	16
1.5.1	Detailed Description	17
1.6	shallot._ApiProxy Class Reference	17
1.6.1	Detailed Description	18
1.7	shallot.Actions.AbstractAction Class Reference	18
1.7.1	Detailed Description	19
1.7.2	Member Function Documentation	19
1.7.2.1	initialize()	19
1.7.2.2	set_enabled()	19
1.7.2.3	set_visible()	20
1.8	shallot.PanelDetails.AbstractPanelDetailValueElement Class Reference	20
1.8.1	Detailed Description	20
1.9	shallot.Actions.ActionAction Class Reference	21
1.9.1	Detailed Description	21
1.9.2	Constructor & Destructor Documentation	21
1.9.2.1	__init__()	21
1.9.3	Member Function Documentation	22
1.9.3.1	action()	22
1.9.3.2	initialize()	22
1.9.3.3	initialize_sync()	23
1.9.3.4	set_enabled()	23
1.9.3.5	set_visible()	23

1.10 shallot.Actions Class Reference . . . . .	23
1.10.1 Detailed Description . . . . .	24
1.10.2 Member Function Documentation . . . . .	25
1.10.2.1 register() . . . . .	25
1.11 shallot.Exceptions.ArgumentException Class Reference . . . . .	25
1.11.1 Detailed Description . . . . .	26
1.11.2 Member Function Documentation . . . . .	26
1.11.2.1 isExceptionClass() . . . . .	26
1.12 shallot.Streaming.BlobReadDataDevice Class Reference . . . . .	26
1.12.1 Detailed Description . . . . .	27
1.12.2 Constructor & Destructor Documentation . . . . .	27
1.12.2.1 __init__() . . . . .	27
1.13 shallot.Bookmarks.Bookmark Class Reference . . . . .	27
1.13.1 Detailed Description . . . . .	28
1.13.2 Member Function Documentation . . . . .	28
1.13.2.1 id() . . . . .	28
1.13.2.2 tags() . . . . .	28
1.14 shallot.Bookmarks Class Reference . . . . .	29
1.14.1 Detailed Description . . . . .	29
1.14.2 Member Function Documentation . . . . .	29
1.14.2.1 add_bookmark() . . . . .	29
1.14.2.2 change_bookmark() . . . . .	30
1.14.2.3 change_bookmark_tags() . . . . .	30
1.14.2.4 move_bookmark_down() . . . . .	30
1.14.2.5 move_bookmark_to_folder() . . . . .	32
1.14.2.6 move_bookmark_up() . . . . .	32
1.14.2.7 remove_bookmark() . . . . .	32
1.15 shallot.Actions.ByRegExpPredicate Class Reference . . . . .	33
1.15.1 Detailed Description . . . . .	33
1.16 shallot.ConfigurationValue.Category Class Reference . . . . .	33

1.16.1 Detailed Description . . . . .	33
1.16.2 Member Data Documentation . . . . .	34
1.16.2.1 Behavior . . . . .	34
1.16.2.2 ExternalTools . . . . .	34
1.16.2.3 GUI . . . . .	34
1.17 shallot.ConfigurationValue Class Reference . . . . .	34
1.17.1 Detailed Description . . . . .	35
1.17.2 Constructor & Destructor Documentation . . . . .	35
1.17.2.1 __init__() . . . . .	35
1.17.3 Member Function Documentation . . . . .	35
1.17.3.1 set_value() . . . . .	36
1.18 shallot.Operations.FilesystemOperationStep.ConflictResolution Class Reference . . . . .	36
1.18.1 Detailed Description . . . . .	36
1.18.2 Member Data Documentation . . . . .	36
1.18.2.1 Indirect . . . . .	37
1.18.2.2 MergeDirectories . . . . .	37
1.18.2.3 OverwriteDestination . . . . .	37
1.18.2.4 RenameDestinationBefore . . . . .	37
1.18.2.5 Skip . . . . .	37
1.18.2.6 Unresolved . . . . .	37
1.18.2.7 UseDifferentDestinationName . . . . .	38
1.19 shallot.Actions.DefaultPrecedenceValues Class Reference . . . . .	38
1.19.1 Detailed Description . . . . .	38
1.19.2 Member Data Documentation . . . . .	38
1.19.2.1 BuiltinSpecialOpen . . . . .	38
1.19.2.2 OpenFile . . . . .	38
1.20 shallot.DetailColumn Class Reference . . . . .	39
1.20.1 Detailed Description . . . . .	39
1.20.2 Constructor & Destructor Documentation . . . . .	40
1.20.2.1 __init__() . . . . .	40

1.20.3	Member Function Documentation	40
1.20.3.1	apply_value()	40
1.20.3.2	compute_value()	41
1.20.3.3	determine_value()	41
1.20.3.4	find_by_name()	41
1.20.3.5	register_as_transferrable()	42
1.20.4	Member Data Documentation	42
1.20.4.1	INDEX_CORE	42
1.20.4.2	INDEX_EXOTIC	42
1.20.4.3	INDEX_INTERESTING	43
1.21	shallot.Actions.DontResolveLinksPredicate Class Reference	43
1.21.1	Detailed Description	43
1.22	shallot.Environment Class Reference	43
1.22.1	Detailed Description	44
1.22.2	Member Data Documentation	44
1.22.2.1	is_debug_build	44
1.22.2.2	shallot_builtinplugin_dir	44
1.22.2.3	shallot_data_dir	44
1.22.2.4	shallot_language	45
1.22.2.5	shallot_systemplugin_dir	45
1.22.2.6	shallot_userplugin_dir	45
1.23	shallot.Eurl Class Reference	45
1.23.1	Detailed Description	46
1.23.2	Member Function Documentation	46
1.23.2.1	as_string()	46
1.23.2.2	basename()	47
1.23.2.3	create()	47
1.23.2.4	enwrap_with_outer_url()	47
1.23.2.5	from_string()	48
1.23.2.6	has_inner_urls()	48

1.23.2.7	<a href="#">has_parent_segment()</a>	48
1.23.2.8	<a href="#">hostname()</a>	48
1.23.2.9	<a href="#">is_prefix_of()</a>	48
1.23.2.10	<a href="#">outer_url()</a>	49
1.23.2.11	<a href="#">outer_url_is_root_directory()</a>	49
1.23.2.12	<a href="#">outermost_inner_eurl()</a>	49
1.23.2.13	<a href="#">parent_segment()</a>	49
1.23.2.14	<a href="#">path()</a>	50
1.23.2.15	<a href="#">root()</a>	50
1.23.2.16	<a href="#">scheme()</a>	50
1.23.2.17	<a href="#">with_appended_segment()</a>	50
1.23.2.18	<a href="#">with_appended_segments()</a>	50
1.24	<a href="#">shallot.Exceptions Class Reference</a>	51
1.24.1	<a href="#">Detailed Description</a>	51
1.25	<a href="#">shallot.Actions.ExecutionInfo Class Reference</a>	51
1.25.1	<a href="#">Detailed Description</a>	53
1.25.2	<a href="#">Member Function Documentation</a>	53
1.25.2.1	<a href="#">add_changed_eurl()</a>	53
1.25.2.2	<a href="#">respect_cancel()</a>	53
1.25.2.3	<a href="#">set_details()</a>	53
1.25.2.4	<a href="#">set_progress()</a>	54
1.25.2.5	<a href="#">set_progress_indeterminate()</a>	54
1.26	<a href="#">shallot.Actions.ExecutionUserFeedback Class Reference</a>	54
1.26.1	<a href="#">Detailed Description</a>	55
1.26.2	<a href="#">Member Function Documentation</a>	55
1.26.2.1	<a href="#">messagebox()</a>	55
1.26.2.2	<a href="#">simple_inputbox()</a>	55
1.26.2.3	<a href="#">simple_messagebox()</a>	56
1.27	<a href="#">shallot.FilePropertyDialog Class Reference</a>	56
1.27.1	<a href="#">Detailed Description</a>	57

1.28 shallot.FileSearch Class Reference . . . . .	57
1.28.1 Detailed Description . . . . .	57
1.29 shallot.Filesystem Class Reference . . . . .	57
1.29.1 Detailed Description . . . . .	58
1.29.2 Member Function Documentation . . . . .	58
1.29.2.1 create_node() . . . . .	58
1.29.2.2 find_nodes_for_eurl() . . . . .	59
1.29.2.3 get_or_create_node() . . . . .	59
1.29.2.4 refresh() . . . . .	59
1.29.2.5 try_get_nodes_for_eurl() . . . . .	61
1.29.3 Member Data Documentation . . . . .	61
1.29.3.1 rootnode . . . . .	61
1.30 shallot.Operations.FilesystemOperation Class Reference . . . . .	61
1.30.1 Detailed Description . . . . .	63
1.30.2 Member Function Documentation . . . . .	63
1.30.2.1 can_copy_item() . . . . .	63
1.30.2.2 can_create_directory() . . . . .	63
1.30.2.3 can_create_file() . . . . .	64
1.30.2.4 can_create_link() . . . . .	64
1.30.2.5 can_delete_item() . . . . .	64
1.30.2.6 can_get_filecontent() . . . . .	65
1.30.2.7 can_move_item() . . . . .	65
1.30.2.8 copy_item() . . . . .	65
1.30.2.9 copy_items() . . . . .	66
1.30.2.10 create_directory() . . . . .	66
1.30.2.11 create_file() . . . . .	66
1.30.2.12 create_link() . . . . .	67
1.30.2.13 delete_directory_if_empty() . . . . .	67
1.30.2.14 delete_item() . . . . .	68
1.30.2.15 get_extendedattribute() . . . . .	68

1.30.2.16 <code>get_extendedattribute_size()</code> . . . . .	68
1.30.2.17 <code>get_file_content()</code> . . . . .	69
1.30.2.18 <code>get_linktarget()</code> . . . . .	69
1.30.2.19 <code>get_type()</code> . . . . .	69
1.30.2.20 <code>itemlist()</code> . . . . .	69
1.30.2.21 <code>itemlist_by_type()</code> . . . . .	70
1.30.2.22 <code>list_extendedattributes()</code> . . . . .	70
1.30.2.23 <code>move_item()</code> . . . . .	70
1.30.2.24 <code>move_items()</code> . . . . .	71
1.30.2.25 <code>remove_extendedattribute()</code> . . . . .	71
1.30.2.26 <code>resolve_eurl_link()</code> . . . . .	71
1.30.2.27 <code>resolve_eurl_link_nonrecursive()</code> . . . . .	72
1.30.2.28 <code>resolve_node_link()</code> . . . . .	72
1.30.2.29 <code>resolve_node_link_nonrecursive()</code> . . . . .	72
1.30.2.30 <code>set_extendedattribute()</code> . . . . .	73
1.31 <code>shallot.Operations.FilesystemOperationProgressMonitor</code> Class Reference . . . . .	73
1.31.1 Detailed Description . . . . .	74
1.31.2 Constructor & Destructor Documentation . . . . .	74
1.31.2.1 <code>__init__()</code> . . . . .	74
1.31.3 Member Function Documentation . . . . .	74
1.31.3.1 <code>changed()</code> . . . . .	74
1.31.3.2 <code>resolve_conflicts()</code> . . . . .	75
1.32 <code>shallot.Operations.FilesystemOperationStep</code> Class Reference . . . . .	75
1.32.1 Detailed Description . . . . .	76
1.33 <code>shallot.Setting.GroupInfo</code> Class Reference . . . . .	76
1.33.1 Detailed Description . . . . .	77
1.33.2 Member Data Documentation . . . . .	77
1.33.2.1 Behavior . . . . .	77
1.33.2.2 <code>DataView</code> . . . . .	77
1.33.2.3 <code>FileHandling</code> . . . . .	77



1.33.2.4	Global	77
1.33.2.5	GUI	77
1.33.2.6	Special	78
1.34	shallot.Filesystem.Handler Class Reference	78
1.34.1	Detailed Description	79
1.34.2	Member Function Documentation	79
1.34.2.1	can_create_directory()	80
1.34.2.2	can_create_file()	80
1.34.2.3	can_create_link()	80
1.34.2.4	can_delete_item()	81
1.34.2.5	can_get_filecontent()	81
1.34.2.6	can_rename_item()	82
1.34.2.7	configure_item()	82
1.34.2.8	create_directory()	82
1.34.2.9	create_file()	84
1.34.2.10	create_link()	84
1.34.2.11	delete_item()	85
1.34.2.12	get_actions()	85
1.34.2.13	get_customattributes()	85
1.34.2.14	get_extendedattribute()	86
1.34.2.15	get_extendedattribute_size()	86
1.34.2.16	get_file_content()	87
1.34.2.17	get_linktarget()	87
1.34.2.18	get_mimetype()	88
1.34.2.19	get_mtime()	88
1.34.2.20	get_size()	88
1.34.2.21	get_type()	89
1.34.2.22	itemlist()	89
1.34.2.23	list_extendedattributes()	90
1.34.2.24	register()	90

1.34.2.25	<code>remove_extendedattribute()</code>	90
1.34.2.26	<code>rename_item()</code>	91
1.34.2.27	<code>set_customattribute()</code>	91
1.34.2.28	<code>set_extendedattribute()</code>	92
1.34.2.29	<code>set_mtime()</code>	92
1.35	<code>shallot.Operations.HandlerTransfer Class Reference</code>	93
1.35.1	Detailed Description	93
1.35.2	Member Function Documentation	93
1.35.2.1	<code>increment_transferred_bytes()</code>	93
1.36	<code>shallot.Actions.HideOnCurrentDirectoryLevelPredicate Class Reference</code>	94
1.36.1	Detailed Description	94
1.37	<code>shallot.Actions.HideOnSelectionLevelPredicate Class Reference</code>	94
1.37.1	Detailed Description	94
1.38	<code>shallot.IntlStringMap.IntlString Class Reference</code>	95
1.38.1	Detailed Description	95
1.38.2	Constructor & Destructor Documentation	95
1.38.2.1	<code>__init__()</code>	95
1.39	<code>shallot.IntlStringMap Class Reference</code>	95
1.39.1	Detailed Description	96
1.39.2	Constructor & Destructor Documentation	96
1.39.2.1	<code>__init__()</code>	96
1.39.3	Member Data Documentation	96
1.39.3.1	<code>map</code>	96
1.40	<code>shallot.Exceptions.IOException Class Reference</code>	97
1.40.1	Detailed Description	97
1.40.2	Member Function Documentation	97
1.40.2.1	<code>isExceptionClass()</code>	97
1.41	<code>shallot.Actions.KeyShortcutPredicate Class Reference</code>	98
1.41.1	Detailed Description	98
1.42	<code>shallot.Logging Class Reference</code>	98

1.42.1 Detailed Description . . . . .	99
1.42.2 Member Function Documentation . . . . .	99
1.42.2.1 log_debug() . . . . .	99
1.42.2.2 log_error() . . . . .	99
1.42.2.3 log_info() . . . . .	99
1.42.2.4 log_warning() . . . . .	100
1.43 shallot.MainWindow Class Reference . . . . .	100
1.43.1 Detailed Description . . . . .	100
1.43.2 Member Function Documentation . . . . .	100
1.43.2.1 jump_to_eurl() . . . . .	100
1.43.2.2 open_items() . . . . .	101
1.44 shallot.Actions.ExecutionUserFeedback.MessageBoxButton Class Reference . . . . .	101
1.44.1 Detailed Description . . . . .	101
1.44.2 Member Data Documentation . . . . .	102
1.44.2.1 Cancel . . . . .	102
1.44.2.2 Continue . . . . .	102
1.44.2.3 No . . . . .	102
1.44.2.4 OK . . . . .	102
1.44.2.5 Retry . . . . .	102
1.44.2.6 Yes . . . . .	103
1.45 shallot.PanelDetails.MultiSelectionPanelDetail Class Reference . . . . .	103
1.45.1 Detailed Description . . . . .	104
1.45.2 Constructor & Destructor Documentation . . . . .	104
1.45.2.1 __init__() . . . . .	104
1.45.3 Member Function Documentation . . . . .	104
1.45.3.1 link_triggered() . . . . .	104
1.45.3.2 register() . . . . .	105
1.45.3.3 set_value() . . . . .	105
1.45.4 Member Data Documentation . . . . .	105
1.45.4.1 INDEX_EXOTIC . . . . .	105

1.45.4.2	INDEX_INTERESTING	106
1.45.4.3	INDEX_VERYINTERESTING	106
1.46	shallot.Filesystem.Node Class Reference	106
1.46.1	Detailed Description	106
1.46.2	Member Function Documentation	107
1.46.2.1	add_detail()	107
1.47	shallot.Filesystem.NodeList Class Reference	107
1.47.1	Detailed Description	107
1.47.2	Member Function Documentation	108
1.47.2.1	add_item()	108
1.47.2.2	begin_iterative_adding()	108
1.47.2.3	end_iterative_adding()	108
1.47.2.4	set_items()	108
1.48	shallot.Filesystem.NodeType Class Reference	109
1.48.1	Detailed Description	109
1.48.2	Member Data Documentation	109
1.48.2.1	Directory	109
1.48.2.2	File	110
1.48.2.3	FIRSTTYPE	110
1.48.2.4	LASTPHYSICALTYPE	110
1.48.2.5	Link	110
1.48.2.6	NONE	110
1.48.2.7	Unknown	110
1.49	shallot.Actions.OnDirectoriesPredicate Class Reference	111
1.49.1	Detailed Description	111
1.50	shallot.Actions.OnFilesPredicate Class Reference	111
1.50.1	Detailed Description	112
1.51	shallot.Actions.OnLinksPredicate Class Reference	112
1.51.1	Detailed Description	112
1.52	shallot.Actions.OnSingleEntrySelectionPredicate Class Reference	112

1.52.1 Detailed Description . . . . .	113
1.53 shallot.Operations.Operation Class Reference . . . . .	113
1.53.1 Detailed Description . . . . .	114
1.53.2 Member Function Documentation . . . . .	114
1.53.2.1 fetch_container_file() . . . . .	114
1.53.2.2 fetch_file() . . . . .	114
1.53.2.3 get_custom_data() . . . . .	114
1.53.2.4 get_detail_from_cache() . . . . .	115
1.53.2.5 set_custom_data() . . . . .	115
1.53.2.6 store_detail_in_cache() . . . . .	115
1.54 shallot.Operations Class Reference . . . . .	116
1.54.1 Detailed Description . . . . .	116
1.55 shallot.PanelDetails.PanelDetail Class Reference . . . . .	117
1.55.1 Detailed Description . . . . .	117
1.55.2 Member Function Documentation . . . . .	117
1.55.2.1 register() . . . . .	117
1.55.3 Member Data Documentation . . . . .	118
1.55.3.1 INDEX_EXOTIC . . . . .	118
1.55.3.2 INDEX_INTERESTING . . . . .	118
1.55.3.3 INDEX_VERYINTERESTING . . . . .	118
1.56 shallot.PanelDetails.PanelDetailInst Class Reference . . . . .	118
1.56.1 Detailed Description . . . . .	119
1.56.2 Member Function Documentation . . . . .	119
1.56.2.1 set_row() . . . . .	119
1.57 shallot.PanelDetails Class Reference . . . . .	119
1.57.1 Detailed Description . . . . .	120
1.58 shallot.PanelDetails.PanelDetailValueElementButton Class Reference . . . . .	120
1.58.1 Detailed Description . . . . .	121
1.58.2 Constructor & Destructor Documentation . . . . .	121
1.58.2.1 __init__() . . . . .	121

1.59 shallot.PanelDetails.PanelDetailValueElementIcon Class Reference . . . . .	121
1.59.1 Detailed Description . . . . .	122
1.59.2 Constructor & Destructor Documentation . . . . .	122
1.59.2.1 __init__() . . . . .	122
1.60 shallot.PanelDetails.PanelDetailValueElementString Class Reference . . . . .	122
1.60.1 Detailed Description . . . . .	123
1.60.2 Constructor & Destructor Documentation . . . . .	123
1.60.2.1 __init__() . . . . .	123
1.61 shallot.PanelDetails.PanelDetailValueElementWaiting Class Reference . . . . .	123
1.61.1 Detailed Description . . . . .	124
1.62 shallot.Actions.PositionIndexPredicate Class Reference . . . . .	124
1.62.1 Detailed Description . . . . .	124
1.63 shallot.Actions.Predicate Class Reference . . . . .	125
1.63.1 Detailed Description . . . . .	125
1.64 shallot.Exceptions.ProgramException Class Reference . . . . .	125
1.64.1 Detailed Description . . . . .	126
1.64.2 Member Function Documentation . . . . .	126
1.64.2.1 isExceptionClass() . . . . .	126
1.65 shallot.FilePropertyDialog.Tab.PropertyButtonConfig Class Reference . . . . .	126
1.65.1 Detailed Description . . . . .	127
1.65.2 Constructor & Destructor Documentation . . . . .	127
1.65.2.1 __init__() . . . . .	127
1.66 shallot.FilePropertyDialog.Tab.PropertyConfig Class Reference . . . . .	127
1.66.1 Detailed Description . . . . .	128
1.66.2 Constructor & Destructor Documentation . . . . .	128
1.66.2.1 __init__() . . . . .	128
1.67 shallot.FilePropertyDialog.Tab.PropertyType Class Reference . . . . .	128
1.67.1 Detailed Description . . . . .	128
1.67.2 Member Data Documentation . . . . .	129
1.67.2.1 IconTextBanner . . . . .	129

1.67.2.2	String	129
1.67.2.3	StringMap	129
1.68	shallot.Streaming.ReadDataDevice Class Reference	129
1.68.1	Detailed Description	130
1.68.2	Member Function Documentation	130
1.68.2.1	getdata()	130
1.69	shallot.Exceptions.RuntimeException Class Reference	130
1.69.1	Detailed Description	131
1.69.2	Member Function Documentation	131
1.69.2.1	isExceptionClass()	131
1.70	shallot.Exceptions.ScriptedException Class Reference	131
1.70.1	Detailed Description	132
1.70.2	Constructor & Destructor Documentation	132
1.70.2.1	__init__()	132
1.70.3	Member Function Documentation	133
1.70.3.1	isExceptionClass()	133
1.71	shallot.FileSearch.SearchCriterion Class Reference	133
1.71.1	Detailed Description	134
1.71.2	Constructor & Destructor Documentation	134
1.71.2.1	__init__()	134
1.71.3	Member Function Documentation	134
1.71.3.1	configure()	134
1.71.3.2	match()	135
1.72	shallot.FileSearch.SearchCriterionFactory Class Reference	135
1.72.1	Detailed Description	136
1.72.2	Constructor & Destructor Documentation	136
1.72.2.1	__init__()	136
1.72.3	Member Function Documentation	136
1.72.3.1	get_searchspec_description()	136
1.72.3.2	is_visible_for()	137

1.72.3.3	<a href="#">register()</a>	137
1.72.4	<a href="#">Member Data Documentation</a>	137
1.72.4.1	<a href="#">INDEX_CORE</a>	138
1.72.4.2	<a href="#">INDEX_EXOTIC</a>	138
1.72.4.3	<a href="#">INDEX_NORMAL</a>	138
1.73	<a href="#">shallot.Setting Class Reference</a>	138
1.73.1	<a href="#">Detailed Description</a>	139
1.73.2	<a href="#">Constructor &amp; Destructor Documentation</a>	139
1.73.2.1	<a href="#">__init__()</a>	139
1.73.3	<a href="#">Member Function Documentation</a>	140
1.73.3.1	<a href="#">get_value()</a>	140
1.73.3.2	<a href="#">register()</a>	140
1.73.3.3	<a href="#">set_value()</a>	140
1.73.3.4	<a href="#">value_description()</a>	141
1.74	<a href="#">shallot.PanelDetails.SingleSelectionPanelDetail Class Reference</a>	141
1.74.1	<a href="#">Detailed Description</a>	142
1.74.2	<a href="#">Constructor &amp; Destructor Documentation</a>	142
1.74.2.1	<a href="#">__init__()</a>	142
1.74.3	<a href="#">Member Function Documentation</a>	142
1.74.3.1	<a href="#">link_triggered()</a>	143
1.74.3.2	<a href="#">register()</a>	143
1.74.3.3	<a href="#">set_value()</a>	143
1.74.4	<a href="#">Member Data Documentation</a>	144
1.74.4.1	<a href="#">INDEX_EXOTIC</a>	144
1.74.4.2	<a href="#">INDEX_INTERESTING</a>	144
1.74.4.3	<a href="#">INDEX_VERYINTERESTING</a>	144
1.75	<a href="#">shallot.Streaming Class Reference</a>	144
1.75.1	<a href="#">Detailed Description</a>	145
1.76	<a href="#">shallot.Streaming.StreamReadDataDevice Class Reference</a>	145
1.76.1	<a href="#">Detailed Description</a>	146



1.76.2	Constructor & Destructor Documentation	146
1.76.2.1	__init__()	146
1.77	shallot.Actions.SubmenuItem Class Reference	146
1.77.1	Detailed Description	147
1.77.2	Constructor & Destructor Documentation	147
1.77.2.1	__init__()	147
1.77.3	Member Function Documentation	147
1.77.3.1	initialize()	147
1.77.3.2	set_enabled()	147
1.77.3.3	set_subitems()	148
1.77.3.4	set_visible()	148
1.78	shallot.FilePropertyDialog.Tab Class Reference	148
1.78.1	Detailed Description	150
1.78.2	Constructor & Destructor Documentation	150
1.78.2.1	__init__()	150
1.78.3	Member Function Documentation	150
1.78.3.1	register()	150
1.78.3.2	selected_index_for_property()	151
1.78.3.3	update_widget()	151
1.78.4	Member Data Documentation	151
1.78.4.1	INDEX_CORE	151
1.78.4.2	INDEX_EXOTIC	151
1.78.4.3	INDEX_IMPORTANT	152
1.78.4.4	INDEX_NORMAL	152
1.78.4.5	INDEX_VERYIMPORTANT	152
1.79	shallot.FilePropertyDialog.TabPropertyIconTextBanner Class Reference	152
1.79.1	Detailed Description	153
1.79.2	Member Function Documentation	153
1.79.2.1	add_icon()	153
1.79.2.2	add_text()	153

1.80 shallot.Environment.Thread Class Reference . . . . .	153
1.80.1 Detailed Description . . . . .	154
1.80.2 Member Function Documentation . . . . .	154
1.80.2.1 execute_threaded() . . . . .	154
1.80.2.2 run() . . . . .	155
1.80.2.3 start() . . . . .	155
1.81 shallot.ThumbnailProvider Class Reference . . . . .	155
1.81.1 Detailed Description . . . . .	156
1.81.2 Member Function Documentation . . . . .	156
1.81.2.1 get_thumbnail() . . . . .	156
1.81.2.2 register() . . . . .	157
1.81.3 Member Data Documentation . . . . .	157
1.81.3.1 INDEX_CORE . . . . .	157
1.81.3.2 INDEX_EXOTIC . . . . .	157
1.81.3.3 INDEX_FALLBACK . . . . .	157
1.81.3.4 INDEX_NORMAL . . . . .	158
1.82 shallot.Environment.Timer Class Reference . . . . .	158
1.82.1 Detailed Description . . . . .	158
1.82.2 Member Function Documentation . . . . .	158
1.82.2.1 run() . . . . .	159
1.82.2.2 start() . . . . .	159
1.82.2.3 start_in_timer() . . . . .	159
<b>Index</b>	<b>161</b>

# Chapter 1

## Shallot Scripting Documentation

### 1.1 Scripting Manual

Shallot has a plugin interface which allows to add functionality to the core from outside. This document is for people who are going to create a new Shallot plugin in order to bring some new functionality to Shallot.

A Shallot plugin developer should have at least basic skills in the [Python](#) programming language and should have read the entire section [Scripting Manual](#) (From the [Feature Overview](#), it might be okay to read just the parts relevant for you).

After the [Scripting Manual](#), you find the complete interface reference. You should use it for finding some technical details (which classes exist, which methods are there, which arguments do they have, ...). It lists the structure of the interface from different perspectives. Some of them are quite useful, others ones are more technical and only interesting in rare situations. Good starting points are those sections:

- The "Class List" provides an overview of all existing classes.
- The "shallot Namespace Reference" gives a coarser overview of all existing subparts.

#### 1.1.1 General Notes

This section covers some notes and hints, which you should know about before you begin with developing Shallot plugins.

##### 1.1.1.1 Plugin Management Assistant

Shallot includes a plugin manager. It helps to work with plugins as it provides an overview of all installed plugins and some management actions you can execute on them (e.g. deleting, disabling).

It is not enabled by default, since it is not required for day-to-day use. For plugin development, you should enable the Plugin Management in the *Tuning* dialog.

It also provides a collection of sample plugins. They can either be used as some kind of a tiny step-by-step tutorial or as templates for your new plugins.

### 1.1.1.2 General Plugin Design

Shallot plugins are Python scripts in a single file with the filename `[pluginname].py` (replacing `[pluginname]` with the actual name). They must `import shallot` and use the methods provided there for interacting with the Shallot core.

### 1.1.1.3 Implement & Register

For the most kinds of functionality you can add to Shallot, the development pattern is as follows (everything happening in your Python plugin file):

- At first you subclass one of the Shallot base classes and put your functionality into the class implementation. The documentation of the base class tells which methods may and/or must be implemented in your subclass.
- Then, in the *main* section of that plugin - this is the unindented root block as it is in typical executable Python scripts - the code must add this new subclass to the Shallot runtime. This happens with one of the `register*` methods in the Shallot interface.

Which particular elements of the Shallot interface are used, depends on what kind of functionality you want to add. The section [Feature Overview](#) below describes the details.

### 1.1.1.4 Position Indexes

For some kinds of pluggable functionality, Shallot keeps an *ordered* list of plugged objects. The semantic of the order can differ, e.g. it can be used for displaying purposes or for an execution order.

The technical mechanism is similar on most places in the interface. Particular functions (i.e. some constructors and register methods) have this two arguments amongst others:

- `positionGroup`: A coarse order information. This must be set to one of the elements in a particular enumeration (see the function documentation for details).
- `positionIndex`: An integer between 0 and (at least) 10000 which controls the order within the chosen group. Lower values mean sooner placement in the order. They are not required to be continuous but can have holes and also duplicates (although the order of duplicates to each other is then undefined). Existing larger indexes will also not shift (as they would do in a typical list) on insertion of a new item.

The documentation texts of them explain what the order is used for. Those two arguments together control the placement in the ordered object list. Typically both ones are optional and have a default:

- `positionGroup`: Some fixed default group. The function documentation might mention details.
- `positionIndex`: Some random-like integer, which is derived from your class name (and so is always the same in each run on each machine).

If so, you can decide to use those defaults, or to manually set either one or both of them.

If the order is critical (e.g. because it is the execution order of a process which only succeeds in a certain order), it is probably required to manually set them. Obviously you must then find out the values from the related objects (e.g. by trial-and-error) and tune your values according to them.

If not, you are fine with the defaults in many cases.

#### 1.1.1.5 The `'_ApiProxy'` Class

Studying the reference, you will see the class `shallot._ApiProxy` be referred at many places. Whenever this is the case, e.g. as a superclass of some interface classes, this is mostly a technical information with no big impact on the developer. It is a common superclass for many classes in the Shallot scripting interface, since it gives them access to some internal parts of the infrastructure. You should not directly come in touch with any specifics of this class.

#### 1.1.1.6 Exceptions

The base class for exceptions in Shallot is `shallot.Exceptions.ScriptedException`. The `shallot.Exceptions` namespace contains more interesting stuff.

Try to only throw those exceptions (either directly or indirectly ) from your code to be failsafe. You should use one of the subclasses in `raise` statements. You should never use subclasses in `expect [Foo]` statements but only `shallot.Exceptions.ScriptedException`. Use `shallot.Exceptions.ScriptedException.isExceptionClass` for checking against a certain class (as string) and rethrow if needed. This is because a exception in Shallot typically has more classes than the direct Python side hierarchy would look like.

#### 1.1.1.7 Localization

Shallot plugins can support more than just one language. For multi-language support, you should create a global instance of `shallot.IntlStringMap` at the beginning and take strings from there when they are used for displaying:

```
import shallot

Strings = shallot.IntlStringMap(
    HelloWorld = {"en": "Hello world!", "de": "Hallo Welt!"},
    SomethingElse = {"en": "Something different", "de": "Etwas anderes"},
)

shallot.Logging.log_info(Strings.HelloWorld)
```

#### 1.1.1.8 Deployment

The deployment of a Shallot plugin is as easy as copying the plugin file. However, the destination differs among the various operating systems. Since there is a system-global location and a per-user location, there is one more degree of variation.

On the target system, you can find out the actual destination paths by means of the 'Find in filesystem' action in the [Plugin Manager](#).

Once the destination path is known, you can choose an arbitrary deployment technique - e.g. manual file copying, or using the native package manager from the operating system - for bringing the plugin file in place.

### 1.1.2 Feature Overview

This sections introduces the different kinds of functionality you can add by means of this plugin interface. Those are the technical building blocks you have to use for implementing your plugin logic.

Most of the sections begin with an abstract explanation of the mechanisms. Read ahead! It should become clear later in the text. Each section includes links to the elements in the scripting interface reference which you need to know. You should read the documentations of this elements.

### 1.1.2.1 Actions

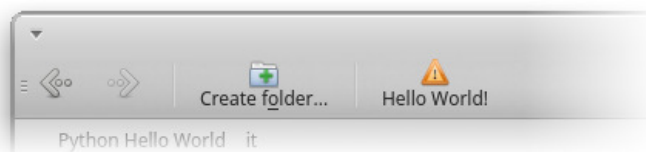
As a first approximation, an action is some piece of code for execution. Conceptually an action is similar to a simple Python function (although it technically is not just a function as you will read later). However, executing them brings a lot of additional functionality (e.g. dialog support for user communication) and infrastructure.

Actions are used in different ways in Shallot:

- The easiest thing to do with an action is to simply execute it. This is similar to directly executing the code, but using the additional functionality.
- Another common usage is to offer them in the user interface for some files or directories, so the user can decide to execute them.
- There are some other places in the plugin interface, which work with actions somehow. Read those documents for details.

The first way gives a very versatile tool. In most places in the plugin code, you can put some parts of your routine into an action and execute it in order to use the additional functionality we will see later.

The second way enables your plugin to provide some code for a given selection of files and/or directories for the user, so the user can manually trigger it. Those are typically represented in context menus or the toolbar.



At this point, our first approximation must see some corrections:

An action is a (indirect) subclass of [shallot.Actions.AbstractAction](#). It has the following subclasses, which are typical base classes for custom implementations:

- [shallot.Actions.ActionAction](#): This kind of action is executable.
- [shallot.Actions.SubmenuAction](#): A submenu is a list of other actions. It is not directly executable, but can be presented as a submenu to the user. This allows to bundle multiple [shallot.Actions.ActionAction](#) implementations which provide related functionality for better overview.

A [shallot.Actions.ActionAction](#) instance (i.e. an instance of your subclass) can be executed by initializing it at first (calling [shallot.Actions.ActionAction.initialize\\_sync](#)) and calling [shallot.Actions.ActionAction.execute](#).

For sample code, read the sources of the 'action' sample plugin (in the [Plugin Management](#)).

In order to provide your action subclass in the user interface, call [shallot.Actions.register](#).

For sample code, read the sources of the 'action.advanced' sample plugin.

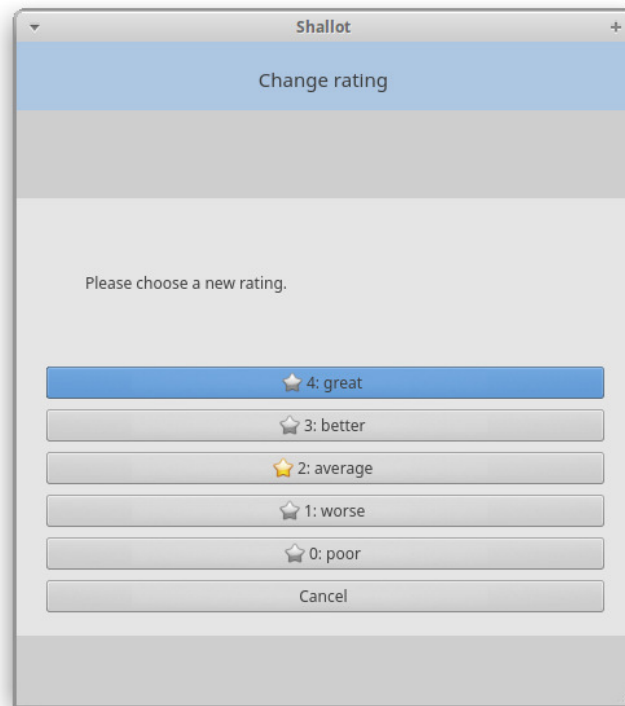
The additional functionality comes with the `info` parameter you get in your [shallot.Actions.ActionAction.action](#) implementation. It is an instance of [shallot.Actions.ExecutionInfo](#). Read this documentation for a full overview. One essential thing you can do with such an object is to have dialogs with the user. The next section shows how to do this.

The [shallot.Actions](#) namespace contains more interesting stuff.

### 1.1.2.2 User Feedback

User feedback is everything about giving some information to the user and waiting for its answer in an interactive way.

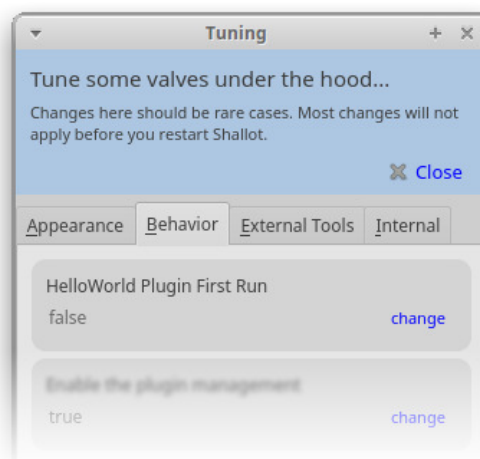
Typical situations where user feedback operations take place are [Action](#) executions. There an instance of [shallot.Actions.ExecutionInfo](#) is available. [shallot.Actions.ExecutionInfo.userfeedback](#) instance gives you access to an instance of [shallot.Actions.ExecutionUserFeedback](#), which is the key element to all kinds of feedback operations.



For sample code, read the sources of the 'userfeedback' sample plugin (in the [Plugin Management](#)).

### 1.1.2.3 Configuration Values

A plugin can use own configuration values which are stored by Shallot and are visible to the user in the *Tuning* dialog.



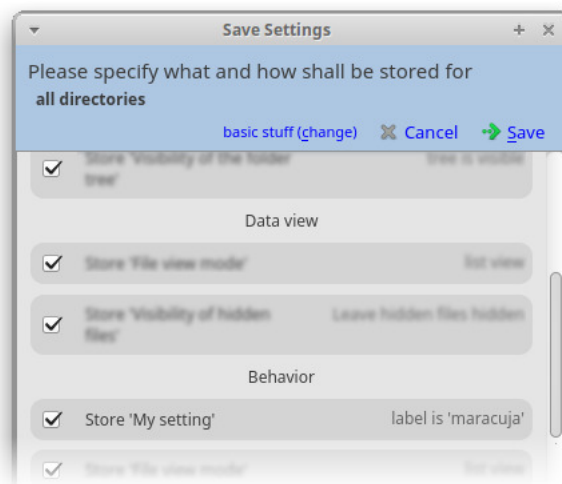
They are fine for making stuff configurable for special situations, which should typically should stay at default. For many other situations, e.g. when the user is expected to changes this value regularly, check if the [Settings features](#) fit better.

Using an own configuration value begins with subclassing [shallot.ConfigurationValue](#). Afterwards, you should create one instance of this class globally and use its methods. No registration is required.

For sample code, read the sources of the 'configurationvalue' sample plugin (in the [Plugin Management](#)).

#### 1.1.2.4 Settings

Settings are a common mechanism for various kinds of customizations the user can make in its Shallot environment. The user can store them, bind them to particular subdirectories and profiles and manage them in the *Settings* dialog.



A plugin can participate in this mechanism by adding own settings.

This begins with subclassing [shallot.Setting](#). Afterwards, [shallot.Setting.register](#) must be used for registering it.

For sample code, read the sources of the 'setting' sample plugin (in the [Plugin Management](#)).

#### 1.1.2.5 Main Window

The [shallot.MainWindow](#) class provides some navigation function and more. Read the class documentation for details. The main window object is globally available as [shallot.MainWindow.current](#).

For sample code, read the sources of the 'mainwindow' sample plugin (in the [Plugin Management](#)).

#### 1.1.2.6 EURLs

EURLs are addresses of objects in the filesystem. They are a very basic part of the Shallot foundation.

Conceptually EURLs are similar to [URLs](#). In order to express cascades of locations, EURLs are defined as a superset of URLs. As a meaningful example, this EURL refers to a file in an archive file, which in turn is located on a remote server:

```
zip:[ftp://ftpserver/foo/bar.zip]//zippedfolder/zippedfile
```

This cascades can have arbitrary depths (although this obviously doesn't make sense in arbitrary combinations).

EURLs are used in many different parts in the scripting interface. They are instances of the class [shallot.Eurl](#), which provide a rich set of getters and creator functions.

For sample code, read the sources of the 'eurl' sample plugin (in the [Plugin Management](#)).



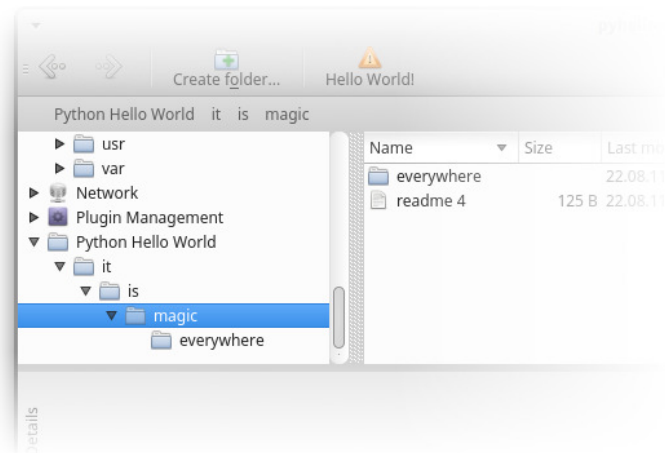
### 1.1.2.7 Filesystem

Shallot keeps an own model of the entire filesystem tree in memory (technically, it determines and stores stuff just on demand, of course). A Shallot plugin has different ways to work with this model.

At first, there are some methods for requesting some informations directly from the model. Find the static methods in [shallot.Filesystem](#) for more details (not all of them request things but some are relevant for the stuff explained later). A key class in interactions with the model, also used at various different parts in the scripting interface, is [shallot.Filesystem.Node](#).

For sample code, read the sources of the 'filesystemnode' sample plugin (in the [Plugin Management](#)).

A different way of using it filesystem model is to provide custom implementations and own nodes in the tree.



This at least requires to subclass [shallot.Filesystem.Handler](#) and registering it with [shallot.Filesystem.Handler.register](#).

In typical scenarios it also requires to create a [shallot.Filesystem.Node](#) which is associated with this handler and to insert it into the model. Some of the static methods of [shallot.Filesystem](#) are used for this procedure. Please note that it is just required to insert root nodes somewhere. The subtree of that node is specified by your handler implementation, but the node insertions (and removals) take place implicitly.

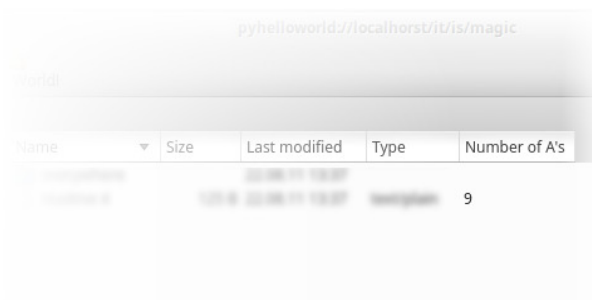
You should also read about [Operations](#), since you might have to deal with the key object of this feature here as well.

For sample code, read the sources of the 'filesystem' sample plugin (in the [Plugin Management](#)).

A filesystem handler has a wide range of possibilities for extending the Shallot functionality. One of them are custom detail columns, which are subject of the next section.

### 1.1.2.8 Detail Columns

A custom detail column, as it is visible in file views, can offer additional information to the user.



Implementing custom detail columns begins with subclassing [shallot.DetailColumn](#). You should then create one instance of this class and add it to some nodes with [shallot.Filesystem.Node.add\\_detail](#).

For sample code, read the sources of the 'filesystem' sample plugin (in the [Plugin Management](#)).

### 1.1.2.9 Operations

Operations are a concept of bundling a bunch of small changes in the filesystem together to one large step. They are also essential for working in cascaded filesystems as they are expressed with [URLs](#). Operations are entirely a backend concept without any direct representations in the user interface.

A small example explains why this bundling is a required thing: There is a ftp server which contains two archive files; a.zip (A) and b.zip (B). They build the roots of the subtrees `zip:[ftp://foo/a.zip]` and `zip:[ftp://foo/b.zip]`, each containing the contents of the respective archive. The user requests a large file copy action for 1000 files in `zip:[ftp://foo/a.zip]` to somewhere in `zip:[ftp://foo/b.zip]`. In an unbundled way, Shallot would at first fetch A, then extract the first file, then fetch B, put the file content into the archive and upload the new version of B again. Afterwards, it would run the same loop for the second file, then for the third, ... In an operation, Shallot would fetch A and B just once, then works on the local versions and just uploads the final new version of B in the end.

This bundling is realized in the [shallot.Operations.Operation](#) class. You get an instance of it in those different ways:

- In the most situations, where filesystem operations potentially make sense, there is an instance directly available as a parameter in a method of your subclasses. The documentation of the Shallot interface classes should help.
- Sometimes there is an instance available, but it is not directly available from the function parameters. In such cases, the documentation should help as well. A typical example is the execution of [Actions](#). Your implementation of [shallot.Actions.ActionAction.action](#) gets called with the parameter `info`, which is a [shallot.Actions.ExecutionInfo](#) instance. It has the member [shallot.Actions.ExecutionInfo.operation](#) which returns a operation object.
- In rare cases you have to explicitly create one. This is when you don't get an existing instance from somewhere, but you need one for some reasons. You can create a new instance with [shallot.Operations.Operation.create](#), but then it must be manually committed with [shallot.Operations.Operation.commit](#) in order to transfer all the new versions to their real destinations.

The key method for using the bundling mechanism is [shallot.Operations.Operation.fetch\\_file](#) (and [shallot.Operations.Operation.fetch\\_container\\_file](#), which is just a convenience variant with a subtle difference in the arguments). Whenever you need to read the content of a file in order to eventually replace it with a modified version, you should use it.

Please note that operations are also by the only convenient method you would even have whenever your code potentially works with cascades of filesystems. Whenever you need to get the actual data streams of the affected archives for some location like `zip:[zip:[ftp://foo/a.zip]/another.zip]/somephoto.jpeg`, calling [shallot.Operations.Operation.fetch\\_file](#) for some part of this address is an easy way (e.g. a [shallot.Filesystem.Handler](#) implementation for a custom archive format will need to do that all the time).

For sample code, read the sources of the 'operation' sample plugin (in the [Plugin Management](#)).

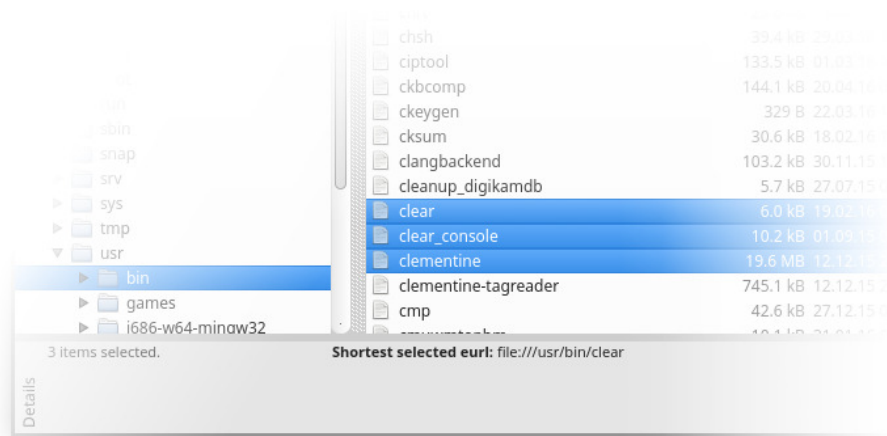
The [shallot.Operations.Operation](#) class provide some more. Amongst others, there is the [shallot.Operations.Operation.filesystem](#) member, which provides you access to an instance of [shallot.Operations.FilesystemOperation](#). It provides some primitive steps you can also execute in a bundled way. See the class documentation for details.

For sample code, read the sources of the 'filesystemoperations' and 'filesystemoperations.advanced' sample plugins.

The [shallot.Operations](#) namespace contains more interesting stuff.

### 1.1.2.10 Panel Details

Panel details can be provided by a plugin in order to provide the user with additional information about a selection of one or more files in the *Details* part of the Shallot window.



This often is used for presenting some special kinds of file metadata.

The first step to a custom panel detail is to subclass either [shallot.PanelDetails.SingleSelectionPanelDetail](#) or [shallot.PanelDetails.MultiSelectionPanelDetail](#). An instance of it must be registered with [shallot.PanelDetails.PanelDetail.register](#).

For sample code, read the sources of the 'paneldetails' sample plugin (in the [Plugin Management](#)).

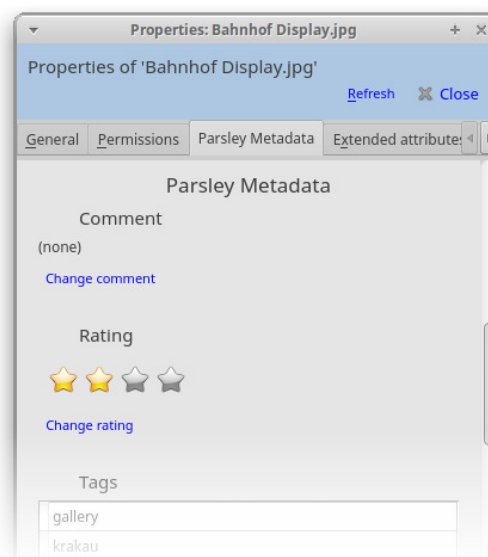
The [shallot.PanelDetails](#) namespace contains more interesting stuff.

### 1.1.2.11 File Property Dialog

The file property dialog is another place where plugins can provide custom pieces of information (and user interactions).

#### 1.1.2.11.1 Dialog Tabs

A custom tab in the property dialog can show one or more information pieces in different views. It can also offer buttons for user interaction.



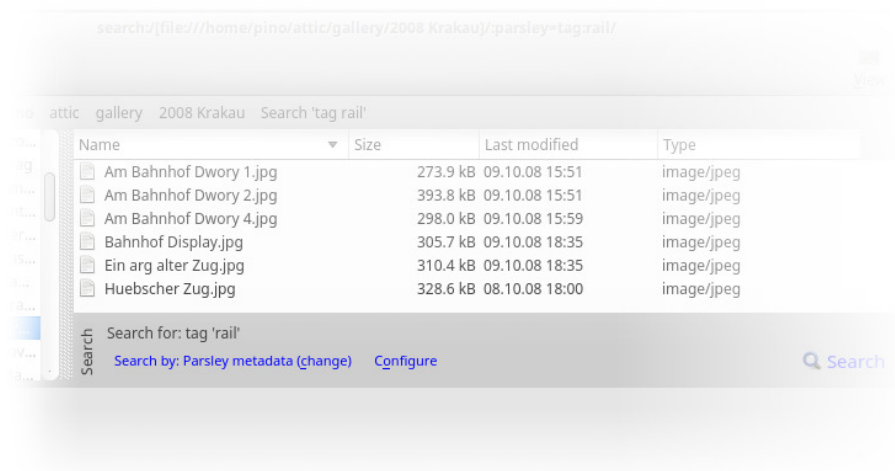
The first step is to subclass [shallot.FilePropertyDialog.Tab](#) and to register this class with [shallot.FilePropertyDialog.Tab.register](#).

For sample code, read the sources of the 'filepropertydialogtab' sample plugin (in the [Plugin Management](#)).

The [shallot.FilePropertyDialog](#) namespace contains more interesting stuff.

### 1.1.2.12 File Searches

Custom search criterion implementations bring additional functionality to Shallot file searches.



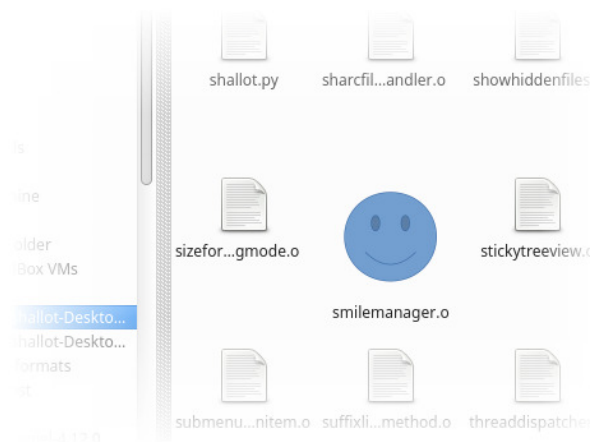
The first step is to subclass [shallot.FileSearch.SearchCriterion](#) and also [shallot.FileSearch.SearchCriterionFactory](#) (the latter one has to refer to the former one in the constructor call; see class documentations). Register the latter one with [shallot.FileSearch.SearchCriterionFactory.register](#).

For sample code, read the sources of the 'searchcriterion' sample plugin (in the [Plugin Management](#)).

The [shallot.FileSearch](#) namespace contains more interesting stuff.

### 1.1.2.13 Thumbnails

Thumbnails are visible in some view modes only. In order to support a new file format or to show certain custom thumbnail content in particular situations, a plugin can add a thumbnail provider implementation to the thumbnail creation mechanism.

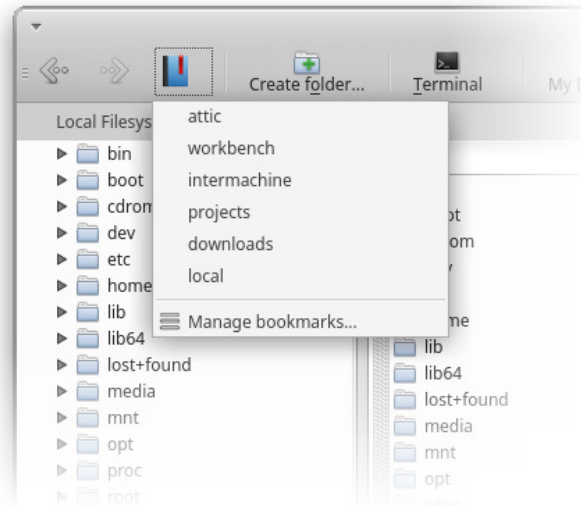


The first step is to subclass [shallot.ThumbnailProvider](#) and to register an instance of this class with [shallot.ThumbnailProvider.register](#).

For sample code, read the sources of the 'thumbnailprovider' sample plugin (in the [Plugin Management](#)).

## 1.1.2.14 Bookmarks

Bookmarks are shortcuts somewhere in the user interface, which allow the user to fastly jump to some common filesystem locations.



The bookmark menu becomes visible once some bookmarks exist. A plugin can deal with bookmarks as well. It can create and manage them with the methods in [shallot.Bookmarks](#).

For sample code, read the sources of the 'bookmarks' sample plugin (in the [Plugin Management](#)).

## 1.1.2.15 Logging

Plugins can write all kinds of information to the Shallot log in order to find potential issues. See [shallot.Logging](#) for details.

## 1.1.2.16 Utilities

There are some utilities for working with the environment available in [shallot.Environment](#). This includes [shallot.Environment.Thread](#) for running code asynchronously, [shallot.Environment.Timer](#) for running a code in regular intervals and more.

## 1.2 Packages

Here are the packages with brief descriptions (if available):

[shallot](#)

The Shallot API for Python modules . . . . . 16

## 1.3 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

shallot._ApiProxy	17
shallot.Actions.ActionAction	21
shallot.Actions.ByRegExpPredicate	33
shallot.Actions.DontResolveLinksPredicate	43
shallot.Actions.HideOnCurrentDirectoryLevelPredicate	94
shallot.Actions.HideOnSelectionLevelPredicate	94
shallot.Actions.KeyShortcutPredicate	98
shallot.Actions.OnDirectoriesPredicate	111
shallot.Actions.OnFilesPredicate	111
shallot.Actions.OnLinksPredicate	112
shallot.Actions.OnSingleEntrySelectionPredicate	112
shallot.Actions.PositionIndexPredicate	124
shallot.Actions.SubmenuAction	146
shallot.ConfigurationValue	34
shallot.DetailColumn	39
shallot.Environment.Thread	153
shallot.Environment.Timer	158
shallot.FilePropertyDialog.Tab	148
shallot.FileSearch.SearchCriterion	133
shallot.FileSearch.SearchCriterionFactory	135
shallot.Filesystem.Handler	78
shallot.Operations.FilesystemOperationProgressMonitor	73
shallot.PanelDetails.MultiSelectionPanelDetail	103
shallot.PanelDetails.SingleSelectionPanelDetail	141
shallot.Setting	138
shallot.Streaming.ReadDataDevice	129
shallot.Streaming.BlobReadDataDevice	26
shallot.Streaming.StreamReadDataDevice	145
shallot.ThumbnailProvider	155
shallot.Actions.AbstractAction	18
shallot.Actions.ActionAction	21
shallot.Actions.SubmenuAction	146
shallot.PanelDetails.AbstractPanelDetailValueElement	20
shallot.PanelDetails.PanelDetailValueElementButton	120
shallot.PanelDetails.PanelDetailValueElementIcon	121
shallot.PanelDetails.PanelDetailValueElementString	122
shallot.PanelDetails.PanelDetailValueElementWaiting	123
shallot.Actions	23
shallot.Bookmarks.Bookmark	27
shallot.Bookmarks	29
shallot.ConfigurationValue.Category	33
shallot.Operations.FilesystemOperationStep.ConflictResolution	36
shallot.Actions.DefaultPrecedenceValues	38
shallot.Environment	43
shallot.Eurl	45
Exception	
shallot.Exceptions.ScriptedException	131
shallot.Exceptions.ProgramException	125
shallot.Exceptions.ArgumentException	25
shallot.Exceptions.RuntimeException	130
shallot.Exceptions.IOException	97
shallot.Exceptions	51

shallot.Actions.ExecutionInfo	51
shallot.Actions.ExecutionUserFeedback	54
shallot.FilePropertyDialog	56
shallot.FileSearch	57
shallot.Filesystem	57
shallot.Operations.FilesystemOperation	61
shallot.Operations.FilesystemOperationStep	75
shallot.Setting.GroupInfo	76
shallot.Operations.HandlerTransfer	93
shallot.IntlStringMap.IntlString	95
shallot.IntlStringMap	95
shallot.Logging	98
shallot.MainWindow	100
shallot.Actions.ExecutionUserFeedback.MessageBoxButton	101
shallot.Filesystem.Node	106
shallot.Filesystem.NodeList	107
shallot.Filesystem.NodeType	109
shallot.Operations.Operation	113
shallot.Operations	116
shallot.PanelDetails.PanelDetail	117
shallot.PanelDetails.MultiSelectionPanelDetail	103
shallot.PanelDetails.SingleSelectionPanelDetail	141
shallot.PanelDetails.PanelDetailInst	118
shallot.PanelDetails	119
shallot.Actions.Predicate	125
shallot.Actions.ByRegExpPredicate	33
shallot.Actions.DontResolveLinksPredicate	43
shallot.Actions.HideOnCurrentDirectoryLevelPredicate	94
shallot.Actions.HideOnSelectionLevelPredicate	94
shallot.Actions.KeyShortcutPredicate	98
shallot.Actions.OnDirectoriesPredicate	111
shallot.Actions.OnFilesPredicate	111
shallot.Actions.OnLinksPredicate	112
shallot.Actions.OnSingleEntrySelectionPredicate	112
shallot.Actions.PositionIndexPredicate	124
shallot.FilePropertyDialog.Tab.PropertyButtonConfig	126
shallot.FilePropertyDialog.Tab.PropertyConfig	127
shallot.FilePropertyDialog.Tab.PropertyType	128
shallot.Streaming	144
shallot.FilePropertyDialog.TabPropertyIconTextBanner	152

## 1.4 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">shallot._ApiProxy</a>	
Common superclass for many classes in this interface	17
<a href="#">shallot.Actions.AbstractAction</a>	
Abstract base class for executable actions or submenu structures of them	18
<a href="#">shallot.PanelDetails.AbstractPanelDetailValueElement</a>	
Abstract base class for an element of a panel detail row's value	20
<a href="#">shallot.Actions.ActionAction</a>	
Abstract base class for an executable action, which can be made visible in menus or the toolbar or executed directly	21

<a href="#">shallot.Actions</a>	
Everything about <a href="#">actions</a>	23
<a href="#">shallot.Exceptions.ArgumentException</a>	
Shallot exception for failed operation due to invalid arguments given to some program part	25
<a href="#">shallot.Streaming.BlobReadDataDevice</a>	
A binary source backed by static binary content in a byte array	26
<a href="#">shallot.Bookmarks.Bookmark</a>	
A bookmark	27
<a href="#">shallot.Bookmarks</a>	
Everything about <a href="#">Bookmarks</a>	29
<a href="#">shallot.Actions.ByRegExpPredicate</a>	
Shows actions only when the selection paths matches a regular expression	33
<a href="#">shallot.ConfigurationValue.Category</a>	
Categories of <a href="#">shallot.ConfigurationValue</a> implementations	33
<a href="#">shallot.ConfigurationValue</a>	
Abstract base class for a <a href="#">configuration value</a>	34
<a href="#">shallot.Operations.FilesystemOperationStep.ConflictResolution</a>	
Enumeration of conflict resolution strategies	36
<a href="#">shallot.Actions.DefaultPrecedenceValues</a>	
Reference values for calculating default precedence values of open actions	38
<a href="#">shallot.DetailColumn</a>	
Abstract base class for a <a href="#">detail column</a> for <a href="#">shallot.Filesystem.Node</a> instances	39
<a href="#">shallot.Actions.DontResolveLinksPredicate</a>	
Disables links resolving	43
<a href="#">shallot.Environment</a>	
<a href="#">Environment</a> information and some <a href="#">utilities</a>	43
<a href="#">shallot.Eurl</a>	
A <a href="#">EURL</a>	45
<a href="#">shallot.Exceptions</a>	
Everything about <a href="#">exceptions</a>	51
<a href="#">shallot.Actions.ExecutionInfo</a>	
An object for signalling action execution state changes between an action implementation and the Shallot core (mostly to the core)	51
<a href="#">shallot.Actions.ExecutionUserFeedback</a>	
An object for communication with the user in action implementations	54
<a href="#">shallot.FilePropertyDialog</a>	
The <a href="#">file property dialog</a>	56
<a href="#">shallot.FileSearch</a>	
File <a href="#">searches</a>	57
<a href="#">shallot.Filesystem</a>	
The <a href="#">filesystem</a>	57
<a href="#">shallot.Operations.FilesystemOperation</a>	
A high-level provider of filesystem operations	61
<a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a>	
Abstract base class for progress monitors, used for monitoring progress of some operations in <a href="#">shallot.Operations.FilesystemOperation</a> and for conflict resolution	73
<a href="#">shallot.Operations.FilesystemOperationStep</a>	
One step in a larger filesystem transfer running inside some operations in <a href="#">shallot.Operations.FilesystemOperation</a>	75
<a href="#">shallot.Setting.GroupInfo</a>	
Enumeration of groups to which a setting can belong	76
<a href="#">shallot.Filesystem.Handler</a>	
Abstract base class for a custom filesystem handler	78
<a href="#">shallot.Operations.HandlerTransfer</a>	
Used for some additional functionality in some transfer operations in <a href="#">shallot.Filesystem.Handler</a> instances	93
<a href="#">shallot.Actions.HideOnCurrentDirectoryLevelPredicate</a>	
Shows actions only when not searching for 'current directory level' actions	94



<a href="#">shallot.Actions.HideOnSelectionLevelPredicate</a>	
Shows actions only when not searching for 'selection level' actions	94
<a href="#">shallot.IntlStringMap.IntlString</a>	
A multi-language string	95
<a href="#">shallot.IntlStringMap</a>	
A multi-language string map used for <a href="#">localization</a> of plugins	95
<a href="#">shallot.Exceptions.IOException</a>	
Shallot exception in IO	97
<a href="#">shallot.Actions.KeyShortcutPredicate</a>	
Sets a keyboard shortcut	98
<a href="#">shallot.Logging</a>	
Methods for <a href="#">logging</a>	98
<a href="#">shallot.MainWindow</a>	
A Shallot <a href="#">main window</a>	100
<a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a>	
Buttons in a message box from <a href="#">shallot.Actions.ExecutionUserFeedback</a>	101
<a href="#">shallot.PanelDetails.MultiSelectionPanelDetail</a>	
Abstract base class for a detail panel entry, which occurs when multiple items are selected	103
<a href="#">shallot.Filesystem.Node</a>	
A model representation of a location in the filesystem tree	106
<a href="#">shallot.Filesystem.NodeList</a>	
A list editor for a filesystem node list	107
<a href="#">shallot.Filesystem.NodeType</a>	
Enumeration of types a <a href="#">shallot.Filesystem.Node</a> can have	109
<a href="#">shallot.Actions.OnDirectoriesPredicate</a>	
Shows actions only on directories	111
<a href="#">shallot.Actions.OnFilesPredicate</a>	
Shows actions only on files	111
<a href="#">shallot.Actions.OnLinksPredicate</a>	
Shows actions only on links	112
<a href="#">shallot.Actions.OnSingleEntrySelectionPredicate</a>	
Shows actions only on single-entry selections	112
<a href="#">shallot.Operations.Operation</a>	
Transactional read and write filesystem accesses	113
<a href="#">shallot.Operations</a>	
Everything about <a href="#">operations on the filesystem</a>	116
<a href="#">shallot.PanelDetails.PanelDetail</a>	
Abstract base class for a detail panel entry	117
<a href="#">shallot.PanelDetails.PanelDetailInst</a>	
The storage for detail value rows, which are the actual content of a <a href="#">shallot.PanelDetails.PanelDetail</a> for an actual file selection	118
<a href="#">shallot.PanelDetails</a>	
Everything about <a href="#">panel details</a>	119
<a href="#">shallot.PanelDetails.PanelDetailValueElementButton</a>	
A link button (for executing some code on user behalf) as element of a panel detail row's value	120
<a href="#">shallot.PanelDetails.PanelDetailValueElementIcon</a>	
An icon as element of a panel detail row's value	121
<a href="#">shallot.PanelDetails.PanelDetailValueElementString</a>	
A piece of text as element of a panel detail row's value	122
<a href="#">shallot.PanelDetails.PanelDetailValueElementWaiting</a>	
A 'waiting' placeholder as element of a panel detail row's value	123
<a href="#">shallot.Actions.PositionIndexPredicate</a>	
Sets a positioning information index	124
<a href="#">shallot.Actions.Predicate</a>	
Controls when and how an action is created	125
<a href="#">shallot.Exceptions.ProgramException</a>	
Shallot exception for program errors (typically logical stuff) in the program or plugin	125

<a href="#">shallot.FilePropertyDialog.Tab.PropertyButtonConfig</a>	
Definitions for buttons in a <a href="#">shallot.FilePropertyDialog.Tab.PropertyConfig</a>	126
<a href="#">shallot.FilePropertyDialog.Tab.PropertyConfig</a>	
Definitions for properties, which present a piece of information in a <a href="#">shallot.FilePropertyDialog.Tab</a>	127
<a href="#">shallot.FilePropertyDialog.Tab.PropertyType</a>	
A type of a single property	128
<a href="#">shallot.Streaming.ReadDataDevice</a>	
A scripted source of binary content	129
<a href="#">shallot.Exceptions.RuntimeException</a>	
Shallot exception for failed operation due to (often external) runtime effects	130
<a href="#">shallot.Exceptions.ScriptedException</a>	
The Shallot exception class	131
<a href="#">shallot.FileSearch.SearchCriterion</a>	
Abstract base class for a search criterion	133
<a href="#">shallot.FileSearch.SearchCriterionFactory</a>	
A factory for a <a href="#">shallot.FileSearch.SearchCriterion</a> class	135
<a href="#">shallot.Setting</a>	
Abstract base class for a scripted Shallot setting (those who have to be stored manually)	138
<a href="#">shallot.PanelDetails.SingleSelectionPanelDetail</a>	
Abstract base class for a detail panel entry, which occurs when one item is selected	141
<a href="#">shallot.Streaming</a>	
Streaming binary data	144
<a href="#">shallot.Streaming.StreamReadDataDevice</a>	
A binary source backed by a file object	145
<a href="#">shallot.Actions.SubmenuAction</a>	
Abstract base class for a submenu action, which can be made visible in menus or the toolbar	146
<a href="#">shallot.FilePropertyDialog.Tab</a>	
Abstract base class for a tab in the Properties dialog	148
<a href="#">shallot.FilePropertyDialog.TabPropertyIconTextBanner</a>	
Represents values for image/text banners as used for <a href="#">shallot.FilePropertyDialog.Tab.PropertyType.IconTextBanner</a>	152
<a href="#">shallot.Environment.Thread</a>	
A thread executes code asynchronously	153
<a href="#">shallot.ThumbnailProvider</a>	
Abstract base class for a <a href="#">thumbnail</a> provider	155
<a href="#">shallot.Environment.Timer</a>	
A timers executes some code recurringly in some time interval	158

## 1.5 shallot Namespace Reference

The Shallot API for Python modules.

### Classes

- class [\\_ApiProxy](#)  
*Common superclass for many classes in this interface.*
- class [Actions](#)  
*Everything about [actions](#).*
- class [Bookmarks](#)  
*Everything about [Bookmarks](#).*
- class [ConfigurationValue](#)  
*Abstract base class for a [configuration value](#).*
- class [DetailColumn](#)

Abstract base class for a *detail column* for *shallot.Filesystem.Node* instances.

- class [Environment](#)  
*Environment* information and some *utilities*.
- class [Eurl](#)  
A *EURL*.
- class [Exceptions](#)  
Everything about *exceptions*.
- class [FilePropertyDialog](#)  
The *file property dialog*.
- class [FileSearch](#)  
File *searches*.
- class [Filesystem](#)  
The *filesystem*.
- class [IntlStringMap](#)  
A multi-language string map used for *localization* of plugins.
- class [Logging](#)  
Methods for *logging*.
- class [MainWindow](#)  
A Shallot *main window*.
- class [Operations](#)  
Everything about *operations on the filesystem*.
- class [PanelDetails](#)  
Everything about *panel details*.
- class [Setting](#)  
Abstract base class for a scripted Shallot setting (those who have to be stored manually).
- class [Streaming](#)  
*Streaming* binary data.
- class [ThumbnailProvider](#)  
Abstract base class for a *thumbnail* provider.

### 1.5.1 Detailed Description

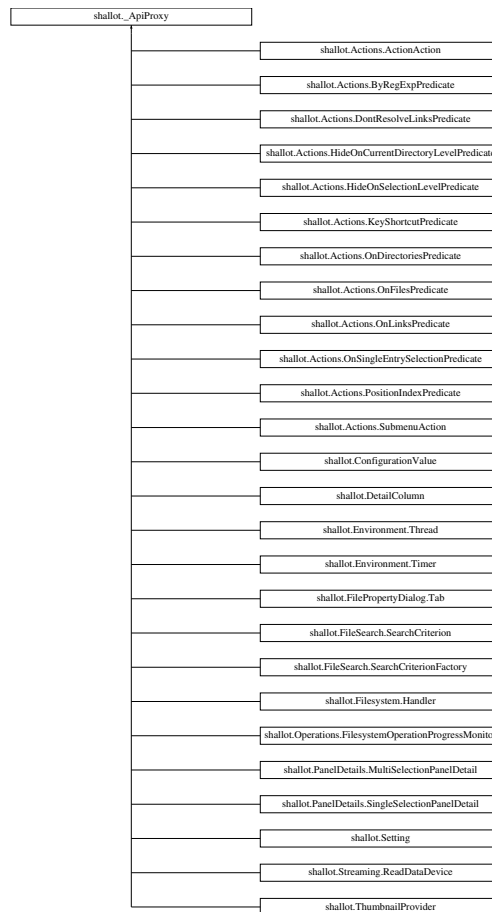
The Shallot API for Python modules.

The manual about the Shallot scripting API gives a full reference.

## 1.6 shallot.\_ApiProxy Class Reference

Common superclass for many classes in this interface.

Inheritance diagram for shallot.\_ApiProxy:



## Public Member Functions

- `def __init__(self, native)`

### 1.6.1 Detailed Description

Common superclass for many classes in this interface.

This class mostly has internal meaning and does not directly offer any services to the plugin developer. Read [more](#).

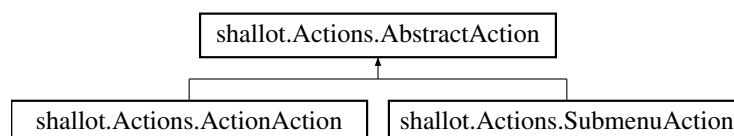
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.7 shallot.Actions.AbstractAction Class Reference

Abstract base class for executable actions or submenu structures of them.

Inheritance diagram for `shallot.Actions.AbstractAction`:



## Public Member Functions

- def `initialize` (self)  
*Initialize the action.*
- def `set_enabled` (self, value)  
*Sets if the item is enabled.*
- def `set_visible` (self, value)  
*Sets the visibility of this item.*
- def `enabled` (self)  
*Checks if this action is enabled.*
- def `visible` (self)  
*Checks the visibility of this item (non-recursively).*

### 1.7.1 Detailed Description

Abstract base class for executable actions or submenu structures of them.

See the subclasses of this class. They can be registered with [shallot.Actions.register](#). They can be returned from [shallot.Filesystem.Handler.get\\_actions](#).

### 1.7.2 Member Function Documentation

#### 1.7.2.1 `initialize()`

```
def shallot.Actions.AbstractAction.initialize (  
    self )
```

Initialize the action.

This should make the time-consuming parts, e.g. for determining a label or enabled state.

Override this method in custom subclasses or leave the default implementation.

#### 1.7.2.2 `set_enabled()`

```
def shallot.Actions.AbstractAction.set_enabled (  
    self,  
    value )
```

Sets if the item is enabled.

**Parameters**

<i>value</i>	The new value.
--------------	----------------

**1.7.2.3 set\_visible()**

```
def shallot.Actions.AbstractAction.set_visible (
    self,
    value )
```

Sets the visibility of this item.

**Parameters**

<i>value</i>	The new value.
--------------	----------------

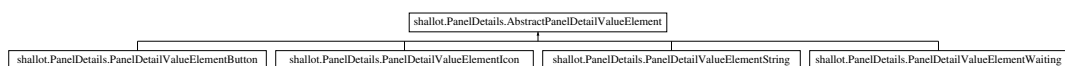
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

**1.8 shallot.PanelDetails.AbstractPanelDetailValueElement Class Reference**

Abstract base class for an element of a panel detail row's value.

Inheritance diagram for shallot.PanelDetails.AbstractPanelDetailValueElement:

**Public Member Functions**

- `def __init__(self, type, value)`

**Public Attributes**

- **type**
- **value**

**1.8.1 Detailed Description**

Abstract base class for an element of a panel detail row's value.

This can be something like a text, an icon or a link button. See the subclasses.

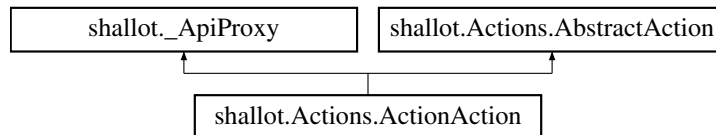
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.9 shallot.Actions.ActionAction Class Reference

Abstract base class for an executable action, which can be made visible in menus or the toolbar or executed directly.

Inheritance diagram for shallot.Actions.ActionAction:



### Public Member Functions

- def `__init__` (self, text, `enabled`=True, icon="", defaultActionPrecedence=0)
- def `action` (self, info)
  - The action implementation, i.e.*
- def `execute` (self)
  - Executes this action.*
- def `initialize_sync` (self)
  - Initializes the action.*
- def `initialize` (self)
  - Initialize the action.*
- def `set_enabled` (self, value)
  - Sets if the item is enabled.*
- def `set_visible` (self, value)
  - Sets the visibility of this item.*
- def `enabled` (self)
  - Checks if this action is enabled.*
- def `visible` (self)
  - Checks the visibility of this item (non-recursively).*

### 1.9.1 Detailed Description

Abstract base class for an executable action, which can be made visible in menus or the toolbar or executed directly.

See [shallot.Actions.AbstractAction](#) for more.

### 1.9.2 Constructor & Destructor Documentation

#### 1.9.2.1 `__init__()`

```

def shallot.Actions.ActionAction.__init__ (
    self,
    text,
    enabled = True,
    icon = "",
    defaultActionPrecedence = 0 )
  
```

## Parameters

<i>text</i>	A label text (can be "" if you want to directly execute it instead of adding it to some menu structure).
<i>enabled</i>	If this action is enabled.
<i>icon</i>	Name of an icon.
<i>defaultActionPrecedence</i>	Precedence value for being a default action. This int value must be higher than all others for becoming the default action. See <a href="#">shallot.Actions.DefaultPrecedenceValues</a> .

### 1.9.3 Member Function Documentation

#### 1.9.3.1 action()

```
def shallot.Actions.ActionAction.action (
    self,
    info )
```

The action implementation, i.e.

what the actions should actually do.

Override this method in custom subclasses.

## Parameters

<i>info</i>	A <a href="#">shallot.Actions.ExecutionInfo</a> execution info object.
-------------	--

#### 1.9.3.2 initialize()

```
def shallot.Actions.AbstractAction.initialize (
    self ) [inherited]
```

Initialize the action.

This should make the time-consuming parts, e.g. for determining a label or enabled state.

Override this method in custom subclasses or leave the default implementation.



### 1.9.3.3 initialize\_sync()

```
def shallot.Actions.ActionAction.initialize_sync (
    self )
```

Initializes the action.

This can be called from outside in order to do the initialization.

### 1.9.3.4 set\_enabled()

```
def shallot.Actions.AbstractAction.set_enabled (
    self,
    value ) [inherited]
```

Sets if the item is enabled.

#### Parameters

<i>value</i>	The new value.
--------------	----------------

### 1.9.3.5 set\_visible()

```
def shallot.Actions.AbstractAction.set_visible (
    self,
    value ) [inherited]
```

Sets the visibility of this item.

#### Parameters

<i>value</i>	The new value.
--------------	----------------

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.10 shallot.Actions Class Reference

Everything about [actions](#).

### Classes

- class [AbstractAction](#)

- Abstract base class for executable actions or submenu structures of them.*
- class [ActionAction](#)
  - Abstract base class for an executable action, which can be made visible in menus or the toolbar or executed directly.*
- class [ByRegExpPredicate](#)
  - Shows actions only when the selection paths matches a regular expression.*
- class [DefaultPrecedenceValues](#)
  - Reference values for calculating default precedence values of open actions.*
- class [DontResolveLinksPredicate](#)
  - Disables links resolving.*
- class [ExecutionInfo](#)
  - An object for signalling action execution state changes between an action implementation and the Shallot core (mostly to the core).*
- class [ExecutionUserFeedback](#)
  - An object for communication with the user in action implementations.*
- class [HideOnCurrentDirectoryLevelPredicate](#)
  - Shows actions only when not searching for 'current directory level' actions.*
- class [HideOnSelectionLevelPredicate](#)
  - Shows actions only when not searching for 'selection level' actions.*
- class [KeyShortcutPredicate](#)
  - Sets a keyboard shortcut.*
- class [OnDirectoriesPredicate](#)
  - Shows actions only on directories.*
- class [OnFilesPredicate](#)
  - Shows actions only on files.*
- class [OnLinksPredicate](#)
  - Shows actions only on links.*
- class [OnSingleEntrySelectionPredicate](#)
  - Shows actions only on single-entry selections.*
- class [PositionIndexPredicate](#)
  - Sets a positioning information index.*
- class [Predicate](#)
  - Controls when and how an action is created.*
- class [SubmenuAction](#)
  - Abstract base class for a submenu action, which can be made visible in menus or the toolbar.*

## Static Public Member Functions

- def [register](#) (actiontype, category="manage", predicates=[])
- Registers a subclass of [shallot.Actions.AbstractAction](#).*

## Public Attributes

- **ActionType**

### 1.10.1 Detailed Description

Everything about [actions](#).

## 1.10.2 Member Function Documentation

### 1.10.2.1 register()

```
def shallot.Actions.register (
    actiontype,
    category = "manage",
    predicates = [] ) [static]
```

Registers a subclass of [shallot.Actions.AbstractAction](#).

This makes it permanently visible in the context menu of selections and/or in the toolbar. If and where the action is visible depends on the action implementation (see constructor parameters), the situation (what is selected in the user interface?) and the registration mode.

#### Parameters

<i>actiontype</i>	The class implementing <a href="#">shallot.Actions.AbstractAction</a> . Its constructor must be callable with just <code>actiontype(nodes)</code> . The <code>nodes</code> parameter contains a list of <a href="#">shallot.Filesystem.Node</a> containing the node selection.
<i>category</i>	The action category. Used for grouping in menus. Typically available are "create", "open" and "manage".
<i>predicates</i>	A list of <a href="#">shallot.Actions.Predicate</a> controlling when and how the action is offered.

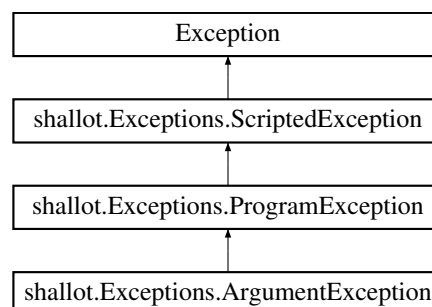
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.11 shallot.Exceptions.ArgumentException Class Reference

Shallot exception for failed operation due to invalid arguments given to some program part.

Inheritance diagram for `shallot.Exceptions.ArgumentException`:



## Public Member Functions

- `def __init__ (self, details=None, message=None, isResumeable=True, detailsAreInteresting=None, _↵  
class="")`  
See [shallot.Exceptions.ScriptedException.\\_\\_init\\_\\_](#) for details.
- `def isExceptionClass (self, classname)`  
Checks if this exception is instance of a given exception class.

### 1.11.1 Detailed Description

Shallot exception for failed operation due to invalid arguments given to some program part.

It typically allows resume but not retry (special cases may override each of them). Read [more about Shallot Exceptions](#).

### 1.11.2 Member Function Documentation

#### 1.11.2.1 isExceptionClass()

```
def shallot.Exceptions.ScriptedException.isExceptionClass (
    self,
    classname ) [inherited]
```

Checks if this exception is instance of a given exception class.

#### Parameters

<i>classname</i>	A exception class name (as string).
------------------	-------------------------------------

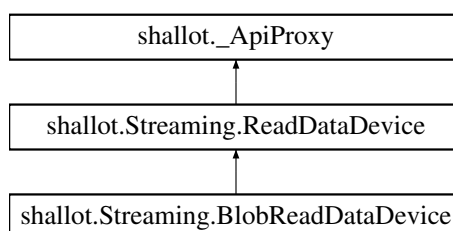
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.12 shallot.Streaming.BlobReadDataDevice Class Reference

A binary source backed by static binary content in a byte array.

Inheritance diagram for `shallot.Streaming.BlobReadDataDevice`:



## Public Member Functions

- def `__init__` (self, content)
- def `getdata` (self)  
See `shallot.Streaming.ReadDataDevice`.
- def `read` (self)  
Reads a chunk of data as Python byte array.
- def `readall` (self)  
Reads the complete data as Python byte array.

## Public Attributes

- **content**

### 1.12.1 Detailed Description

A binary source backed by static binary content in a byte array.

### 1.12.2 Constructor & Destructor Documentation

#### 1.12.2.1 `__init__`()

```
def shallot.Streaming.BlobReadDataDevice.__init__ (
    self,
    content )
```

#### Parameters

<code>content</code>	The byte array containing the content.
----------------------	--

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.13 shallot.Bookmarks.Bookmark Class Reference

A bookmark.

## Public Member Functions

- def `__init__` (self)

- Can't be constructed directly.*
- `def url (self)`  
*The location this bookmark points to.*
- `def folder (self)`  
*The subcollection (as list of strings).*
- `def id (self)`  
*The bookmark id.*
- `def label (self)`  
*A textual description of this bookmark.*
- `def tags (self)`  
*The tags strings.*

### 1.13.1 Detailed Description

A bookmark.

Change it with the methods in [shallot.Bookmarks](#).

### 1.13.2 Member Function Documentation

#### 1.13.2.1 id()

```
def shallot.Bookmarks.Bookmark.id (  
    self )
```

The bookmark id.

It is not display anywhere but only used for the management methods in [shallot.Bookmarks](#).

#### 1.13.2.2 tags()

```
def shallot.Bookmarks.Bookmark.tags (  
    self )
```

The tags strings.

This value is solely used by plugins for their bookkeeping. It can allow to programmatically find a particular bookmark (if the creator wrote meaningful information in it).

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.14 shallot.Bookmarks Class Reference

Everything about [Bookmarks](#).

### Classes

- class [Bookmark](#)  
*A bookmark.*

### Static Public Member Functions

- def [get\\_bookmarks](#) ()  
*Returns a list of all existing [shallot.Bookmarks.Bookmark](#) instances.*
- def [has\\_bookmarks](#) ()  
*Returns True if there are any bookmarks stored.*
- def [add\\_bookmark](#) (eurl, folder=[], label=None, tags=None)  
*Adds a new bookmark.*
- def [remove\\_bookmark](#) (id)  
*Removes a bookmark.*
- def [change\\_bookmark](#) (id, eurl, label=None)  
*Changes bookmark data.*
- def [change\\_bookmark\\_tags](#) (id, tags)  
*Changes bookmark tags.*
- def [move\\_bookmark\\_up](#) (id)  
*Moves a bookmark upwards in its collection.*
- def [move\\_bookmark\\_down](#) (id)  
*Moves a bookmark downwards in its collection.*
- def [move\\_bookmark\\_to\\_folder](#) (id, folder)  
*Moves a bookmark to another subcollection.*

### 1.14.1 Detailed Description

Everything about [Bookmarks](#).

### 1.14.2 Member Function Documentation

#### 1.14.2.1 [add\\_bookmark\(\)](#)

```
def shallot.Bookmarks.add_bookmark (
    eurl,
    folder = [],
    label = None,
    tags = None ) [static]
```

Adds a new bookmark.

**Parameters**

<i>eurl</i>	The location to bookmark.
<i>folder</i>	A string list describing in which subcollection this bookmark should be placed.
<i>label</i>	The textual description of the new bookmark.
<i>tags</i>	A string for internal bookkeeping. Use it for recognizing your own bookmarks later on. See <a href="#">shallot.Bookmarks.Bookmark.tags</a> .

**1.14.2.2 change\_bookmark()**

```
def shallot.Bookmarks.change_bookmark (
    id,
    eurl,
    label = None ) [static]
```

Changes bookmark data.

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
<i>eurl</i>	The new location.
<i>label</i>	The new textual description.

**1.14.2.3 change\_bookmark\_tags()**

```
def shallot.Bookmarks.change_bookmark_tags (
    id,
    tags ) [static]
```

Changes bookmark tags.

See [shallot.Bookmarks.Bookmark.tags](#).

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
<i>tags</i>	The new tags string.s

**1.14.2.4 move\_bookmark\_down()**

```
def shallot.Bookmarks.move_bookmark_down (
    id ) [static]
```



Moves a bookmark downwards in its collection.

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
-----------	--

**1.14.2.5 move\_bookmark\_to\_folder()**

```
def shallot.Bookmarks.move_bookmark_to_folder (  
    id,  
    folder ) [static]
```

Moves a bookmark to another subcollection.

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
<i>folder</i>	The new subcollection (as list of strings).

**1.14.2.6 move\_bookmark\_up()**

```
def shallot.Bookmarks.move_bookmark_up (  
    id ) [static]
```

Moves a bookmark upwards in its collection.

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
-----------	--

**1.14.2.7 remove\_bookmark()**

```
def shallot.Bookmarks.remove_bookmark (  
    id ) [static]
```

Removes a bookmark.

**Parameters**

<i>id</i>	The bookmark id this call is about (See <a href="#">shallot.Bookmarks.Bookmark.id</a> ).
-----------	--

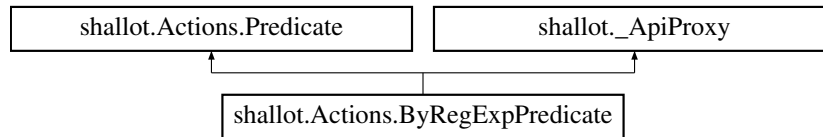
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.15 shallot.Actions.ByRegExpPredicate Class Reference

Shows actions only when the selection paths matches a regular expression.

Inheritance diagram for shallot.Actions.ByRegExpPredicate:



### Public Member Functions

- `def __init__(self, pattern)`

#### 1.15.1 Detailed Description

Shows actions only when the selection paths matches a regular expression.

See base class for more information.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.16 shallot.ConfigurationValue.Category Class Reference

Categories of [shallot.ConfigurationValue](#) implementations.

### Static Public Attributes

- `GUI = internalstuff.configurationcategory_gui()`  
*User interface category.*
- `Behavior = internalstuff.configurationcategory_behavior()`  
*Behavioral configuration category.*
- `ExternalTools = internalstuff.configurationcategory_externaltools()`  
*External tools category.*

#### 1.16.1 Detailed Description

Categories of [shallot.ConfigurationValue](#) implementations.

They are used for grouping them in the dialogs. There is no difference in behavior implied by this choice.

## 1.16.2 Member Data Documentation

### 1.16.2.1 Behavior

```
shallot.ConfigurationValue.Category.Behavior = internalstuff.configurationcategory_behavior()  
[static]
```

Behavioral configuration category.

### 1.16.2.2 ExternalTools

```
shallot.ConfigurationValue.Category.ExternalTools = internalstuff.configurationcategory_↔  
externaltools() [static]
```

External tools category.

### 1.16.2.3 GUI

```
shallot.ConfigurationValue.Category.GUI = internalstuff.configurationcategory_gui() [static]
```

User interface category.

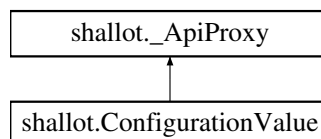
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.17 shallot.ConfigurationValue Class Reference

Abstract base class for a [configuration value](#).

Inheritance diagram for shallot.ConfigurationValue:



### Classes

- class [Category](#)  
*Categories of [shallot.ConfigurationValue](#) implementations.*

## Public Member Functions

- def `__init__` (self, name, defaultvalue, category=0, description="", longdescription="", changehint=None)
- def `value` (self)  
*Returns the current value.*
- def `set_value` (self, `value`)  
*Sets the value.*

## Public Attributes

- `vtype`

### 1.17.1 Detailed Description

Abstract base class for a [configuration value](#).

### 1.17.2 Constructor & Destructor Documentation

#### 1.17.2.1 `__init__()`

```
def shallot.ConfigurationValue.__init__ (
    self,
    name,
    defaultvalue,
    category = 0,
    description = "",
    longdescription = "",
    changehint = None )
```

#### Parameters

<i>name</i>	The config value name.
<i>defaultvalue</i>	The default value.
<i>category</i>	The category. One of <a href="#">shallot.ConfigurationValue.Category</a> .
<i>description</i>	The short description.
<i>longdescription</i>	The long description.
<i>changehint</i>	A label hint for change buttons in user interfaces.

### 1.17.3 Member Function Documentation

### 1.17.3.1 `set_value()`

```
def shallot.ConfigurationValue.set_value (
    self,
    value )
```

Sets the value.

#### Parameters

<i>value</i>	The new value.
--------------	----------------

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.18 shallot.Operations.FilesystemOperationStep.ConflictResolution Class Reference

Enumeration of conflict resolution strategies.

### Static Public Attributes

- [MergeDirectories](#) = internalstuff.operationstep\_conflictresolution\_mergedirectories()  
*Merge source into destination directory.*
- [OverwriteDestination](#) = internalstuff.operationstep\_conflictresolution\_overwritdestination()  
*Overwrite the destination.*
- [RenameDestinationBefore](#) = internalstuff.operationstep\_conflictresolution\_renamedestinationbefore()  
*Rename the file at the destination before transferring.*
- [UseDifferentDestinationName](#) = internalstuff.operationstep\_conflictresolution\_usedifferentdestinationname()  
*Transfer to another destination filename.*
- [Skip](#) = internalstuff.operationstep\_conflictresolution\_skip()  
*Skip this element.*
- [Unresolved](#) = internalstuff.operationstep\_conflictresolution\_unresolved()  
*No strategy.*
- [Indirect](#) = internalstuff.operationstep\_conflictresolution\_indirect()  
*Indirectly solved.*

### 1.18.1 Detailed Description

Enumeration of conflict resolution strategies.

### 1.18.2 Member Data Documentation

### 1.18.2.1 Indirect

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.Indirect = internalstuff.↔  
operationstep_conflictresolution_indirect() [static]
```

Indirectly solved.

Only set by the engine.

### 1.18.2.2 MergeDirectories

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.MergeDirectories = internalstuff.↔  
operationstep_conflictresolution_mergedirectories() [static]
```

Merge source into destination directory.

### 1.18.2.3 OverwriteDestination

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.OverwriteDestination = internalstuff.↔  
operationstep_conflictresolution_overwritedestination() [static]
```

Overwrite the destination.

### 1.18.2.4 RenameDestinationBefore

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.RenameDestinationBefore = internalstuff.↔  
operationstep_conflictresolution_renamedestinationbefore() [static]
```

Rename the file at the destination before transferring.

### 1.18.2.5 Skip

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.Skip = internalstuff.operationstep↔  
_conflictresolution_skip() [static]
```

Skip this element.

### 1.18.2.6 Unresolved

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.Unresolved = internalstuff.↔  
operationstep_conflictresolution_unresolved() [static]
```

No strategy.

### 1.18.2.7 UseDifferentDestinationName

```
shallot.Operations.FilesystemOperationStep.ConflictResolution.UseDifferentDestinationName =
internalstuff.operationstep_conflictresolution_usedifferentdestinationname() [static]
```

Transfer to another destination filename.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.19 shallot.Actions.DefaultPrecedenceValues Class Reference

Reference values for calculating default precedence values of open actions.

### Static Public Attributes

- [OpenFile](#) = internalstuff.actiondefaultprecedencevalues\_openfile()  
*Normal file open action.*
- [BuiltinSpecialOpen](#) = internalstuff.actiondefaultprecedencevalues\_builtin\_specialopen()  
*Special open action.*

### 1.19.1 Detailed Description

Reference values for calculating default precedence values of open actions.

### 1.19.2 Member Data Documentation

#### 1.19.2.1 BuiltinSpecialOpen

```
shallot.Actions.DefaultPrecedenceValues.BuiltinSpecialOpen = internalstuff.actiondefaultprecedencevalues↔
_builtin_specialopen() [static]
```

Special open action.

#### 1.19.2.2 OpenFile

```
shallot.Actions.DefaultPrecedenceValues.OpenFile = internalstuff.actiondefaultprecedencevalues↔
_openfile() [static]
```

Normal file open action.

The documentation for this class was generated from the following file:

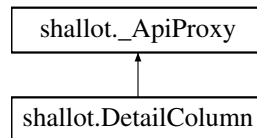
- /home/pino/projects/shallot/shallot.py



## 1.20 shallot.DetailColumn Class Reference

Abstract base class for a [detail column](#) for [shallot.Filesystem.Node](#) instances.

Inheritance diagram for shallot.DetailColumn:



### Public Member Functions

- `def __init__ (self, name, displayname, positionGroup=None, positionIndex=None, sort_doTypediff=True, defaultWidth=-1, isRightAligned=False)`
- `def determine_value (self, node, operation)`  
*Determines the column value for a node.*
- `def apply_value (self, eurl, operation, value)`  
*Set the detail value for a given eurl.*
- `def compute_value (self, node, operation)`  
*Queries the column value for a node.*

### Static Public Member Functions

- `def find_by_name (name)`  
*Finds a [shallot.DetailColumn](#) implementation by name.*
- `def register_as_transferrable (index, detailcolumn)`  
*Registers a [shallot.DetailColumn](#) for transferring those details in file transfers (e.g.*

### Static Public Attributes

- `INDEX_CORE = internalstuff.displayindex_core()`  
*A integer value to be used as base display index for very interesting columns.*
- `INDEX_INTERESTING = internalstuff.displayindex_interesting()`  
*A integer value to be used as base display index for interesting columns.*
- `INDEX_EXOTIC = internalstuff.displayindex_exotic()`  
*A integer value to be used as base display index for not generally interesting columns.*

#### 1.20.1 Detailed Description

Abstract base class for a [detail column](#) for [shallot.Filesystem.Node](#) instances.

Those can e.g. be seen in the file list view, but can also be used internally by other places.

It encapsulates the retrieval logic and metadata for one piece of additional information a [shallot.Filesystem.Node](#) can have (e.g. the filesize). Retrieving the values is designed to be asynchronous. Each instance represents one column, while the actual logic is implemented in subclasses. For a new detail column, subclass this class and implement at least [shallot.DetailColumn.determine\\_value](#).

## 1.20.2 Constructor & Destructor Documentation

### 1.20.2.1 `__init__()`

```
def shallot.DetailColumn.__init__ (
    self,
    name,
    displayname,
    positionGroup = None,
    positionIndex = None,
    sort_doTypediff = True,
    defaultWidth = -1,
    isRightAligned = False )
```

#### Parameters

<i>name</i>	The internal name (must be unique).
<i>displayname</i>	The label text.
<i>positionGroup</i>	Controls display order. See <a href="#">Position Indexes</a> for details. Use one of the <code>INDEX_*</code> values from <a href="#">shallot.DetailColumn</a> .
<i>positionIndex</i>	Controls display order. See <a href="#">Position Indexes</a> for details.
<i>sort_doTypediff</i>	Shall sorting differ between files and directories?
<i>defaultWidth</i>	The default width in pixels (optional).
<i>isRightAligned</i>	If the column values are right aligned (optional).

## 1.20.3 Member Function Documentation

### 1.20.3.1 `apply_value()`

```
def shallot.DetailColumn.apply_value (
    self,
    eurl,
    operation,
    value )
```

Set the detail value for a given eurl.

Used for transferring details in file transfers (see [shallot.DetailColumn.register\\_as\\_transferrable](#)).

Override this method in custom subclasses or leave the default implementation.

## Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> for which the value must be determined.
<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>value</i>	The detail value (as string).

## 1.20.3.2 compute\_value()

```
def shallot.DetailColumn.compute_value (
    self,
    node,
    operation )
```

Queries the column value for a node.

## Parameters

<i>node</i>	The <a href="#">shallot.Filesystem.Node</a> node for which the value must be determined.
<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.

## 1.20.3.3 determine\_value()

```
def shallot.DetailColumn.determine_value (
    self,
    node,
    operation )
```

Determines the column value for a node.

Override this method in custom subclasses. Only used internally. For querying foreign detail columns, use `compute_value` instead.

## Parameters

<i>node</i>	The <a href="#">shallot.Filesystem.Node</a> node for which the value must be determined.
<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.

## 1.20.3.4 find\_by\_name()

```
def shallot.DetailColumn.find_by_name (
    name ) [static]
```

Finds a [shallot.DetailColumn](#) implementation by name.

#### Parameters

<i>name</i>	The detail column name.
-------------	-------------------------

#### 1.20.3.5 register\_as\_transferrable()

```
def shallot.DetailColumn.register_as_transferrable (
    index,
    detailcolumn ) [static]
```

Registers a [shallot.DetailColumn](#) for transferring those details in file transfers (e.g. when the user copies files). This requires to implement some methods.

#### Parameters

<i>index</i>	An integer which controls the order of transferring.
<i>detailcolumn</i>	The <a href="#">shallot.DetailColumn</a> detail column.

### 1.20.4 Member Data Documentation

#### 1.20.4.1 INDEX\_CORE

```
shallot.DetailColumn.INDEX_CORE = internalstuff.displayindex_core() [static]
```

A integer value to be used as base display index for very interesting columns.

#### 1.20.4.2 INDEX\_EXOTIC

```
shallot.DetailColumn.INDEX_EXOTIC = internalstuff.displayindex_exotic() [static]
```

A integer value to be used as base display index for not generally interesting columns.

## 1.20.4.3 INDEX\_INTERESTING

```
shallot.DetailColumn.INDEX_INTERESTING = internalstuff.displayindex_interesting() [static]
```

A integer value to be used as base display index for interesting columns.

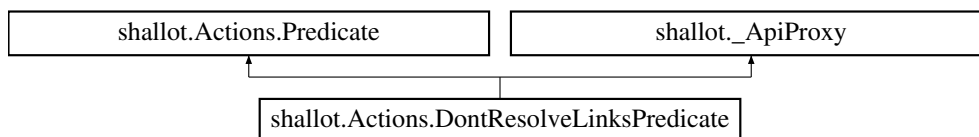
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.21 shallot.Actions.DontResolveLinksPredicate Class Reference

Disables links resolving.

Inheritance diagram for shallot.Actions.DontResolveLinksPredicate:



## Public Member Functions

- def `__init__`(self)

## 1.21.1 Detailed Description

Disables links resolving.

See base class for more information.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.22 shallot.Environment Class Reference

[Environment](#) information and some [utilities](#).

## Classes

- class [Thread](#)  
A thread executes code asynchronously.
- class [Timer](#)  
A timers executes some code recurringly in some time interval.

## Static Public Attributes

- `is_debug_build` = `internalstuff.is_debug_build()`  
*If Shallot was built in debug mode.*
- `shallot_data_dir` = `internalstuff.shallotDataDir()`  
*The directory containing the shallot data files.*
- `shallot_builtinplugin_dir` = `internalstuff.shallotBuiltinPluginDir()`  
*The directory containing the shallot builtin plugin files.*
- `shallot_systemplugin_dir` = `internalstuff.shallotSystemPluginDir()`  
*The directory containing the shallot system plugin files.*
- `shallot_userplugin_dir` = `internalstuff.shallotUserPluginDir()`  
*The directory containing the shallot user plugin files.*
- `shallot_language` = `internalstuff.shallotLanguage()`  
*The language code of the current user interface language.*

### 1.22.1 Detailed Description

[Environment](#) information and some [utilities](#).

### 1.22.2 Member Data Documentation

#### 1.22.2.1 `is_debug_build`

```
shallot.Environment.is_debug_build = internalstuff.is_debug_build() [static]
```

If Shallot was built in debug mode.

#### 1.22.2.2 `shallot_builtinplugin_dir`

```
shallot.Environment.shallot_builtinplugin_dir = internalstuff.shallotBuiltinPluginDir() [static]
```

The directory containing the shallot builtin plugin files.

#### 1.22.2.3 `shallot_data_dir`

```
shallot.Environment.shallot_data_dir = internalstuff.shallotDataDir() [static]
```

The directory containing the shallot data files.

## 1.22.2.4 shallot\_language

```
shallot.Environment.shallot_language = internalstuff.shallotLanguage() [static]
```

The language code of the current user interface language.

## 1.22.2.5 shallot\_systemplugin\_dir

```
shallot.Environment.shallot_systemplugin_dir = internalstuff.shallotSystemPluginDir() [static]
```

The directory containing the shallot system plugin files.

## 1.22.2.6 shallot\_userplugin\_dir

```
shallot.Environment.shallot_userplugin_dir = internalstuff.shallotUserPluginDir() [static]
```

The directory containing the shallot user plugin files.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.23 shallot.Eurl Class Reference

A [EURL](#).

## Public Member Functions

- `def __init__ (self)`  
*Can't be constructed directly.*
- `def as_string (self)`  
*Returns the textual value.*
- `def __str__ (self)`
- `def basename (self)`  
*Returns the last path segment.*
- `def parent_segment (self)`  
*Returns the parent [shallot.Eurl](#).*
- `def with_appended_segment (self, segment)`  
*Returns a new [shallot.Eurl](#) from this one with `"/basename"` appended.*
- `def with_appended_segments (self, path)`  
*Returns a new [shallot.Eurl](#) with path segments `"/pa/t/h/"` appended.*
- `def enwrap_with_outer_url (self, scheme, hostname, path)`  
*Returns a new [shallot.Eurl](#) containing this one packed as embedding and new outer parts `scheme, hostname and path`.*

- def `path` (self)  
*Returns the path part (from the outer url of this `shallot.Eurl`).*
- def `root` (self)  
*Returns the root `shallot.Eurl` from this one.*
- def `hostname` (self)  
*Returns the hostname part (from the outer url of this `shallot.Eurl`).*
- def `scheme` (self)  
*Returns the scheme (from the outer url of this `shallot.Eurl`).*
- def `outer_url` (self)  
*Returns a new `shallot.Eurl` containing only the outer part of this one (strips the embeddings).*
- def `outermost_inner_eurl` (self)  
*Returns a new `shallot.Eurl` containing only the embedding of this one.*
- def `has_inner_urls` (self)  
*Checks if this `shallot.Eurl` has embeddings.*
- def `outer_url_is_root_directory` (self)  
*Checks if this `shallot.Eurl` is a root path (with or without embeddings).*
- def `has_parent_segment` (self)  
*Checks if this `shallot.Eurl` has a parent segment.*
- def `is_prefix_of` (self, eurl)  
*Checks if this `shallot.Eurl` is a prefix of another one.*

## Static Public Member Functions

- def `from_string` (s)  
*Returns a `shallot.Eurl` from a string.*
- def `create` (scheme, hostname, path)  
*Returns a `shallot.Eurl` from three parts.*

### 1.23.1 Detailed Description

A `EURL`.

You can get instances with global methods like `shallot.Eurl.create` and `shallot.Eurl.from_string`.

Please note: Instances can be associated with any kind of elements in the filesystem (files, directories, links, ...). It might also point to something which does not exist at all. The documentation sometimes explicitly makes a difference between those kinds (files, directories, links, ...; often called 'node type'). But it often uses the term 'file' implicitly while meaning all kinds of elements; assuming that e.g. a directory is just a special kind of a file. It should be clear from the particular context which meaning applies.

### 1.23.2 Member Function Documentation

#### 1.23.2.1 `as_string()`

```
def shallot.Eurl.as_string (
    self )
```

Returns the textual value.

This is what `shallot.Eurl.from_string` would expect as parameter.



### 1.23.2.2 basename()

```
def shallot.Eurl.basename (
    self )
```

Returns the last path segment.

This is the text behind the last slash. Examples: "baz" for `file:///foo/bar/baz`. "" for `file:///`.

### 1.23.2.3 create()

```
def shallot.Eurl.create (
    scheme,
    hostname,
    path ) [static]
```

Returns a `shallot.Eurl` from three parts.

The structure is `scheme://hostname/path`.

#### Parameters

<i>scheme</i>	The scheme part.
<i>hostname</i>	The hostname.
<i>path</i>	The path.

### 1.23.2.4 enwrap\_with\_outer\_url()

```
def shallot.Eurl.enwrap_with_outer_url (
    self,
    scheme,
    hostname,
    path )
```

Returns a new `shallot.Eurl` containing this one packed as embedding and new outer parts `scheme`, `hostname` and `path`.

Example: `scheme:[file:///a/b/c.zip]/hostname/p/a/t/h` for `file:///a/b/c.zip`.

#### Parameters

<i>scheme</i>	The scheme part of the new outer eurl.
<i>hostname</i>	The hostname of the new outer eurl.
<i>path</i>	The path of the new outer eurl.

### 1.23.2.5 from\_string()

```
def shallot.Eurl.from_string (
    s ) [static]
```

Returns a [shallot.Eurl](#) from a string.

#### Parameters

s	The eurl string. This is what <a href="#">shallot.Eurl.as_string</a> would return.
---	--

### 1.23.2.6 has\_inner\_urls()

```
def shallot.Eurl.has_inner_urls (
    self )
```

Checks if this [shallot.Eurl](#) has embeddings.

Examples: true for `foo:[bar:///foo]/host/`. false for `foo://host/`.

### 1.23.2.7 has\_parent\_segment()

```
def shallot.Eurl.has_parent_segment (
    self )
```

Checks if this [shallot.Eurl](#) has a parent segment.

This indicates if [shallot.Eurl.parent\\_segment](#) would return 0.

### 1.23.2.8 hostname()

```
def shallot.Eurl.hostname (
    self )
```

Returns the hostname part (from the outer url of this [shallot.Eurl](#)).

Examples: "livingroom-pc" for `smb://livingroom-pc/foo/bar/baz`. "" for `file:///foo/bar/baz`.

### 1.23.2.9 is\_prefix\_of()

```
def shallot.Eurl.is_prefix_of (
    self,
    eurl )
```

Checks if this [shallot.Eurl](#) is a prefix of another one.

This is not an equivalent to a string comparison but it checks parent relationships according to [shallot.Eurl.parent\\_segment](#).

## Parameters

<i>eurl</i>	The longer <a href="#">shallot.Eurl</a> .
-------------	---

1.23.2.10 `outer_url()`

```
def shallot.Eurl.outer_url (
    self )
```

Returns a new [shallot.Eurl](#) containing only the outer part of this one (strips the embeddings).

Example: `foobar://host/foo/bar/baz` for `foobar:[zip:[file:///a/b/c.zip]//d/e]//foo/bar/baz`.

1.23.2.11 `outer_url_is_root_directory()`

```
def shallot.Eurl.outer_url_is_root_directory (
    self )
```

Checks if this [shallot.Eurl](#) is a root path (with or without embeddings).

Examples: True for `foo://host/`. False for `foo://host/a`. True for `foo:[bar:///goo]/host/`. False for `foo:[bar:///goo]/host/a`. True for `foo:[bar:///goo]//`. False for `foo:[bar:///goo]//a`.

1.23.2.12 `outermost_inner_eurl()`

```
def shallot.Eurl.outermost_inner_eurl (
    self )
```

Returns a new [shallot.Eurl](#) containing only the embedding of this one.

Example: `zip:[file:///a/b/c.zip]//d/e` for `foobar:[zip:[file:///a/b/c.zip]//d/e]//foo/bar/b`

1.23.2.13 `parent_segment()`

```
def shallot.Eurl.parent_segment (
    self )
```

Returns the parent [shallot.Eurl](#).

At first, this traverses path segments. For a root path `eurl` with embeddings, it returns the embedding. If none are available, it returns 0. Examples: `foo://host/foo` for `foo://host/foo/bar`. `foo://host/` for `foo://host/boo`. `foo:[bar://host/goo]/anotherhost/` for `foo:[bar://host/goo]/anotherhost/boo`. `bar://host/foo` for `foo:[bar://host/foo]/host/`. 0 for `foo://host/`.

#### 1.23.2.14 path()

```
def shallot.Eurl.path (
    self )
```

Returns the path part (from the outer url of this [shallot.Eurl](#)).

Examples: `"/foo/bar/baz"` for `smb://livingroom-pc/foo/bar/baz`. `"/foo/bar/baz"` for `file:///foo/bar/baz`. `"/"` for `file:///`.

#### 1.23.2.15 root()

```
def shallot.Eurl.root (
    self )
```

Returns the root [shallot.Eurl](#) from this one.

Example: `zip:[file:///a/b/c.zip]//` for `zip:[file:///a/b/c.zip]//foo/bar/baz`.

#### 1.23.2.16 scheme()

```
def shallot.Eurl.scheme (
    self )
```

Returns the scheme (from the outer url of this [shallot.Eurl](#)).

This is what comes before the `://`. Example: `"file"` for `file:///foo/bar/baz`.

#### 1.23.2.17 with\_appended\_segment()

```
def shallot.Eurl.with_appended_segment (
    self,
    segment )
```

Returns a new [shallot.Eurl](#) from this one with `"/basename"` appended.

The parameter is assumed to be a single path segment.

##### Parameters

<i>segment</i>	The basename to be appended.
----------------	------------------------------

#### 1.23.2.18 with\_appended\_segments()

```
def shallot.Eurl.with_appended_segments (
    self,
    path )
```

Returns a new [shallot.Eurl](#) with path segments `"/pa/t/h/"` appended.

The parameter may contain `" / "s` for dividing path segments.

#### Parameters

<i>path</i>	The path segments to be appended, like <code>"foo/bar"</code> .
-------------	---

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.24 shallot.Exceptions Class Reference

Everything about [exceptions](#).

### Classes

- class [ArgumentException](#)  
*Shallot exception for failed operation due to invalid arguments given to some program part.*
- class [IOException](#)  
*Shallot exception in IO.*
- class [ProgramException](#)  
*Shallot exception for program errors (typically logical stuff) in the program or plugin.*
- class [RuntimeException](#)  
*Shallot exception for failed operation due to (often external) runtime effects.*
- class [ScriptedException](#)  
*The Shallot exception class.*

### 1.24.1 Detailed Description

Everything about [exceptions](#).

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.25 shallot.Actions.ExecutionInfo Class Reference

An object for signalling action execution state changes between an action implementation and the Shallot core (mostly to the core).

## Public Member Functions

- `def __init__ (self)`  
*Can't be constructed directly.*
- `def set_visualprocessfeedback_active (self, v)`  
*Sets the visibility of a visual feedback.*
- `def is_visualprocessfeedback_active (self)`  
*Is visual feedback visibility enforced?*
- `def set_manual_intervention_needed (self, v)`  
*Set if manual intervention is needed.*
- `def is_manual_intervention_needed (self)`  
*Is manual intervention needed?*
- `def set_details (self, fromverb, fromobjectname, toverb, toobjectname)`  
*Sets the progress details.*
- `def from_verb (self)`  
*Gets the current verb on from-side.*
- `def from_objectname (self)`  
*Gets the current object name on from-side.*
- `def to_verb (self)`  
*Gets the current verb on to-side.*
- `def to_objectname (self)`  
*Gets the current object name on to-side.*
- `def set_head (self, txt)`  
*Sets the head text.*
- `def head (self)`  
*Gets the head text.*
- `def set_progress (self, done, all, label)`  
*Sets a current determinate progress.*
- `def set_progress_indeterminate (self, label)`  
*Sets a current indeterminate progress.*
- `def progress_done (self)`  
*How many items are done?*
- `def progress_all (self)`  
*How many items are to do in sum?*
- `def progress_text (self)`  
*The textual representation of the progress.*
- `def is_cancelled (self)`  
*Is the action cancelled?*
- `def cancel (self)`  
*Cancel the action.*
- `def respect_cancel (self)`  
*Respect a cancel request.*
- `def operation (self)`  
*Gets the [shallot.Operations.Operation](#) transactional operation object.*
- `def add_changed_eurl (self, eurl)`  
*Requests updating the filesystem model info for an item.*
- `def userfeedback (self)`  
*The [shallot.Actions.ExecutionUserFeedback](#) user feedback object.*

### 1.25.1 Detailed Description

An object for signalling action execution state changes between an action implementation and the Shallot core (mostly to the core).

Most calls lead to changes in the information presented by the progress dialog.

### 1.25.2 Member Function Documentation

#### 1.25.2.1 add\_changed\_url()

```
def shallot.Actions.ExecutionInfo.add_changed_url (
    self,
    url )
```

Requests updating the filesystem model info for an item.

##### Parameters

<i>url</i>	The item location to be updated as <a href="#">shallot.Eurl</a> .
------------	---

#### 1.25.2.2 respect\_cancel()

```
def shallot.Actions.ExecutionInfo.respect_cancel (
    self )
```

Respect a cancel request.

This should be called from time to time (inside loops for example).

#### 1.25.2.3 set\_details()

```
def shallot.Actions.ExecutionInfo.set_details (
    self,
    fromverb,
    fromobjectname,
    toverb,
    toobjectname )
```

Sets the progress details.

##### Parameters

<i>fromverb</i>	The verb on from-side (the source).
<i>fromobjectname</i>	(file path) The objectname on from-side (the source).
<i>toverb</i>	The verb on to-side (the destination).
<i>toobjectname</i>	(file path) The objectname on to-side (the destination).

#### 1.25.2.4 `set_progress()`

```
def shallot.Actions.ExecutionInfo.set_progress (
    self,
    done,
    all,
    label )
```

Sets a current determinate progress.

##### Parameters

<i>done</i>	The number of finished items.
<i>all</i>	The gross number of items.
<i>label</i>	Progress description text.

#### 1.25.2.5 `set_progress_indeterminate()`

```
def shallot.Actions.ExecutionInfo.set_progress_indeterminate (
    self,
    label )
```

Sets a current indeterminate progress.

##### Parameters

<i>label</i>	Progress description text.
--------------	----------------------------

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.26 `shallot.Actions.ExecutionUserFeedback` Class Reference

An object for communication with the user in action implementations.

### Classes

- class [MessageBoxButton](#)  
Buttons in a message box from [shallot.Actions.ExecutionUserFeedback](#).



## Public Member Functions

- `def __init__ (self)`  
*Can't be constructed directly.*
- `def simple_messagebox (self, message, buttons, icon, defaultbutton, abortbutton)`  
*A simple message box.*
- `def messagebox (self, message, buttons, icon, defaultbutton, abortbutton)`  
*A message box.*
- `def simple_inputbox (self, question, defaulttext)`  
*A simple input box.*

### 1.26.1 Detailed Description

An object for communication with the user in action implementations.

It can be used to ask the user for some information or to just give some information to the user.

### 1.26.2 Member Function Documentation

#### 1.26.2.1 messagebox()

```
def shallot.Actions.ExecutionUserFeedback.messagebox (
    self,
    message,
    buttons,
    icon,
    defaultbutton,
    abortbutton )
```

A message box.

Returns the index of the chosen button.

#### Parameters

<i>message</i>	The message text.
<i>buttons</i>	A list of buttons as string list.
<i>icon</i>	An icon name.
<i>defaultbutton</i>	Answer for keyboard Enter as <a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a> .
<i>abortbutton</i>	Answer for keyboard Esc as <a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a> .

#### 1.26.2.2 simple\_inputbox()

```
def shallot.Actions.ExecutionUserFeedback.simple_inputbox (
    self,
```

```
question,
defaulttext )
```

A simple input box.

#### Parameters

<i>question</i>	The question.
<i>defaulttext</i>	The default answer.

#### 1.26.2.3 simple\_messagebox()

```
def shallot.Actions.ExecutionUserFeedback.simple_messagebox (
    self,
    message,
    buttons,
    icon,
    defaultbutton,
    abortbutton )
```

A simple message box.

Returns the chosen button.

#### Parameters

<i>message</i>	The message text.
<i>buttons</i>	A list of buttons as or-combination of <a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a> .
<i>icon</i>	An icon name.
<i>defaultbutton</i>	Answer for keyboard Enter as <a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a> . May be 0 for no definition.
<i>abortbutton</i>	Answer for keyboard Esc as <a href="#">shallot.Actions.ExecutionUserFeedback.MessageBoxButton</a> . May be 0 for no definition.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.27 shallot.FilePropertyDialog Class Reference

The [file property dialog](#).

#### Classes

- class [Tab](#)  
*Abstract base class for a tab in the Properties dialog.*
- class [TabPropertyIconTextBanner](#)  
*Represents values for image/text banners as used for [shallot.FilePropertyDialog.Tab.PropertyType.IconTextBanner](#).*

## Public Attributes

- **tabclass**

### 1.27.1 Detailed Description

The [file property dialog](#).

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.28 shallot.FileSearch Class Reference

File [searches](#).

### Classes

- class [SearchCriterion](#)  
*Abstract base class for a search criterion.*
- class [SearchCriterionFactory](#)  
*A factory for a [shallot.FileSearch.SearchCriterion](#) class.*

### 1.28.1 Detailed Description

File [searches](#).

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.29 shallot.Filesystem Class Reference

The [filesystem](#).

### Classes

- class [Handler](#)  
*Abstract base class for a custom filesystem handler.*
- class [Node](#)  
*A model representation of a location in the filesystem tree.*
- class [NodeList](#)  
*A list editor for a filesystem node list.*
- class [NodeType](#)  
*Enumeration of types a [shallot.Filesystem.Node](#) can have.*

## Static Public Member Functions

- def [find\\_nodes\\_for\\_eurl](#) (eurl)  
*Get a list of [shallot.Filesystem.Node](#) for a [shallot.Eurl](#).*
- def [try\\_get\\_nodes\\_for\\_eurl](#) (eurl)  
*Returns a list of [shallot.Filesystem.Node](#) for a [shallot.Eurl](#).*
- def [refresh](#) (eurl, forceFindParent=False, withDetails=True)  
*Request to refresh the internal model information for a [shallot.Eurl](#).*
- def [get\\_or\\_create\\_node](#) (eurl, handler, nodetype, parentnode, doinsert=True, showInitialLoadingLabel=True)  
*Returns a [shallot.Filesystem.Node](#) for using it as a child node in the model.*
- def [create\\_node](#) (eurl, handler, nodetype, parentnode, doinsert=True, showInitialLoadingLabel=True)  
*Creates a new [shallot.Filesystem.Node](#) for placing it into the model.*

## Static Public Attributes

- [rootnode](#) = `internalstuff.modelRootNode()`  
*The [shallot.Filesystem.Node](#) which is the root node of the entire model.*

### 1.29.1 Detailed Description

The [filesystem](#).

### 1.29.2 Member Function Documentation

#### 1.29.2.1 `create_node()`

```
def shallot.Filesystem.create_node (
    eurl,
    handler,
    nodetype,
    parentnode,
    doinsert = True,
    showInitialLoadingLabel = True ) [static]
```

Creates a new [shallot.Filesystem.Node](#) for placing it into the model.

If such a node (with the same eurl for the same parent node) does not exist, it generates a new one. If there already is such a node alive, but currently not placed in the model, it recycles this one. This can happen when references exist to a node which is not yet inserted or which is removed meanwhile. It is not allowed to call this method when such a node already exists in the model. Use this function for getting a [shallot.Filesystem.Node](#), which is to be added to the model now or later. It is typically used within a [shallot.Filesystem.Handler](#) implementation.

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the new item.	
<i>handler</i>	The <a href="#">shallot.Filesystem.Handler</a> of the new node. This must always be the same one as the handler registered for the scheme of the eurl!	
<i>nodetype</i>	The <a href="#">shallot.Filesystem.NodeType</a> node type.	
<i>parentnode</i>	The <a href="#">shallot.Filesystem.Node</a> parent node.	Generated by Doxygen
<i>doinsert</i>	If this function shall actually add the node to the model (to be exact, in some situations, this will not actually take place nonetheless).	
<i>showInitialLoadingLabel</i>	If a 'loading...' label shall be visible at beginning.	

## 1.29.2.2 find\_nodes\_for\_eurl()

```
def shallot.Filesystem.find_nodes_for_eurl (
    eurl ) [static]
```

Get a list of [shallot.Filesystem.Node](#) for a [shallot.Eurl](#).

If it is unknown to the model so far, it tries to build them. In typical cases, this list either contains one element, or is empty if the filesystem handlers decide that this file does not exist. But in some cases, there is also more than one node for one [shallot.Eurl](#) (when the [Eurl](#) appears on more than one place in the tree). It only returns nodes, which are 'alive', i.e. which have a living parent and which are a child of this parent.

## 1.29.2.3 get\_or\_create\_node()

```
def shallot.Filesystem.get_or_create_node (
    eurl,
    handler,
    nodetype,
    parentnode,
    doinsert = True,
    showInitialLoadingLabel = True ) [static]
```

Returns a [shallot.Filesystem.Node](#) for using it as a child node in the model.

It either creates a new one, if there currently is no node for this eurl in this parentnode, or returns the existing one. Even for existing ones, this call can change the nodetype of that node.

## Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the new node.
<i>handler</i>	The <a href="#">shallot.Filesystem.Handler</a> of the new node. This must always be the same one as the handler registered for the scheme of the eurl!
<i>nodetype</i>	The <a href="#">shallot.Filesystem.NodeType</a> node type.
<i>parentnode</i>	The <a href="#">shallot.Filesystem.Node</a> parent node.
<i>doinsert</i>	If this function shall actually add the node to the model (to be exact, in some situations, this will not actually take place nonetheless).
<i>showInitialLoadingLabel</i>	If a 'loading...' label shall be visible at beginning.

## 1.29.2.4 refresh()

```
def shallot.Filesystem.refresh (
    eurl,
    forceFindParent = False,
    withDetails = True ) [static]
```

Request to refresh the internal model information for a [shallot.Eurl](#).

This may be a place which is already known (then a change of some metadata or the removal is detected) or a formerly unknown place (then new nodes get inserted in the model).

## Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> of the node.
<i>forceFindParent</i>	Forcefully create such model information if unknown before.
<i>withDetails</i>	If detail columns assigned to an existing node should also be updated.

## 1.29.2.5 try\_get\_nodes\_for\_eurl()

```
def shallot.Filesystem.try_get_nodes_for_eurl (
    url ) [static]
```

Returns a list of [shallot.Filesystem.Node](#) for a [shallot.Eurl](#).

It only considers the current state of the in-memory model. It will only return nodes which are already known to the model so far. This operation is cheaper and handling only the known nodes is sufficient in many situations. In typical cases, this list either contains one element, or is empty. But in some cases, there is also more than one node for one [shallot.Eurl](#) (when the [Eurl](#) appears on more than one place in the tree). It only returns nodes, which are 'alive', i.e. which have a living parent and which are a child of this parent.

## 1.29.3 Member Data Documentation

## 1.29.3.1 rootnode

```
shallot.Filesystem.rootnode = internalstuff.modelRootNode() [static]
```

The [shallot.Filesystem.Node](#) which is the root node of the entire model.

It is the parent for all toplevel nodes.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.30 shallot.Operations.FilesystemOperation Class Reference

A high-level provider of filesystem operations.

## Public Member Functions

- def `__init__` (self)  
*Can't be constructed directly.*
- def `itemlist` (self, eurl)  
*Gets a list of [shallot.Eurl](#) in a directory.*
- def `itemlist_by_type` (self, eurl, nodetype)  
*Gets a list of [shallot.Eurl](#) in a directory which have a certain type.*
- def `get_type` (self, eurl)  
*Gets the [shallot.Filesystem.NodeType](#) node type for an entry.*
- def `get_linktarget` (self, eurl)  
*Gets the link target for an entry (if it is a link).*
- def `can_get_filecontent` (self, eurl)  
*Can we get the file content for an entry?*
- def `get_file_content` (self, eurl)  
*Gets the file content for an entry as [shallot.Streaming.ReadDataDevice](#).*
- def `can_create_directory` (self, eurl)  
*Can we create a given directory?*
- def `create_directory` (self, eurl, progressmon)  
*Creates a directory.*
- def `can_create_link` (self, eurl)  
*Can we create a given link?*
- def `create_link` (self, eurl, tgt, progressmon)  
*Creates a link.*
- def `can_create_file` (self, eurl)  
*Can we create a given file?*
- def `create_file` (self, eurl, contentdevice, progressmon)  
*Create a file.*
- def `can_delete_item` (self, eurl)  
*Can we delete a given entry?*
- def `delete_item` (self, eurl, progressmon)  
*Delete an entry.*
- def `delete_directory_if_empty` (self, eurl, progressmon)  
*Delete a directory entry if empty.*
- def `can_move_item` (self, src, dest)  
*Can we move a given entry?*
- def `move_items` (self, src, tgt, progressmon)  
*Move entries.*
- def `move_item` (self, src, tgt, progressmon)  
*Move an entry.*
- def `can_copy_item` (self, src, dest)  
*Can we copy a given entry?*
- def `copy_items` (self, src, tgt, progressmon)  
*Copy entries.*
- def `copy_item` (self, src, tgt, progressmon)  
*Copy an entry.*
- def `list_extendedattributes` (self, eurl)  
*Lists all keys of extended attributes stored in the filesystem for an entry.*
- def `get_extendedattribute_size` (self, eurl, attribute)  
*Returns the size of an extended attribute value stored in the filesystem for an entry.*
- def `get_extendedattribute` (self, eurl, attribute)



- Returns the value (as byte string) of an extended attribute value stored in the filesystem for an entry.*
- def [set\\_extendedattribute](#) (self, eurl, attribute, value)  
*Store the value of an extended attribute in the filesystem for an entry.*
- def [remove\\_extendedattribute](#) (self, eurl, attribute)  
*Removes an extended attribute stored in the filesystem for an entry.*
- def [resolve\\_node\\_link](#) (self, node)  
*Resolve a link as node.*
- def [resolve\\_node\\_link\\_nonrecursive](#) (self, node)  
*Resolve a link as node without recursion.*
- def [resolve\\_eurl\\_link](#) (self, eurl)  
*Resolve a link as [Eurl](#).*
- def [resolve\\_eurl\\_link\\_nonrecursive](#) (self, eurl)  
*Resolve a link as [Eurl](#) without recursion.*

### 1.30.1 Detailed Description

A high-level provider of filesystem operations.

It is always based on the transaction of a [shallot.Operations.Operation](#) and also accessible from such an instance.

Some operations allow to specify an [shallot.Operations.FilesystemOperationProgressMonitor](#). You can specify an instance for getting some additional functionality, like progress notification, conflict resolution and more. Please note that not each operation uses each aspect of [shallot.Operations.FilesystemOperationProgressMonitor](#) (e.g. some will not do any conflict resolution).

### 1.30.2 Member Function Documentation

#### 1.30.2.1 can\_copy\_item()

```
def shallot.Operations.FilesystemOperation.can_copy_item (
    self,
    src,
    dest )
```

Can we copy a given entry?

Parameters

<i>src</i>	The <a href="#">shallot.Eurl</a> of the item to be copied.
<i>dest</i>	The <a href="#">shallot.Eurl</a> destination path. If it is not known, use <code>None</code> .

#### 1.30.2.2 can\_create\_directory()

```
def shallot.Operations.FilesystemOperation.can_create_directory (
```

```
self,  
eurl )
```

Can we create a given directory?

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.3 can\_create\_file()

```
def shallot.Operations.FilesystemOperation.can_create_file (  
    self,  
    eurl )
```

Can we create a given file?

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.4 can\_create\_link()

```
def shallot.Operations.FilesystemOperation.can_create_link (  
    self,  
    eurl )
```

Can we create a given link?

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.5 can\_delete\_item()

```
def shallot.Operations.FilesystemOperation.can_delete_item (  
    self,  
    eurl )
```

Can we delete a given entry?

## Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> of the considered item.
------------	--

## 1.30.2.6 can\_get\_filecontent()

```
def shallot.Operations.FilesystemOperation.can_get_filecontent (
    self,
    url )
```

Can we get the file content for an entry?

## Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> of the considered item.
------------	--

## 1.30.2.7 can\_move\_item()

```
def shallot.Operations.FilesystemOperation.can_move_item (
    self,
    src,
    dest )
```

Can we move a given entry?

## Parameters

<i>src</i>	The <a href="#">shallot.Eurl</a> of the item to be moved.
<i>dest</i>	The <a href="#">shallot.Eurl</a> destination path. If it is not known, use <code>None</code> .

## 1.30.2.8 copy\_item()

```
def shallot.Operations.FilesystemOperation.copy_item (
    self,
    src,
    tgt,
    progressmon )
```

Copy an entry.

## Parameters

<i>src</i>	The <a href="#">shallot.Eurl</a> of the item to be copied.
------------	--

## Parameters

<i>tgt</i>	The <a href="#">shallot.Eurl</a> of the new destination.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

1.30.2.9 `copy_items()`

```
def shallot.Operations.FilesystemOperation.copy_items (
    self,
    src,
    tgt,
    progressmon )
```

Copy entries.

## Parameters

<i>src</i>	List of <a href="#">shallot.Eurl</a> of the item to be copied.
<i>tgt</i>	The <a href="#">shallot.Eurl</a> of the new common parent destination.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

1.30.2.10 `create_directory()`

```
def shallot.Operations.FilesystemOperation.create_directory (
    self,
    eurl,
    progressmon )
```

Creates a directory.

## Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the new directory.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

1.30.2.11 `create_file()`

```
def shallot.Operations.FilesystemOperation.create_file (
    self,
```

```

    eurl,
    contentdevice,
    progressmon )

```

Create a file.

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the new file.
<i>contentdevice</i>	a <a href="#">shallot.Streaming.ReadDataDevice</a> content device.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

#### 1.30.2.12 create\_link()

```

def shallot.Operations.FilesystemOperation.create_link (
    self,
    eurl,
    tgt,
    progressmon )

```

Creates a link.

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the new link.
<i>tgt</i>	Target path of this link.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

#### 1.30.2.13 delete\_directory\_if\_empty()

```

def shallot.Operations.FilesystemOperation.delete_directory_if_empty (
    self,
    eurl,
    progressmon )

```

Delete a directory entry if empty.

#### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the directory to delete.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

#### 1.30.2.14 delete\_item()

```
def shallot.Operations.FilesystemOperation.delete_item (
    self,
    eurl,
    progressmon )
```

Delete an entry.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the item to delete.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

#### 1.30.2.15 get\_extendedattribute()

```
def shallot.Operations.FilesystemOperation.get_extendedattribute (
    self,
    eurl,
    attribute )
```

Returns the value (as byte string) of an extended attribute value stored in the filesystem for an entry.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
<i>attribute</i>	The attribute key.

#### 1.30.2.16 get\_extendedattribute\_size()

```
def shallot.Operations.FilesystemOperation.get_extendedattribute_size (
    self,
    eurl,
    attribute )
```

Returns the size of an extended attribute value stored in the filesystem for an entry.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
<i>attribute</i>	The attribute key.

#### 1.30.2.17 get\_file\_content()

```
def shallot.Operations.FilesystemOperation.get_file_content (
    self,
    eurl )
```

Gets the file content for an entry as [shallot.Streaming.ReadDataDevice](#).

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.18 get\_linktarget()

```
def shallot.Operations.FilesystemOperation.get_linktarget (
    self,
    eurl )
```

Gets the link target for an entry (if it is a link).

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.19 get\_type()

```
def shallot.Operations.FilesystemOperation.get_type (
    self,
    eurl )
```

Gets the [shallot.Filesystem.NodeType](#) node type for an entry.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
-------------	--

#### 1.30.2.20 itemlist()

```
def shallot.Operations.FilesystemOperation.itemlist (
    self,
    eurl )
```

Gets a list of [shallot.Eurl](#) in a directory.

**Parameters**

<i>url</i>	The <a href="#">shallot.Eurl</a> of the directory to list.
------------	--

**1.30.2.21 itemlist\_by\_type()**

```
def shallot.Operations.FilesystemOperation.itemlist_by_type (
    self,
    url,
    nodetype )
```

Gets a list of [shallot.Eurl](#) in a directory which have a certain type.

**Parameters**

<i>url</i>	The <a href="#">shallot.Eurl</a> of the directory to list.
<i>nodetype</i>	The <a href="#">shallot.Filesystem.NodeType</a> node type to fetch.

**1.30.2.22 list\_extendedattributes()**

```
def shallot.Operations.FilesystemOperation.list_extendedattributes (
    self,
    url )
```

Lists all keys of extended attributes stored in the filesystem for an entry.

**Parameters**

<i>url</i>	The <a href="#">shallot.Eurl</a> of the considered item.
------------	--

**1.30.2.23 move\_item()**

```
def shallot.Operations.FilesystemOperation.move_item (
    self,
    src,
    tgt,
    progressmon )
```

Move an entry.

**Parameters**

<i>src</i>	The <a href="#">shallot.Eurl</a> of the item to be moved.
------------	---



## Parameters

<i>tgt</i>	The <a href="#">shallot.Eurl</a> of the new destination.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

## 1.30.2.24 move\_items()

```
def shallot.Operations.FilesystemOperation.move_items (
    self,
    src,
    tgt,
    progressmon )
```

Move entries.

## Parameters

<i>src</i>	List of <a href="#">shallot.Eurl</a> of the item to be moved.
<i>tgt</i>	The <a href="#">shallot.Eurl</a> of the new common parent destination.
<i>progressmon</i>	A <a href="#">shallot.Operations.FilesystemOperationProgressMonitor</a> instance for some additional functionality (or <code>None</code> ). See <a href="#">shallot.Operations.FilesystemOperation</a> for details.

## 1.30.2.25 remove\_extendedattribute()

```
def shallot.Operations.FilesystemOperation.remove_extendedattribute (
    self,
    eurl,
    attribute )
```

Removes an extended attribute stored in the filesystem for an entry.

## Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
<i>attribute</i>	The attribute key.

## 1.30.2.26 resolve\_eurl\_link()

```
def shallot.Operations.FilesystemOperation.resolve_eurl_link (
    self,
    eurl )
```

Resolve a link as [Eurl](#).

**Parameters**

<i>url</i>	A <a href="#">shallot.Eurl</a> to resolve.
------------	--

**1.30.2.27 resolve\_eurl\_link\_nonrecursive()**

```
def shallot.Operations.FilesystemOperation.resolve_eurl_link_nonrecursive (
    self,
    url )
```

Resolve a link as [Eurl](#) without recursion.

**Parameters**

<i>url</i>	A <a href="#">shallot.Eurl</a> to resolve.
------------	--

**1.30.2.28 resolve\_node\_link()**

```
def shallot.Operations.FilesystemOperation.resolve_node_link (
    self,
    node )
```

Resolve a link as node.

**Parameters**

<i>node</i>	A <a href="#">shallot.Filesystem.Node</a> to resolve.
-------------	---

**1.30.2.29 resolve\_node\_link\_nonrecursive()**

```
def shallot.Operations.FilesystemOperation.resolve_node_link_nonrecursive (
    self,
    node )
```

Resolve a link as node without recursion.

**Parameters**

<i>node</i>	A <a href="#">shallot.Filesystem.Node</a> to resolve.
-------------	---

## 1.30.2.30 set\_extendedattribute()

```
def shallot.Operations.FilesystemOperation.set_extendedattribute (
    self,
    eurl,
    attribute,
    value )
```

Store the value of an extended attribute in the filesystem for an entry.

## Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> of the considered item.
<i>attribute</i>	The attribute key.
<i>value</i>	The attribute value (as byte string).

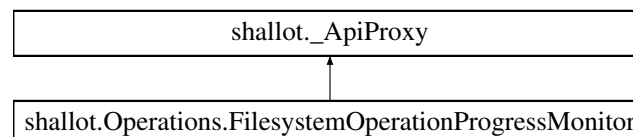
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.31 shallot.Operations.FilesystemOperationProgressMonitor Class Reference

Abstract base class for progress monitors, used for monitoring progress of some operations in [shallot.Operations.FilesystemOperation](#) and for conflict resolution.

Inheritance diagram for shallot.Operations.FilesystemOperationProgressMonitor:



## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, actionExecution=None)
- def [changed](#) (self)  
*Reacts on progress changes.*
- def [resolve\\_conflicts](#) (self, steps)  
*Resolves upcoming filesystem conflicts.*
- def [has\\_item\\_info](#) (self)  
*If the monitor currently has information about progress in terms of item counts.*
- def [done\\_items](#) (self)  
*How many items are transferred so far.*
- def [all\\_items](#) (self)  
*How many items are to be transferred in total.*
- def [has\\_bytes\\_info](#) (self)  
*If the monitor currently has information about progress in terms of transferred bytes.*
- def [done\\_bytes](#) (self)

- *How many bytes are transferred so far.*  
• def `all_bytes` (self)
- *How many bytes are to be transferred in total.*  
• def `get_item_info_from` (self)
- *The current source as textual representation.*  
• def `get_item_info_to` (self)
- *The current destination as textual representation.*  
• def `estimation` (self)
- *The current performance estimation as textual representation.*

### 1.31.1 Detailed Description

Abstract base class for progress monitors, used for monitoring progress of some operations in [shallot.Operations](#).↔ [FilesystemOperation](#) and for conflict resolution.

It makes sense to directly instantiate this class in some cases.

### 1.31.2 Constructor & Destructor Documentation

#### 1.31.2.1 `__init__()`

```
def shallot.Operations.FilesystemOperationProgressMonitor.__init__ (
    self,
    actionExecution = None )
```

##### Parameters

<i>actionExecution</i>	Optional instance of <a href="#">shallot.Actions.ExecutionInfo</a> . If available, it provides some additional features (e.g. the user may cancel the transfer).
------------------------	--

### 1.31.3 Member Function Documentation

#### 1.31.3.1 `changed()`

```
def shallot.Operations.FilesystemOperationProgressMonitor.changed (
    self )
```

Reacts on progress changes.

Override this method in custom subclasses or leave the default implementation.

## 1.31.3.2 resolve\_conflicts()

```
def shallot.Operations.FilesystemOperationProgressMonitor.resolve_conflicts (
    self,
    steps )
```

Resolves upcoming filesystem conflicts.

Override this method in custom subclasses or leave the default implementation.

## Parameters

<code>steps</code>	A list of <a href="#">shallot.Operations.FilesystemOperationStep</a> which stay in conflict.
--------------------	--

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.32 shallot.Operations.FilesystemOperationStep Class Reference

One step in a larger filesystem transfer running inside some operations in [shallot.Operations.FilesystemOperation](#).

## Classes

- class [ConflictResolution](#)  
*Enumeration of conflict resolution strategies.*

## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)  
*Can't be constructed directly.*
- def [conflict\\_description](#) (self)  
*The conflict description text.*
- def [conflict\\_resolution\\_rename\\_destination\\_before\\_to](#) (self)  
*The new destination name (if conflict\_resolution is [shallot.Operations.FilesystemOperationStep.ConflictResolution.RenameDestinationBefore](#))*
- def [conflict\\_resolution\\_different\\_destination\\_name\\_to](#) (self)  
*The new destination name (if conflict\_resolution is [shallot.Operations.FilesystemOperationStep.ConflictResolution.UseDifferentDestinationName](#))*
- def [set\\_conflict\\_resolve\\_skip](#) (self)  
*Set the conflict resolution to [shallot.Operations.FilesystemOperationStep.ConflictResolution.Skip](#).*
- def [set\\_conflict\\_resolve\\_overwrite\\_destination](#) (self)  
*Set the conflict resolution to [shallot.Operations.FilesystemOperationStep.ConflictResolution.OverwriteDestination](#).*
- def [set\\_conflict\\_resolve\\_rename\\_destination\\_before](#) (self, newname)  
*Set the conflict resolution to [shallot.Operations.FilesystemOperationStep.ConflictResolution.RenameDestinationBefore](#).*

- `def set_conflict_resolve_use_different_destination_name (self, newname)`  
*Set the conflict resolution to `shallot.Operations.FilesystemOperationStep.ConflictResolution.UseDifferentDestinationName`.*
- `def set_conflict_resolve_merge_directories (self)`  
*Set the conflict resolution to `shallot.Operations.FilesystemOperationStep.ConflictResolution.MergeDirectories`.*
- `def sourcetype (self)`  
*The `shallot.Filesystem.NodeType` of the source.*
- `def source (self)`  
*The source `shallot.Eurl` location.*
- `def original_destination (self)`  
*The original destination `shallot.Eurl` location.*
- `def effective_destination (self)`  
*The effective destination `shallot.Eurl` location with applied conflict resolution strategies.*
- `def set_conflict_resolve_merge_directories_check_allowed (self)`  
*Checks if it is allowed to solve this conflict by merging directories.*
- `def conflict_resolution (self)`  
*The currently selected `shallot.Operations.FilesystemOperationStep.ConflictResolution` conflict resolution.*

### 1.32.1 Detailed Description

One step in a larger filesystem transfer running inside some operations in `shallot.Operations.FilesystemOperation`.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.33 shallot.Setting.GroupInfo Class Reference

Enumeration of groups to which a setting can belong.

### Static Public Attributes

- `Global` = `internalstuff.settinggroup_global()`  
*Global aspects group.*
- `GUI` = `internalstuff.settinggroup_gui()`  
*User interface aspects group.*
- `FileHandling` = `internalstuff.settinggroup_filehandling()`  
*File handling aspects group.*
- `DataView` = `internalstuff.settinggroup_dataview()`  
*Data view aspects group.*
- `Behavior` = `internalstuff.settinggroup_behavior()`  
*Behavioral aspects group.*
- `Special` = `internalstuff.settinggroup_special()`  
*Special aspects group.*

### 1.33.1 Detailed Description

Enumeration of groups to which a setting can belong.

This is just a matter of grouping the for presentation.

### 1.33.2 Member Data Documentation

#### 1.33.2.1 Behavior

```
shallot.Setting.GroupInfo.Behavior = internalstuff.settinggroup_behavior() [static]
```

Behavioral aspects group.

#### 1.33.2.2 DataView

```
shallot.Setting.GroupInfo.DataView = internalstuff.settinggroup_dataview() [static]
```

Data view aspects group.

#### 1.33.2.3 FileHandling

```
shallot.Setting.GroupInfo.FileHandling = internalstuff.settinggroup_filehandling() [static]
```

File handling aspects group.

#### 1.33.2.4 Global

```
shallot.Setting.GroupInfo.Global = internalstuff.settinggroup_global() [static]
```

Global aspects group.

#### 1.33.2.5 GUI

```
shallot.Setting.GroupInfo.GUI = internalstuff.settinggroup_gui() [static]
```

User interface aspects group.

### 1.33.2.6 Special

```
shallot.Setting.GroupInfo.Special = internalstuff.settinggroup_special() [static]
```

Special aspects group.

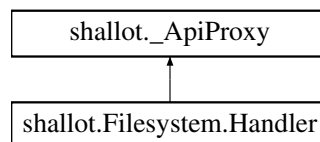
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.34 shallot.Filesystem.Handler Class Reference

Abstract base class for a custom filesystem handler.

Inheritance diagram for shallot.Filesystem.Handler:



### Public Member Functions

- def `__init__`(self)
- def `configure_item`(self, operation, node)  
*Configure a newly created node (e.g.*
- def `list_extendedattributes`(self, operation, eurl)  
*Returns a list of extended attributes which exist for a location.*
- def `get_extendedattribute_size`(self, operation, eurl, attribname)  
*Returns the size (in byte) of the value for a particular extended attribute.*
- def `get_extendedattribute`(self, operation, eurl, attribname)  
*Gets the value for a particular extended attribute.*
- def `set_extendedattribute`(self, operation, eurl, attribname, value)  
*Sets the value for a particular extended attribute.*
- def `remove_extendedattribute`(self, operation, eurl, attribname)  
*Removes an extended attribute.*
- def `get_customattributes`(self, operation, eurl)  
*Returns a dict<string,string> with custom attributes for a node.*
- def `set_customattribute`(self, operation, eurl, key, value)  
*Sets a custom attribute to an entry.*
- def `get_mimetype`(self, operation, eurl)  
*Determine mime type of a file.*
- def `get_type`(self, operation, eurl)  
*Determine if a file is regular, or a dir, or a link, ...*
- def `get_size`(self, operation, eurl)  
*Determine the size of a file.*
- def `get_mtime`(self, operation, eurl)  
*Determine the mtime of a file.*



- def [set\\_mtime](#) (self, operation, eurl, mtime)  
*Set the mtime of a file.*
- def [can\\_create\\_directory](#) (self, operation, eurl)  
*Is it allowed to create subdirectories in a certain directory?*
- def [create\\_directory](#) (self, operation, eurl)  
*Create a directory.*
- def [can\\_create\\_file](#) (self, operation, eurl)  
*Is it allowed to create files in a certain directory?*
- def [can\\_delete\\_item](#) (self, operation, eurl)  
*Is it allowed to delete a certain file/directory/link/...?*
- def [delete\\_item](#) (self, operation, eurl)  
*Delete a file/directory/link/...*
- def [can\\_rename\\_item](#) (self, operation, src)  
*Is it allowed to move a certain file/directory/link/...?*
- def [rename\\_item](#) (self, operation, src, destpath)  
*Move a file/directory/link/...*
- def [itemlist](#) (self, operation, eurl, nodetype, nodelist)  
*Determine a list of subelements in a certain directory.*
- def [can\\_get\\_filecontent](#) (self, operation, eurl)  
*Is it allowed to get the content of a certain file?*
- def [get\\_file\\_content](#) (self, operation, eurl)  
*Get the content of a file as a [shallot.Streaming.ReadDataDevice](#).*
- def [create\\_file](#) (self, operation, eurl, outstream, handlertransfer)  
*Creates a file with some content.*
- def [get\\_actions](#) (self, eurls)  
*Which actions (see former example) are to be offered for certain files?*
- def [can\\_create\\_link](#) (self, operation, eurl)  
*Is it allowed to create a link in a certain directory?*
- def [create\\_link](#) (self, operation, eurl, tgt)  
*Create a link.*
- def [get\\_linktarget](#) (self, operation, eurl)  
*Resolve a link.*

### Static Public Member Functions

- def [register](#) (scheme, handler)  
*Registers a [shallot.Filesystem.Handler](#) filesystem handler implementation.*

#### 1.34.1 Detailed Description

Abstract base class for a custom filesystem handler.

Subclass and register it for implementing a new virtual filesystem, which lets new nodes appear somewhere in the filesystem tree and controls how to handle them. Use [shallot.Filesystem.Handler.register](#) for registration.

#### 1.34.2 Member Function Documentation

#### 1.34.2.1 `can_create_directory()`

```
def shallot.Filesystem.Handler.can_create_directory (
    self,
    operation,
    eurl )
```

Is it allowed to create subdirectories in a certain directory?

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.2 `can_create_file()`

```
def shallot.Filesystem.Handler.can_create_file (
    self,
    operation,
    eurl )
```

Is it allowed to create files in a certain directory?

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.3 `can_create_link()`

```
def shallot.Filesystem.Handler.can_create_link (
    self,
    operation,
    eurl )
```

Is it allowed to create a link in a certain directory?

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

## 1.34.2.4 can\_delete\_item()

```
def shallot.Filesystem.Handler.can_delete_item (
    self,
    operation,
    eurl )
```

Is it allowed to delete a certain file/directory/link/...?

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

## 1.34.2.5 can\_get\_filecontent()

```
def shallot.Filesystem.Handler.can_get_filecontent (
    self,
    operation,
    eurl )
```

Is it allowed to get the content of a certain file?

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.6 can\_rename\_item()

```
def shallot.Filesystem.Handler.can_rename_item (
    self,
    operation,
    src )
```

Is it allowed to move a certain file/directory/link/...?

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>src</i>	The <a href="#">shallot.Eurl</a> address pointing to the potential item to be moved.

#### 1.34.2.7 configure\_item()

```
def shallot.Filesystem.Handler.configure_item (
    self,
    operation,
    node )
```

Configure a newly created node (e.g.

setting another icon or changing the display name).

Override this method in custom subclasses or leave the default implementation.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>node</i>	The <a href="#">shallot.Filesystem.Node</a> filesystem node to configure.

#### 1.34.2.8 create\_directory()

```
def shallot.Filesystem.Handler.create_directory (
    self,
    operation,
    eurl )
```

Create a directory.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

1.34.2.9 `create_file()`

```
def shallot.Filesystem.Handler.create_file (
    self,
    operation,
    eurl,
    outstream,
    handlertransfer )
```

Creates a file with some content.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>outstream</i>	the content stream to be transferred into the new file
<i>handlertransfer</i>	<a href="#">shallot.Operations.HandlerTransfer</a> may be used for some better integration, like cancel support and progress monitoring.

1.34.2.10 `create_link()`

```
def shallot.Filesystem.Handler.create_link (
    self,
    operation,
    eurl,
    tgt )
```

Create a link.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>tgt</i>	The link destination path (as string).

#### 1.34.2.11 delete\_item()

```
def shallot.Filesystem.Handler.delete_item (
    self,
    operation,
    eurl )
```

Delete a file/directory/link/...

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.12 get\_actions()

```
def shallot.Filesystem.Handler.get_actions (
    self,
    eurls )
```

Which actions (see former example) are to be offered for certain files?

Override this method in custom subclasses.

##### Parameters

<i>eurls</i>	A list of <a href="#">shallot.Eurl</a> addresses this call is referred to.
--------------	--

#### 1.34.2.13 get\_customattributes()

```
def shallot.Filesystem.Handler.get_customattributes (
    self,
    operation,
    eurl )
```

Returns a dict<string,string> with custom attributes for a node.

Override this method in custom subclasses or leave the default implementation. In contrast to extended attributes, those ones are not directly stored in the filesystems, but are managed in a handler specific way (e.g. file permissions).

#### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.14 `get_extendedattribute()`

```
def shallot.Filesystem.Handler.get_extendedattribute (
    self,
    operation,
    eurl,
    attribname )
```

Gets the value for a particular extended attribute.

Override this method in custom subclasses or leave the default implementation. Extended attributes are all kinds of properties of a file which aren't handled otherwise. It can contain filesystem's extended attributes, permission information and more.

#### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>attribname</i>	The attribute name.

#### 1.34.2.15 `get_extendedattribute_size()`

```
def shallot.Filesystem.Handler.get_extendedattribute_size (
    self,
    operation,
    eurl,
    attribname )
```

Returns the size (in byte) of the value for a particular extended attribute.

Override this method in custom subclasses or leave the default implementation. Extended attributes are all kinds of properties of a file which aren't handled otherwise. It can contain filesystem's extended attributes, permission information and more.



## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>attribname</i>	The attribute name.

1.34.2.16 `get_file_content()`

```
def shallot.Filesystem.Handler.get_file_content (
    self,
    operation,
    eurl )
```

Get the content of a file as a [shallot.Streaming.ReadDataDevice](#).

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

1.34.2.17 `get_linktarget()`

```
def shallot.Filesystem.Handler.get_linktarget (
    self,
    operation,
    eurl )
```

Resolve a link.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.18 `get_mimetype()`

```
def shallot.Filesystem.Handler.get_mimetype (
    self,
    operation,
    eurl )
```

Determine mime type of a file.

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.19 `get_mtime()`

```
def shallot.Filesystem.Handler.get_mtime (
    self,
    operation,
    eurl )
```

Determine the mtime of a file.

Override this method in custom subclasses.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.20 `get_size()`

```
def shallot.Filesystem.Handler.get_size (
    self,
    operation,
    eurl )
```

Determine the size of a file.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

1.34.2.21 `get_type()`

```
def shallot.Filesystem.Handler.get_type (
    self,
    operation,
    eurl )
```

Determine if a file is regular, or a dir, or a link, ...

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

1.34.2.22 `itemlist()`

```
def shallot.Filesystem.Handler.itemlist (
    self,
    operation,
    eurl,
    nodetype,
    nodelist )
```

Determine a list of subelements in a certain directory.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>nodetype</i>	Which kind of items are requested. See <a href="#">shallot.Filesystem.NodeType</a> .
<i>nodelist</i>	A <a href="#">shallot.Filesystem.NodeList</a> object which controls the entry list for the referred directory. Operate on this object for adding children.

#### 1.34.2.23 list\_extendedattributes()

```
def shallot.Filesystem.Handler.list_extendedattributes (
    self,
    operation,
    eurl )
```

Returns a list of extended attributes which exist for a location.

Override this method in custom subclasses or leave the default implementation. Extended attributes are all kinds of properties of a file which aren't handled otherwise. It can contain filesystem's extended attributes, permission information and more.

##### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.

#### 1.34.2.24 register()

```
def shallot.Filesystem.Handler.register (
    scheme,
    handler ) [static]
```

Registers a [shallot.Filesystem.Handler](#) filesystem handler implementation.

##### Parameters

<i>scheme</i>	The scheme name (first part in a <a href="#">shallot.Eurl</a> , like <i>file</i> ).
<i>handler</i>	The <a href="#">shallot.Filesystem.Handler</a> filesystem handler.

#### 1.34.2.25 remove\_extendedattribute()

```
def shallot.Filesystem.Handler.remove_extendedattribute (
    self,
    operation,
    eurl,
    attribname )
```

Removes an extended attribute.

Override this method in custom subclasses or leave the default implementation. Extended attributes are all kinds of properties of a file which aren't handled otherwise. It can contain filesystem's extended attributes, permission information and more.

**Parameters**

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>attribname</i>	The attribute name.

**1.34.2.26 rename\_item()**

```
def shallot.Filesystem.Handler.rename_item (
    self,
    operation,
    src,
    destpath )
```

Move a file/directory/link/...

to somewhere else (can be a simple renaming).

Override this method in custom subclasses.

**Parameters**

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>src</i>	The <a href="#">shallot.Eurl</a> address pointing to the item to be moved.
<i>destpath</i>	The new path for the src item.

**1.34.2.27 set\_customattribute()**

```
def shallot.Filesystem.Handler.set_customattribute (
    self,
    operation,
    eurl,
    key,
    value )
```

Sets a custom attribute to an entry.

Override this method in custom subclasses or leave the default implementation. See also `get_customattributes`.

**Parameters**

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>key</i>	The attribute key.
<i>value</i>	The attribute value.

**1.34.2.28 set\_extendedattribute()**

```
def shallot.Filesystem.Handler.set_extendedattribute (
    self,
    operation,
    eurl,
    attribname,
    value )
```

Sets the value for a particular extended attribute.

Override this method in custom subclasses or leave the default implementation. Extended attributes are all kinds of properties of a file which aren't handled otherwise. It can contain filesystem's extended attributes, permission information and more.

**Parameters**

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>attribname</i>	The attribute name.
<i>value</i>	The new attribute value (as byte string).

**1.34.2.29 set\_mtime()**

```
def shallot.Filesystem.Handler.set_mtime (
    self,
    operation,
    eurl,
    mtime )
```

Set the mtime of a file.

Override this method in custom subclasses.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> address this call is referred to.
<i>mtime</i>	The <code>datetime.datetime</code> , which specifies the new modification time.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.35 shallot.Operations.HandlerTransfer Class Reference

Used for some additional functionality in some transfer operations in [shallot.Filesystem.Handler](#) instances.

### Public Member Functions

- `def __init__ (self)`  
*Can't be constructed directly.*
- `def increment_transferred_bytes (self, donebytes)`  
*Notify that a certain amount of data (in byte) are transferred.*
- `def respect_cancel (self)`  
*Called from time to time for allowing the user to cancel the transfer.*

### 1.35.1 Detailed Description

Used for some additional functionality in some transfer operations in [shallot.Filesystem.Handler](#) instances.

### 1.35.2 Member Function Documentation

#### 1.35.2.1 increment\_transferred\_bytes()

```
def shallot.Operations.HandlerTransfer.increment_transferred_bytes (
    self,
    donebytes )
```

Notify that a certain amount of data (in byte) are transferred.

This is used for progress monitoring.

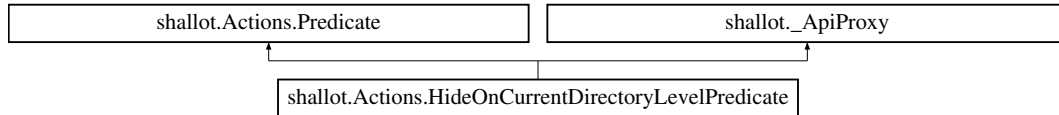
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.36 shallot.Actions.HideOnCurrentDirectoryLevelPredicate Class Reference

Shows actions only when not searching for 'current directory level' actions.

Inheritance diagram for shallot.Actions.HideOnCurrentDirectoryLevelPredicate:



### Public Member Functions

- `def __init__(self)`

#### 1.36.1 Detailed Description

Shows actions only when not searching for 'current directory level' actions.

See base class for more information.

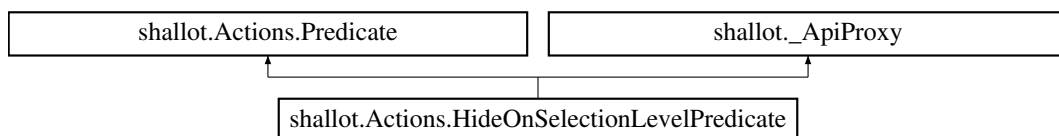
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.37 shallot.Actions.HideOnSelectionLevelPredicate Class Reference

Shows actions only when not searching for 'selection level' actions.

Inheritance diagram for shallot.Actions.HideOnSelectionLevelPredicate:



### Public Member Functions

- `def __init__(self)`

#### 1.37.1 Detailed Description

Shows actions only when not searching for 'selection level' actions.

See base class for more information.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`



## 1.38 shallot.IntlStringMap.IntlString Class Reference

A multi-language string.

### Public Member Functions

- `def \_\_init\_\_ (self, strings)`  
*Creates a multi-language string by a dictionary, considering the keys as language code and the values as the string in that language.*
- `def get (self)`  
*Returns the best variant as plain Python string, depending on the current user interface language.*

### 1.38.1 Detailed Description

A multi-language string.

### 1.38.2 Constructor & Destructor Documentation

#### 1.38.2.1 `__init__()`

```
def shallot.IntlStringMap.IntlString.__init__ (
    self,
    strings )
```

Creates a multi-language string by a dictionary, considering the keys as language code and the values as the string in that language.

Example: `'shallot.IntlStringMap.IntlString({'en':'yes', 'de':'ja'})'`

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.39 shallot.IntlStringMap Class Reference

A multi-language string map used for [localization](#) of plugins.

### Classes

- class [IntlString](#)  
*A multi-language string.*

## Public Member Functions

- `def __init__ (self, stringdicts)`  
*Creates a map of internationalized strings.*

## Public Attributes

- `map`  
*A map of [shallot.IntlStringMap.IntlString](#) instances.*

### 1.39.1 Detailed Description

A multi-language string map used for [localization](#) of plugins.

Internally it uses [shallot.IntlStringMap.IntlString](#) instances, but is more convenient for dealing with a larger amount of strings.

### 1.39.2 Constructor & Destructor Documentation

#### 1.39.2.1 `__init__()`

```
def shallot.IntlStringMap.__init__ (
    self,
    stringdicts )
```

Creates a map of internationalized strings.

For each keyword-argument, this map will get a member with the key name as member name. The values are dictionaries in the same form as for [shallot.IntlStringMap.IntlString](#). The members will be the best available language variants as plain Python strings.

Example: `Strings = shallot.IntlStringMap(Foo = {"en":"foo", "it":"Fuh"}, Bar = {"en":"bar", "it":"Barra"})`

Then `Strings.Foo` might contain the Python string "Fuh".

### 1.39.3 Member Data Documentation

#### 1.39.3.1 `map`

```
shallot.IntlStringMap.map
```

A map of [shallot.IntlStringMap.IntlString](#) instances.

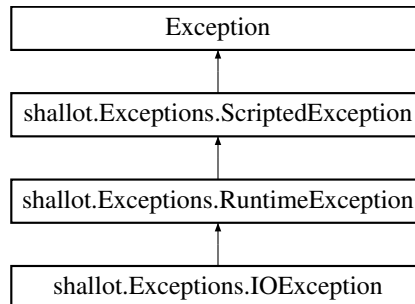
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.40 shallot.Exceptions.IOException Class Reference

Shallot exception in IO.

Inheritance diagram for shallot.Exceptions.IOException:



### Public Member Functions

- `def __init__ (self, details=None, message=None, isRetryable=None, autoRetryRecommendedIn=-1, detailsAreInteresting=None, _class="")`  
*See [shallot.Exceptions.ScriptedException.\\_\\_init\\_\\_](#) for details.*
- `def isExceptionClass (self, classname)`  
*Checks if this exception is instance of a given exception class.*

### 1.40.1 Detailed Description

Shallot exception in IO.

It allows resume and typically allows retry (special cases may override each of them). Read [more about Shallot Exceptions](#).

### 1.40.2 Member Function Documentation

#### 1.40.2.1 isExceptionClass()

```
def shallot.Exceptions.ScriptedException.isExceptionClass (
    self,
    classname ) [inherited]
```

Checks if this exception is instance of a given exception class.

#### Parameters

<i>classname</i>	A exception class name (as string).
------------------	-------------------------------------

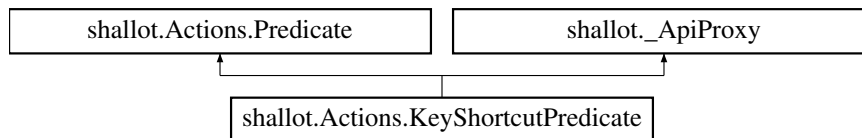
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.41 shallot.Actions.KeyShortcutPredicate Class Reference

Sets a keyboard shortcut.

Inheritance diagram for shallot.Actions.KeyShortcutPredicate:



### Public Member Functions

- def `__init__` (self, shortcut, triggers\_on\_currentdirectory\_level)

### 1.41.1 Detailed Description

Sets a keyboard shortcut.

See base class for more information.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.42 shallot.Logging Class Reference

Methods for [logging](#).

### Static Public Member Functions

- def `log_debug` (s)  
*Logs a message with debug severity.*
- def `log_info` (s)  
*Logs a message with info severity.*
- def `log_warning` (s)  
*Logs a message with warning severity.*
- def `log_error` (s)  
*Logs a message with message severity.*

### 1.42.1 Detailed Description

Methods for [logging](#).

### 1.42.2 Member Function Documentation

#### 1.42.2.1 log\_debug()

```
def shallot.Logging.log_debug (  
    s ) [static]
```

Logs a message with debug severity.

##### Parameters

s	The message.
---	--------------

#### 1.42.2.2 log\_error()

```
def shallot.Logging.log_error (  
    s ) [static]
```

Logs a message with message severity.

##### Parameters

s	The message.
---	--------------

#### 1.42.2.3 log\_info()

```
def shallot.Logging.log_info (  
    s ) [static]
```

Logs a message with info severity.

##### Parameters

s	The message.
---	--------------

#### 1.42.2.4 log\_warning()

```
def shallot.Logging.log_warning (
    s ) [static]
```

Logs a message with warning severity.

##### Parameters

s	The message.
---	--------------

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.43 shallot.MainWindow Class Reference

A Shallot [main window](#).

### Public Member Functions

- def [jump\\_to\\_eurl](#) (self, eurl)  
*Let the current view jump to another location.*
- def [get\\_current\\_directory\\_node](#) (self)  
*Gets the [shallot.Filesystem.Node](#) selected in the current view.*

### Static Public Member Functions

- def [open\\_items](#) (eurls)  
*Opens a list of files, largely as if the user had doubleclicked on them.*
- def [current](#) ()  
*Returns the current [shallot.MainWindow](#).*

#### 1.43.1 Detailed Description

A Shallot [main window](#).

#### 1.43.2 Member Function Documentation

##### 1.43.2.1 jump\_to\_eurl()

```
def shallot.MainWindow.jump_to_eurl (
    self,
    eurl )
```

Let the current view jump to another location.

## Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> address to jump to.
------------	--

## 1.43.2.2 open\_items()

```
def shallot.MainWindow.open_items (
    eurls ) [static]
```

Opens a list of files, largely as if the user had doubleclicked on them.

## Parameters

<i>eurls</i>	a list of <a href="#">shallot.Eurl</a> .
--------------	--

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.44 shallot.Actions.ExecutionUserFeedback.MessageBoxButton Class Reference

Buttons in a message box from [shallot.Actions.ExecutionUserFeedback](#).

## Static Public Attributes

- [OK](#) = internalstuff.messageboxbutton\_ok()  
*OK button.*
- [Continue](#) = internalstuff.messageboxbutton\_continue()  
*Continue button.*
- [Cancel](#) = internalstuff.messageboxbutton\_cancel()  
*Cancel button.*
- [Retry](#) = internalstuff.messageboxbutton\_retry()  
*Retry button.*
- [Yes](#) = internalstuff.messageboxbutton\_yes()  
*Yes button.*
- [No](#) = internalstuff.messageboxbutton\_no()  
*No button.*

## 1.44.1 Detailed Description

Buttons in a message box from [shallot.Actions.ExecutionUserFeedback](#).

## 1.44.2 Member Data Documentation

### 1.44.2.1 Cancel

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.Cancel = internalstuff.messageboxbutton↵  
_cancel() [static]
```

Cancel button.

### 1.44.2.2 Continue

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.Continue = internalstuff.messageboxbutton↵  
_continue() [static]
```

Continue button.

### 1.44.2.3 No

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.No = internalstuff.messageboxbutton↵  
no() [static]
```

No button.

### 1.44.2.4 OK

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.OK = internalstuff.messageboxbutton↵  
ok() [static]
```

OK button.

### 1.44.2.5 Retry

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.Retry = internalstuff.messageboxbutton↵  
_retry() [static]
```

Retry button.



## 1.44.2.6 Yes

```
shallot.Actions.ExecutionUserFeedback.MessageBoxButton.Yes = internalstuff.messageboxbutton_↵
yes() [static]
```

Yes button.

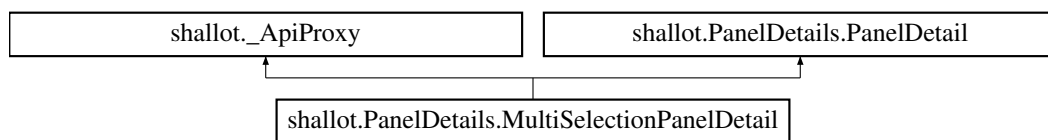
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.45 shallot.PanelDetails.MultiSelectionPanelDetail Class Reference

Abstract base class for a detail panel entry, which occurs when multiple items are selected.

Inheritance diagram for shallot.PanelDetails.MultiSelectionPanelDetail:



## Public Member Functions

- def `__init__` (self, positionGroup=None, positionIndex=None, valueWidthHint=4)
- def `set_value` (self, detail, nodes, operation)
 

*This method is called whenever an output must be determined for certain nodes.*
- def `link_triggered` (self, nodes, linktarget)
 

*This method is called whenever the user triggers a link.*

## Static Public Member Functions

- def `register` (detail)
 

*Registers a `shallot.PanelDetails.PanelDetail` in the Shallot core.*

## Static Public Attributes

- `INDEX_VERYINTERESTING` = `internalstuff.paneldetail_positionindex_veryinteresting()`

*A integer value to be used as base display index for very interesting details.*
- `INDEX_INTERESTING` = `internalstuff.paneldetail_positionindex_interesting()`

*A integer value to be used as base display index for interesting details.*
- `INDEX_EXOTIC` = `internalstuff.paneldetail_positionindex_exotic()`

*A integer value to be used as base display index for exotic details.*

### 1.45.1 Detailed Description

Abstract base class for a detail panel entry, which occurs when multiple items are selected.

See [shallot.PanelDetails.PanelDetail.register](#) for registering custom implementations to shallot.

### 1.45.2 Constructor & Destructor Documentation

#### 1.45.2.1 `__init__()`

```
def shallot.PanelDetails.MultiSelectionPanelDetail.__init__ (
    self,
    positionGroup = None,
    positionIndex = None,
    valueWidthHint = 4 )
```

##### Parameters

<i>positionGroup</i>	Controls display order. See <a href="#">Position Indexes</a> for details. Use one of the <code>INDEX_*</code> values from <a href="#">shallot.PanelDetails.PanelDetail</a> .
<i>positionIndex</i>	Controls display order. See <a href="#">Position Indexes</a> for details.
<i>valueWidthHint</i>	A width in centimeters to reserve for printing the values.

### 1.45.3 Member Function Documentation

#### 1.45.3.1 `link_triggered()`

```
def shallot.PanelDetails.MultiSelectionPanelDetail.link_triggered (
    self,
    nodes,
    linktarget )
```

This method is called whenever the user triggers a link.

Override in custom implementations if links are used.

##### Parameters

<i>nodes</i>	The selected list of <a href="#">shallot.Filesystem.Node</a> object.
<i>linktarget</i>	The link target string, as specified in <a href="#">shallot.PanelDetails.PanelDetailValueElementButton.__init__</a> .

## 1.45.3.2 register()

```
def shallot.PanelDetails.PanelDetail.register (
    detail ) [static], [inherited]
```

Registers a [shallot.PanelDetails.PanelDetail](#) in the Shallot core.

## Parameters

<i>detail</i>	The <a href="#">shallot.PanelDetails.PanelDetail</a> instance.
---------------	--

## 1.45.3.3 set\_value()

```
def shallot.PanelDetails.MultiSelectionPanelDetail.set_value (
    self,
    detail,
    nodes,
    operation )
```

This method is called whenever an output must be determined for certain nodes.

It must return a list of 2-tuples (one for each row). Those are each name/value pairs with the label of that row and the value (which is a list of [shallot.PanelDetails.AbstractPanelDetailValueElement](#)). For large waiting times, you should just return an empty value list for your rows, start an asynchronous execution and use [shallot.PanelDetailInst.set\\_row](#) in the end.

Override this method in custom subclasses.

## Parameters

<i>detail</i>	A <a href="#">shallot.PanelDetailInst</a> object for printing the details. Only used in advanced cases.
<i>nodes</i>	The selected list of <a href="#">shallot.Filesystem.Node</a> object.
<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.

## 1.45.4 Member Data Documentation

## 1.45.4.1 INDEX\_EXOTIC

```
shallot.PanelDetails.PanelDetail.INDEX_EXOTIC = internalstuff.paneldetail_positionindex_↵
exotic() [static], [inherited]
```

A integer value to be used as base display index for exotic details.

#### 1.45.4.2 INDEX\_INTERESTING

```
shallot.PanelDetails.PanelDetail.INDEX_INTERESTING = internalstuff.paneldetail_positionindex↔
_indexing() [static], [inherited]
```

A integer value to be used as base display index for interesting details.

#### 1.45.4.3 INDEX\_VERYINTERESTING

```
shallot.PanelDetails.PanelDetail.INDEX_VERYINTERESTING = internalstuff.paneldetail_positionindex↔
_veryinteresting() [static], [inherited]
```

A integer value to be used as base display index for very interesting details.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.46 shallot.Filesystem.Node Class Reference

A model representation of a location in the filesystem tree.

### Public Member Functions

- def `__init__` (self)  
*Can't be constructed directly.*
- def `eurl` (self)  
*Return the [shallot.Eurl](#) entry address.*
- def `nodetype` (self)  
*Returns the [shallot.Filesystem.NodeType](#) node type.*
- def `add_detail` (self, column)  
*Adds a detail to this node.*
- def `set_icon` (self, icon)  
*Sets the node icon.*
- def `set_displayname` (self, displayname)  
*Sets the displayed name of the node.*

#### 1.46.1 Detailed Description

A model representation of a location in the filesystem tree.

It represents stuff like a file or a directory. Those instances are used at many places for all kinds of operations.

Please note: Instances can be associated with any kind of elements in the filesystem (files, directories, links, ...). It might also point to something which does not exist at all (see `parentnode`). The documentation sometimes explicitly makes a difference between those kinds (files, directories, links, ...; often called 'node type'). But it often uses the term 'file' implicitly while meaning all kinds of elements; assuming that e.g. a directory is just a special kind of a file. It should be clear from the particular context which meaning applies.

## 1.46.2 Member Function Documentation

### 1.46.2.1 add\_detail()

```
def shallot.Filesystem.Node.add_detail (
    self,
    column )
```

Adds a detail to this node.

#### Parameters

<i>column</i>	The <a href="#">shallot.DetailColumn</a> instance to add.
---------------	---

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.47 shallot.Filesystem.NodeList Class Reference

A list editor for a filesystem node list.

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)  
*Can't be constructed directly.*
- def [set\\_items](#) (self, items)  
*Sets a new list content.*
- def [add\\_item](#) (self, item)  
*Adds a new children to the list, directly showing that in the user interface, while you can proceed filling the list.*
- def [begin\\_iterative\\_adding](#) (self)  
*Must be called before you begin to iteratively fill the children list (e.g.*
- def [end\\_iterative\\_adding](#) (self)  
*Must be called after you iteratively filled the children list with [shallot.Filesystem.NodeList.add\\_item](#).*

### 1.47.1 Detailed Description

A list editor for a filesystem node list.

They are mainly used for specifying child nodes in [shallot.Filesystem.Handler.itemlist](#) and provide some help there. In most cases, you should just use [shallot.Filesystem.NodeList.set\\_items](#).

## 1.47.2 Member Function Documentation

### 1.47.2.1 `add_item()`

```
def shallot.Filesystem.NodeList.add_item (
    self,
    item )
```

Adds a new children to the list, directly showing that in the user interface, while you can proceed filling the list.

Please read [shallot.Filesystem.NodeList.begin\\_iterative\\_adding](#) as well! In most cases, you don't need this function.

#### Parameters

<i>item</i>	A the new children node. This must be the basename.
-------------	---

### 1.47.2.2 `begin_iterative_adding()`

```
def shallot.Filesystem.NodeList.begin_iterative_adding (
    self )
```

Must be called before you begin to iteratively fill the children list (e.g.

with [shallot.Filesystem.NodeList.add\\_item](#)). You must also call [shallot.Filesystem.NodeList.end\\_iterative\\_adding](#) afterwards.

### 1.47.2.3 `end_iterative_adding()`

```
def shallot.Filesystem.NodeList.end_iterative_adding (
    self )
```

Must be called after you iteratively filled the children list with [shallot.Filesystem.NodeList.add\\_item](#).

This automatically removes all the old nodes, which you haven't added in this session.

### 1.47.2.4 `set_items()`

```
def shallot.Filesystem.NodeList.set_items (
    self,
    items )
```

Sets a new list content.

This function is all you need in most situations. For iteratively adding nodes, which makes the intermediate results visible in the user interface, see [shallot.Filesystem.NodeList.add\\_item](#).

## Parameters

<i>items</i>	A list of strings providing the new children nodes. The list must contain the basenames of all existing children (with the node type you got as argument).
--------------	--

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.48 shallot.Filesystem.NodeType Class Reference

Enumeration of types a [shallot.Filesystem.Node](#) can have.

### Static Public Attributes

- [File](#) = internalstuff.filesystemnodetype\_file()  
*Usual file.*
- [Directory](#) = internalstuff.filesystemnodetype\_directory()  
*Directory.*
- [Link](#) = internalstuff.filesystemnodetype\_link()  
*Link.*
- [Unknown](#) = internalstuff.filesystemnodetype\_unknown()  
*Unknown.*
- [FIRSTTYPE](#) = internalstuff.filesystemnodetype\_firsttype()  
*Points to the first type (used for generating int ranges).*
- [LASTPHYSICALTYPE](#) = internalstuff.filesystemnodetype\_lastphysicaltype()  
*Points to the last physical type, excluding internal magic types (used for generating int ranges).*
- [NONE](#) = internalstuff.filesystemnodetype\_none()  
*Does not exist.*

### 1.48.1 Detailed Description

Enumeration of types a [shallot.Filesystem.Node](#) can have.

### 1.48.2 Member Data Documentation

#### 1.48.2.1 Directory

```
shallot.Filesystem.NodeType.Directory = internalstuff.filesystemnodetype_directory() [static]
```

Directory.

#### 1.48.2.2 File

```
shallot.Filesystem.NodeType.File = internalstuff.filesystemnodetype_file() [static]
```

Usual file.

#### 1.48.2.3 FIRSTTYPE

```
shallot.Filesystem.NodeType.FIRSTTYPE = internalstuff.filesystemnodetype_firsttype() [static]
```

Points to the first type (used for generating int ranges).

#### 1.48.2.4 LASTPHYSICALTYPE

```
shallot.Filesystem.NodeType.LASTPHYSICALTYPE = internalstuff.filesystemnodetype_lastphysicaltype()  
[static]
```

Points to the last physical type, excluding internal magic types (used for generating int ranges).

#### 1.48.2.5 Link

```
shallot.Filesystem.NodeType.Link = internalstuff.filesystemnodetype_link() [static]
```

Link.

#### 1.48.2.6 NONE

```
shallot.Filesystem.NodeType.NONE = internalstuff.filesystemnodetype_none() [static]
```

Does not exist.

#### 1.48.2.7 Unknown

```
shallot.Filesystem.NodeType.Unknown = internalstuff.filesystemnodetype_unknown() [static]
```

Unknown.

The documentation for this class was generated from the following file:

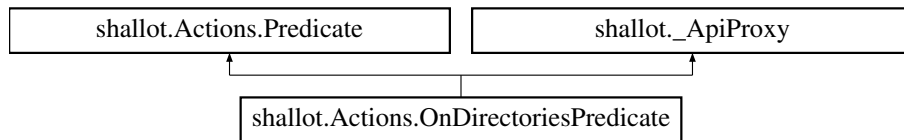
- /home/pino/projects/shallot/shallot.py



## 1.49 shallot.Actions.OnDirectoriesPredicate Class Reference

Shows actions only on directories.

Inheritance diagram for shallot.Actions.OnDirectoriesPredicate:



### Public Member Functions

- `def __init__(self)`

#### 1.49.1 Detailed Description

Shows actions only on directories.

See base class for more information.

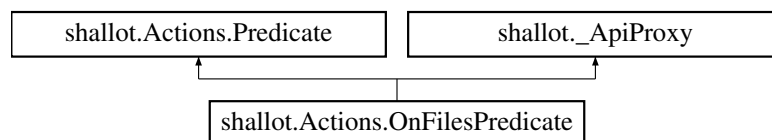
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.50 shallot.Actions.OnFilesPredicate Class Reference

Shows actions only on files.

Inheritance diagram for shallot.Actions.OnFilesPredicate:



### Public Member Functions

- `def __init__(self)`

### 1.50.1 Detailed Description

Shows actions only on files.

See base class for more information.

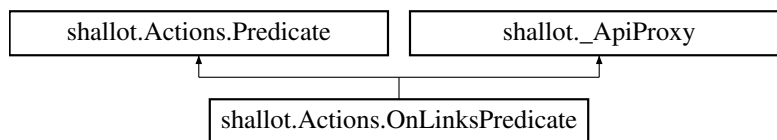
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.51 shallot.Actions.OnLinksPredicate Class Reference

Shows actions only on links.

Inheritance diagram for shallot.Actions.OnLinksPredicate:



### Public Member Functions

- def `__init__`(self)

### 1.51.1 Detailed Description

Shows actions only on links.

See base class for more information.

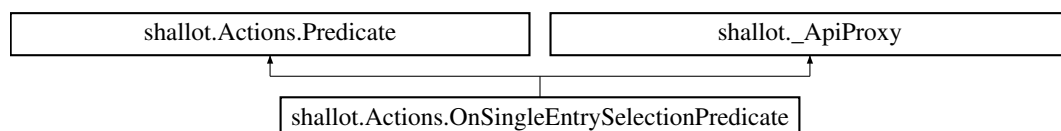
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.52 shallot.Actions.OnSingleEntrySelectionPredicate Class Reference

Shows actions only on single-entry selections.

Inheritance diagram for shallot.Actions.OnSingleEntrySelectionPredicate:



## Public Member Functions

- `def __init__ (self)`

### 1.52.1 Detailed Description

Shows actions only on single-entry selections.

See base class for more information.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.53 shallot.Operations.Operation Class Reference

Transactional read and write filesystem accesses.

## Public Member Functions

- `def fetch\_container\_file (self, eurl, namehint)`  
*Fetches the container file for a eurl locally and returns the local path.*
- `def fetch\_file (self, eurl, namehint)`  
*Fetches a file locally and returns the local path.*
- `def abort (self)`  
*Aborts transaction dropping all pending changes.*
- `def commit (self)`  
*Commits transaction applying all pending changes.*
- `def enable\_detailscache (self)`  
*Enable details caching.*
- `def is\_detailscache\_enabled (self)`  
*Is details caching enabled?*
- `def store\_detail\_in\_cache (self, eurl, column, value)`  
*Stores a column detail in cache.*
- `def get\_detail\_from\_cache (self, eurl, column)`  
*Gets a column value from cache.*
- `def set\_custom\_data (self, eurl, key, value)`  
*Stores custom data.*
- `def get\_custom\_data (self, eurl, key)`  
*Gets stored custom data.*
- `def filesystem (self)`  
*Gets the [shallot.Operations.FilesystemOperation](#) filesystem operation object.*

## Static Public Member Functions

- `def create ()`  
*Creates a new [shallot.Operations.Operation](#).*

### 1.53.1 Detailed Description

Transactional read and write filesystem accesses.

Read [more](#).

### 1.53.2 Member Function Documentation

#### 1.53.2.1 `fetch_container_file()`

```
def shallot.Operations.Operation.fetch_container_file (
    self,
    eurl,
    namehint )
```

Fetches the container file for a eurl locally and returns the local path.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> location to fetch the container for.
<i>namehint</i>	A small file name prefix like "temp".

#### 1.53.2.2 `fetch_file()`

```
def shallot.Operations.Operation.fetch_file (
    self,
    eurl,
    namehint )
```

Fetches a file locally and returns the local path.

##### Parameters

<i>eurl</i>	The <a href="#">shallot.Eurl</a> location to fetch the container for.
<i>namehint</i>	A small file name prefix like "temp".

#### 1.53.2.3 `get_custom_data()`

```
def shallot.Operations.Operation.get_custom_data (
    self,
```

```
    url,  
    key )
```

Gets stored custom data.

#### Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> instance you want to get data about (or None).
<i>key</i>	Custom data key name.

#### 1.53.2.4 get\_detail\_from\_cache()

```
def shallot.Operations.Operation.get_detail_from_cache (  
    self,  
    url,  
    column )
```

Gets a column value from cache.

#### Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> to get a detail value for.
<i>column</i>	The <a href="#">shallot.DetailColumn</a> to get a detail value for.

#### 1.53.2.5 set\_custom\_data()

```
def shallot.Operations.Operation.set_custom_data (  
    self,  
    url,  
    key,  
    value )
```

Stores custom data.

#### Parameters

<i>url</i>	The <a href="#">shallot.Eurl</a> instance you want to store data about (or None).
<i>key</i>	Custom data key name.
<i>value</i>	Custom data value.

#### 1.53.2.6 store\_detail\_in\_cache()

```
def shallot.Operations.Operation.store_detail_in_cache (  
    self,
```

```

    url,
    column,
    value )

```

Stores a column detail in cache.

#### Parameters

<i>url</i>	The <a href="#">shallot.Url</a> to store a detail value for.
<i>column</i>	The <a href="#">shallot.DetailColumn</a> to store a detail value for.
<i>value</i>	The column value for this entry.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.54 shallot.Operations Class Reference

Everything about [operations on the filesystem](#).

### Classes

- class [FilesystemOperation](#)  
*A high-level provider of filesystem operations.*
- class [FilesystemOperationProgressMonitor](#)  
*Abstract base class for progress monitors, used for monitoring progress of some operations in [shallot.Operations](#).↔ [FilesystemOperation](#) and for conflict resolution.*
- class [FilesystemOperationStep](#)  
*One step in a larger filesystem transfer running inside some operations in [shallot.Operations.FilesystemOperation](#).*
- class [HandlerTransfer](#)  
*Used for some additional functionality in some transfer operations in [shallot.Filesystem.Handler](#) instances.*
- class [Operation](#)  
*Transactional read and write filesystem accesses.*

### 1.54.1 Detailed Description

Everything about [operations on the filesystem](#).

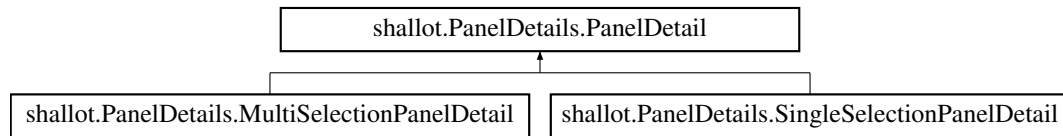
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.55 shallot.PanelDetails.PanelDetail Class Reference

Abstract base class for a detail panel entry.

Inheritance diagram for shallot.PanelDetails.PanelDetail:



### Public Member Functions

- `def __init__(self)`

### Static Public Member Functions

- `def register(detail)`  
Registers a [shallot.PanelDetails.PanelDetail](#) in the Shallot core.

### Static Public Attributes

- `INDEX_VERYINTERESTING = internalstuff.paneldetail_positionindex_veryinteresting()`  
A integer value to be used as base display index for very interesting details.
- `INDEX_INTERESTING = internalstuff.paneldetail_positionindex_interesting()`  
A integer value to be used as base display index for interesting details.
- `INDEX_EXOTIC = internalstuff.paneldetail_positionindex_exotic()`  
A integer value to be used as base display index for exotic details.

#### 1.55.1 Detailed Description

Abstract base class for a detail panel entry.

See the subclasses.

#### 1.55.2 Member Function Documentation

##### 1.55.2.1 register()

```
def shallot.PanelDetails.PanelDetail.register (
    detail ) [static]
```

Registers a [shallot.PanelDetails.PanelDetail](#) in the Shallot core.

**Parameters**

<i>detail</i>	The <a href="#">shallot.PanelDetails.PanelDetail</a> instance.
---------------	--

**1.55.3 Member Data Documentation****1.55.3.1 INDEX\_EXOTIC**

```
shallot.PanelDetails.PanelDetail.INDEX_EXOTIC = internalstuff.paneldetail_positionindex↔
exotic() [static]
```

A integer value to be used as base display index for exotic details.

**1.55.3.2 INDEX\_INTERESTING**

```
shallot.PanelDetails.PanelDetail.INDEX_INTERESTING = internalstuff.paneldetail_positionindex↔
_interesting() [static]
```

A integer value to be used as base display index for interesting details.

**1.55.3.3 INDEX\_VERYINTERESTING**

```
shallot.PanelDetails.PanelDetail.INDEX_VERYINTERESTING = internalstuff.paneldetail_positionindex↔
_veryinteresting() [static]
```

A integer value to be used as base display index for very interesting details.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

**1.56 shallot.PanelDetails.PanelDetailInst Class Reference**

The storage for detail value rows, which are the actual content of a [shallot.PanelDetails.PanelDetail](#) for an actual file selection.



## Public Member Functions

- def `__init__` (self, \_detail)  
*Can't be constructed directly.*
- def `set_row` (self, row, val)  
*Sets the value for one row.*

### 1.56.1 Detailed Description

The storage for detail value rows, which are the actual content of a [shallot.PanelDetails.PanelDetail](#) for an actual file selection.

### 1.56.2 Member Function Documentation

#### 1.56.2.1 `set_row()`

```
def shallot.PanelDetails.PanelDetailInst.set_row (
    self,
    row,
    val )
```

Sets the value for one row.

#### Parameters

<i>row</i>	The label of that row.
<i>val</i>	The value (which is a list of <a href="#">shallot.PanelDetails.AbstractPanelDetailValueElement</a> ).

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.57 shallot.PanelDetails Class Reference

Everything about [panel details](#).

## Classes

- class [AbstractPanelDetailValueElement](#)  
*Abstract base class for an element of a panel detail row's value.*
- class [MultiSelectionPanelDetail](#)  
*Abstract base class for a detail panel entry, which occurs when multiple items are selected.*
- class [PanelDetail](#)

*Abstract base class for a detail panel entry.*

- class [PanelDetailInst](#)

*The storage for detail value rows, which are the actual content of a [shallot.PanelDetails.PanelDetail](#) for an actual file selection.*

- class [PanelDetailValueElementButton](#)

*A link button (for executing some code on user behalf) as element of a panel detail row's value.*

- class [PanelDetailValueElementIcon](#)

*An icon as element of a panel detail row's value.*

- class [PanelDetailValueElementString](#)

*A piece of text as element of a panel detail row's value.*

- class [PanelDetailValueElementWaiting](#)

*A 'waiting' placeholder as element of a panel detail row's value.*

- class [SingleSelectionPanelDetail](#)

*Abstract base class for a detail panel entry, which occurs when one item is selected.*

### 1.57.1 Detailed Description

Everything about [panel details](#).

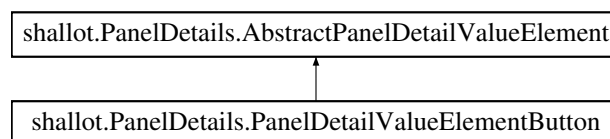
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.58 shallot.PanelDetails.PanelDetailValueElementButton Class Reference

A link button (for executing some code on user behalf) as element of a panel detail row's value.

Inheritance diagram for `shallot.PanelDetails.PanelDetailValueElementButton`:



### Public Member Functions

- `def \_\_init\_\_(self, text, target, autohide=True)`

### Public Attributes

- `type`
- `value`

### 1.58.1 Detailed Description

A link button (for executing some code on user behalf) as element of a panel detail row's value.

Use [shallot.PanelDetails.SingleSelectionPanelDetail.link\\_triggered](#) or [shallot.PanelDetails.MultiSelectionPanelDetail.link\\_triggered](#) for handling user actions.

### 1.58.2 Constructor & Destructor Documentation

#### 1.58.2.1 \_\_init\_\_()

```
def shallot.PanelDetails.PanelDetailValueElementButton.__init__ (
    self,
    text,
    target,
    autohide = True )
```

#### Parameters

<i>text</i>	The text.
<i>target</i>	The link target name (will be passed to the handler function).
<i>autohide</i>	If the button shall only be visible when the mouse cursor is in the details panel.

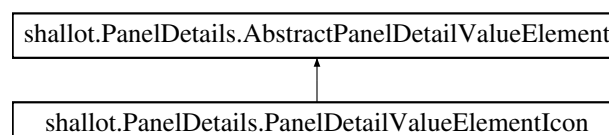
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.59 shallot.PanelDetails.PanelDetailValueElementIcon Class Reference

An icon as element of a panel detail row's value.

Inheritance diagram for shallot.PanelDetails.PanelDetailValueElementIcon:



### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, icon)

## Public Attributes

- **type**
- **value**

### 1.59.1 Detailed Description

An icon as element of a panel detail row's value.

### 1.59.2 Constructor & Destructor Documentation

#### 1.59.2.1 `__init__()`

```
def shallot.PanelDetails.PanelDetailValueElementIcon.__init__(  
    self,  
    icon )
```

#### Parameters

<i>icon</i>	The icon name.
-------------	----------------

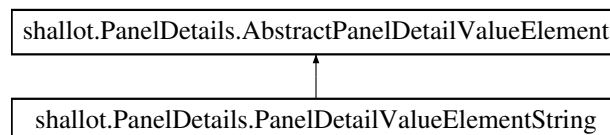
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.60 shallot.PanelDetails.PanelDetailValueElementString Class Reference

A piece of text as element of a panel detail row's value.

Inheritance diagram for shallot.PanelDetails.PanelDetailValueElementString:



## Public Member Functions

- `def \_\_init\_\_(self, text)`

## Public Attributes

- **type**
- **value**

### 1.60.1 Detailed Description

A piece of text as element of a panel detail row's value.

### 1.60.2 Constructor & Destructor Documentation

#### 1.60.2.1 \_\_init\_\_()

```
def shallot.PanelDetails.PanelDetailValueElementString.__init__ (
    self,
    text )
```

#### Parameters

<i>text</i>	The text.
-------------	-----------

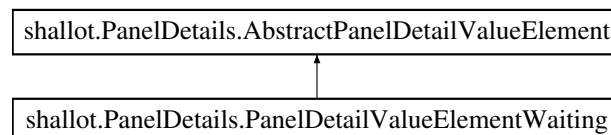
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.61 shallot.PanelDetails.PanelDetailValueElementWaiting Class Reference

A 'waiting' placeholder as element of a panel detail row's value.

Inheritance diagram for shallot.PanelDetails.PanelDetailValueElementWaiting:



## Public Member Functions

- **def \_\_init\_\_ (self)**

## Public Attributes

- **type**
- **value**

### 1.61.1 Detailed Description

A 'waiting' placeholder as element of a panel detail row's value.

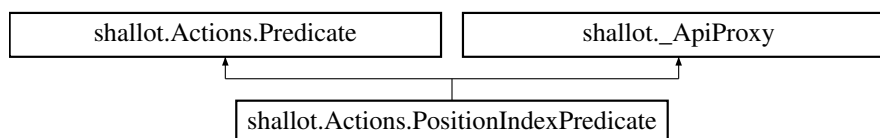
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.62 shallot.Actions.PositionIndexPredicate Class Reference

Sets a positioning information index.

Inheritance diagram for shallot.Actions.PositionIndexPredicate:



## Public Member Functions

- **def \_\_init\_\_**(self, index)

### 1.62.1 Detailed Description

Sets a positioning information index.

See base class for more information.

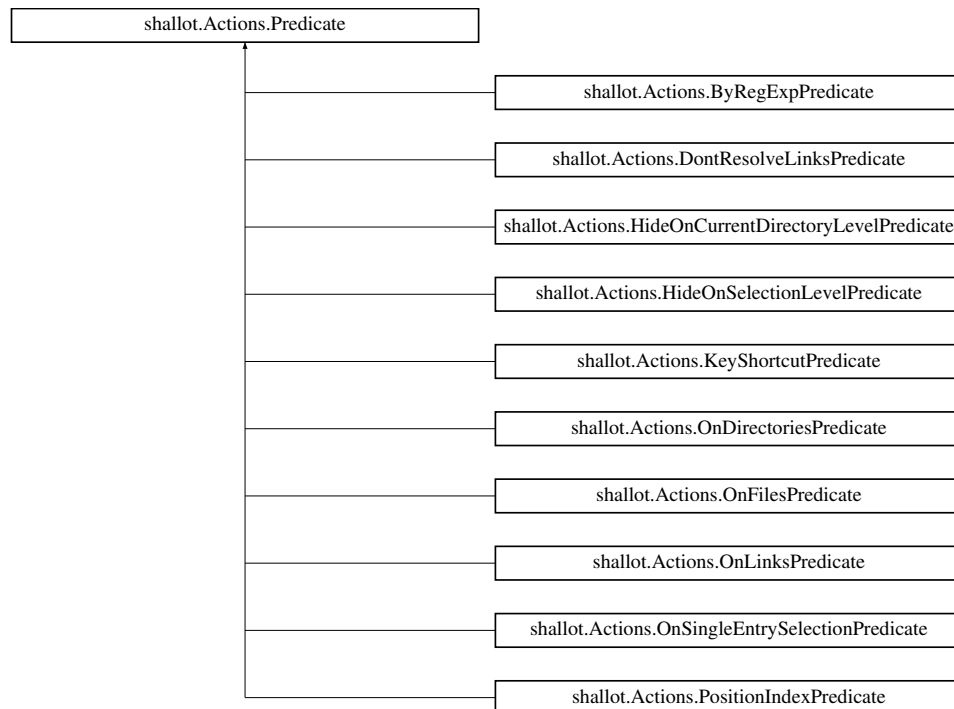
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.63 shallot.Actions.Predicate Class Reference

Controls when and how an action is created.

Inheritance diagram for shallot.Actions.Predicate:



### 1.63.1 Detailed Description

Controls when and how an action is created.

Used in [shallot.Actions.register](#). See base class for more information.

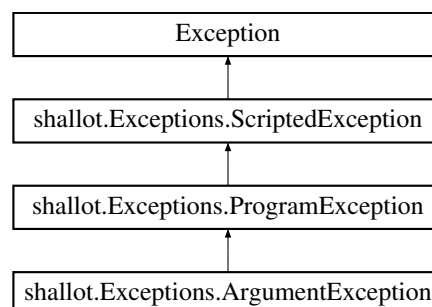
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.64 shallot.Exceptions.ProgramException Class Reference

Shallot exception for program errors (typically logical stuff) in the program or plugin.

Inheritance diagram for shallot.Exceptions.ProgramException:



## Public Member Functions

- `def __init__ (self, details=None, message=None, isResumeable=True, detailsAreInteresting=None, _↔  
class="")`  
See [shallot.Exceptions.ScriptedException.\\_\\_init\\_\\_](#) for details.
- `def isExceptionClass (self, classname)`  
Checks if this exception is instance of a given exception class.

### 1.64.1 Detailed Description

Shallot exception for program errors (typically logical stuff) in the program or plugin.

It typically allows resume but no retry (special cases may override each of them). Read [more about Shallot Exceptions](#).

### 1.64.2 Member Function Documentation

#### 1.64.2.1 isExceptionClass()

```
def shallot.Exceptions.ScriptedException.isExceptionClass (
    self,
    classname ) [inherited]
```

Checks if this exception is instance of a given exception class.

#### Parameters

<i>classname</i>	A exception class name (as string).
------------------	-------------------------------------

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.65 shallot.FilePropertyDialog.Tab.PropertyButtonConfig Class Reference

Definitions for buttons in a [shallot.FilePropertyDialog.Tab.PropertyConfig](#).

## Public Member Functions

- `def __init__ (self, label, fct)`



## Public Attributes

- **label**
- **fct**
- **id**

### 1.65.1 Detailed Description

Definitions for buttons in a [shallot.FilePropertyDialog.Tab.PropertyConfig](#).

### 1.65.2 Constructor & Destructor Documentation

#### 1.65.2.1 `__init__()`

```
def shallot.FilePropertyDialog.Tab.PropertyButtonConfig.__init__ (
    self,
    label,
    fct )
```

#### Parameters

<i>label</i>	The button text.
<i>fct</i>	The function to execute when the user clicks on the button.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.66 shallot.FilePropertyDialog.Tab.PropertyConfig Class Reference

Definitions for properties, which present a piece of information in a [shallot.FilePropertyDialog.Tab](#).

## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, title, propertytype, buttons=[ ])

## Public Attributes

- **title**
- **propertytype**
- **buttons**

### 1.66.1 Detailed Description

Definitions for properties, which present a piece of information in a [shallot.FilePropertyDialog.Tab](#).

### 1.66.2 Constructor & Destructor Documentation

#### 1.66.2.1 `__init__()`

```
def shallot.FilePropertyDialog.Tab.PropertyConfig.__init__ (
    self,
    title,
    propertytype,
    buttons = [] )
```

##### Parameters

<i>title</i>	The property title/name.
<i>propertytype</i>	The value type. This controls what you have to return in <a href="#">shallot.FilePropertyDialog.Tab.update_widget</a> and how it is displayed. See the <code>PROPERTYTYPE_*</code> constants.
<i>buttons</i>	A list of <a href="#">shallot.FilePropertyDialog.Tab.PropertyButtonConfig</a> specifying which buttons to show for offering user interaction.

The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.67 shallot.FilePropertyDialog.Tab.PropertyType Class Reference

A type of a single property.

### Static Public Attributes

- [String](#) = `internalstuff.filepropertydialogtab_propertytype_string()`  
*A single string.*
- [StringMap](#) = `internalstuff.filepropertydialogtab_propertytype_stringmap()`  
*A table of key/value string pairs.*
- [IconTextBanner](#) = `internalstuff.filepropertydialogtab_propertytype_icontextbanner()`  
*A mixed banner with texts and icons in a horizontal alignment.*

#### 1.67.1 Detailed Description

A type of a single property.

Used in [shallot.FilePropertyDialog.Tab.PropertyConfig](#).

## 1.67.2 Member Data Documentation

### 1.67.2.1 IconTextBanner

```
shallot.FilePropertyDialog.Tab.PropertyType.IconTextBanner = internalstuff.filepropertydialogtab↔
_propertytype_iconcontextbanner() [static]
```

A mixed banner with texts and icons in a horizontal alignment.

If chosen, return a [shallot.FilePropertyDialog.TabPropertyIconTextBanner](#) in `update_widget`.

### 1.67.2.2 String

```
shallot.FilePropertyDialog.Tab.PropertyType.String = internalstuff.filepropertydialogtab↔
propertytype_string() [static]
```

A single string.

If chosen, return a string in `update_widget`.

### 1.67.2.3 StringMap

```
shallot.FilePropertyDialog.Tab.PropertyType.StringMap = internalstuff.filepropertydialogtab↔
propertytype_stringmap() [static]
```

A table of key/value string pairs.

If chosen, return a list of (string,string) tuples in `update_widget`.

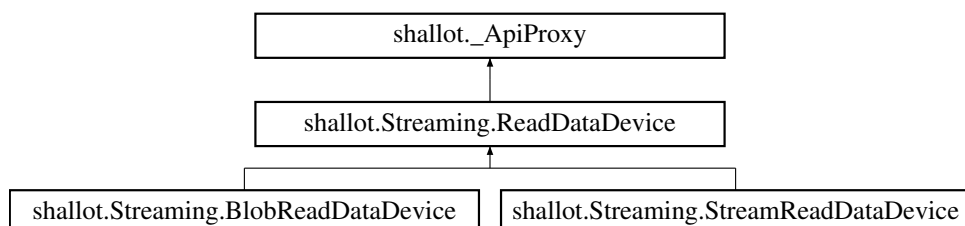
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.68 shallot.Streaming.ReadDataDevice Class Reference

A scripted source of binary content.

Inheritance diagram for `shallot.Streaming.ReadDataDevice`:



## Public Member Functions

- `def __init__(self)`
- `def getdata(self)`  
*Returns a Python byte array containing the content.*
- `def read(self)`  
*Reads a chunk of data as Python byte array.*
- `def readall(self)`  
*Reads the complete data as Python byte array.*

### 1.68.1 Detailed Description

A scripted source of binary content.

Override this class for providing own binary content. You might often find it more convenient to use one of the subclasses.

### 1.68.2 Member Function Documentation

#### 1.68.2.1 `getdata()`

```
def shallot.Streaming.ReadDataDevice.getdata (
    self )
```

Returns a Python byte array containing the content.

Override this method in custom subclasses.

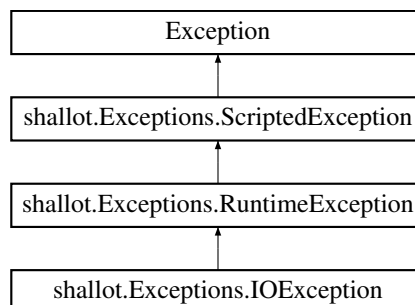
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.69 shallot.Exceptions.RuntimeException Class Reference

Shallot exception for failed operation due to (often external) runtime effects.

Inheritance diagram for `shallot.Exceptions.RuntimeException`:



## Public Member Functions

- `def __init__ (self, details=None, message=None, isRetryable=None, autoRetryRecommendedIn=-1, detailsAreInteresting=None, _class="")`  
*See [shallot.Exceptions.ScriptedException.\\_\\_init\\_\\_](#) for details.*
- `def isExceptionClass (self, classname)`  
*Checks if this exception is instance of a given exception class.*

### 1.69.1 Detailed Description

Shallot exception for failed operation due to (often external) runtime effects.

It allows resume and optionally allows retry (special cases may override each of them). Read [more about Shallot Exceptions](#).

### 1.69.2 Member Function Documentation

#### 1.69.2.1 isExceptionClass()

```
def shallot.Exceptions.ScriptedException.isExceptionClass (
    self,
    classname ) [inherited]
```

Checks if this exception is instance of a given exception class.

#### Parameters

<i>classname</i>	A exception class name (as string).
------------------	-------------------------------------

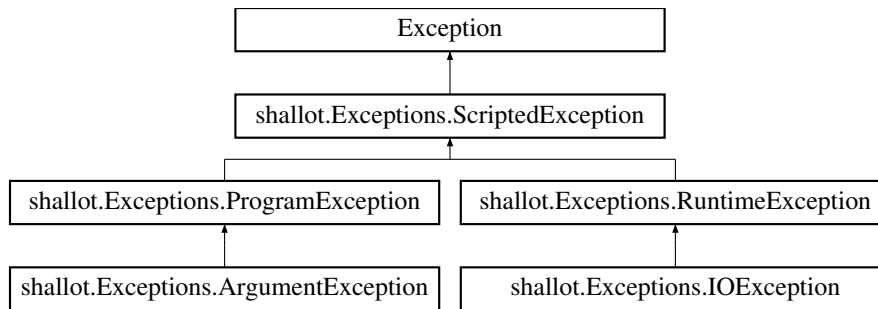
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.70 shallot.Exceptions.ScriptedException Class Reference

The Shallot exception class.

Inheritance diagram for `shallot.Exceptions.ScriptedException`:



## Public Member Functions

- `def __init__ (self, details=None, message=None, isRuntime=None, isResumeable=None, isRetryable=None, autoRetryRecommendedIn=-1, detailsAreInteresting=None, _class="")`
- `def isExceptionClass (self, classname)`  
*Checks if this exception is instance of a given exception class.*

### 1.70.1 Detailed Description

The Shallot exception class.

### 1.70.2 Constructor & Destructor Documentation

#### 1.70.2.1 \_\_init\_\_()

```

def shallot.Exceptions.ScriptedException.__init__ (
    self,
    details = None,
    message = None,
    isRuntime = None,
    isResumeable = None,
    isRetryable = None,
    autoRetryRecommendedIn = -1,
    detailsAreInteresting = None,
    _class = "" )

```

#### Parameters

<i>details</i>	Detail text.
<i>message</i>	Message text.
<i>isRuntime</i>	Is a runtime error (instead of a program logic error).
<i>isResumeable</i>	Is resumeable.
<i>isRetryable</i>	Is retryable.
<i>detailsAreInteresting</i>	If the details contain information which is directly interesting for (and consumable by) the end user.
<i>autoRetryRecommendedIn</i>	Recommended retry interval in milliseconds.
<i>_class</i>	Only used internally.

### 1.70.3 Member Function Documentation

#### 1.70.3.1 isExceptionClass()

```
def shallot.Exceptions.ScriptedException.isExceptionClass (
    self,
    classname )
```

Checks if this exception is instance of a given exception class.

##### Parameters

<code>classname</code>	A exception class name (as string).
------------------------	-------------------------------------

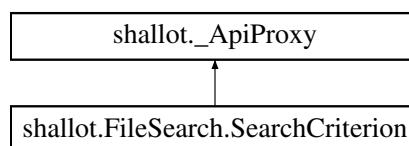
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.71 shallot.FileSearch.SearchCriterion Class Reference

Abstract base class for a search criterion.

Inheritance diagram for shallot.FileSearch.SearchCriterion:



### Public Member Functions

- def `__init__` (self, factory)  
*Each implementation must offer a constructor with this signature.*
- def `configure` (self, info)  
*Asks some questions to the user (with `info` parameter) in order to determine a configuration and returns it as a list of strings.*
- def `searchspec` (self)  
*Returns the current search configuration as string list.*
- def `match` (self, operation, eurl)  
*Determines if a file matches the configured search criteria (returns bool).*

### 1.71.1 Detailed Description

Abstract base class for a search criterion.

It implements custom file search filters. Implement this class as well as [shallot.FileSearch.SearchCriterionFactory](#) and register this pair with [shallot.FileSearch.SearchCriterionFactory.register](#).

### 1.71.2 Constructor & Destructor Documentation

#### 1.71.2.1 `__init__()`

```
def shallot.FileSearch.SearchCriterion.__init__ (
    self,
    factory )
```

Each implementation must offer a constructor with this signature.

It must forward the `factory` parameter to this constructor.

### 1.71.3 Member Function Documentation

#### 1.71.3.1 `configure()`

```
def shallot.FileSearch.SearchCriterion.configure (
    self,
    info )
```

Asks some questions to the user (with `info` parameter) in order to determine a configuration and returns it as a list of strings.

Some other methods must be able to interpret this list in order to get back this configuration.

Override this method in custom subclasses.

#### Parameters

<i>info</i>	A <a href="#">shallot.Actions.ExecutionInfo</a> execution info object.
-------------	--



## 1.71.3.2 match()

```
def shallot.FileSearch.SearchCriterion.match (
    self,
    operation,
    eurl )
```

Determines if a file matches the configured search criteria (returns bool).

Override this method in custom subclasses. Use searchspec for getting the current configuration.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>eurl</i>	The <a href="#">shallot.Eurl</a> on which the filter logic must act.

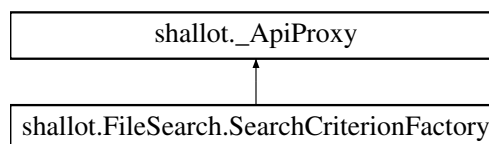
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.72 shallot.FileSearch.SearchCriterionFactory Class Reference

A factory for a [shallot.FileSearch.SearchCriterion](#) class.

Inheritance diagram for shallot.FileSearch.SearchCriterionFactory:



## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, key, ctype, description)  
*Call this constructor from subclasses.*
- def [get\\_searchspec\\_description](#) (self, val)  
*Returns the description of a search configuration (for displaying in the user interface).*
- def [is\\_visible\\_for](#) (self, operation, node)  
*Determines if this search criterion should be offered for a node.*

## Static Public Member Functions

- def [register](#) (criterionfactory, positionGroup=None, positionIndex=None)  
*Registers a custom [shallot.FileSearch.SearchCriterionFactory](#) instance for offering custom file searching filters.*

## Static Public Attributes

- `INDEX_CORE` = `internalstuff.searchcriterionfactory_index_core()`  
*Base index value for core (i.e.*
- `INDEX_NORMAL` = `internalstuff.searchcriterionfactory_index_normal()`  
*Base index value for search criteria with normal importance.*
- `INDEX_EXOTIC` = `internalstuff.searchcriterionfactory_index_exotic()`  
*Base index value for exotic search criteria.*

### 1.72.1 Detailed Description

A factory for a `shallot.FileSearch.SearchCriterion` class.

### 1.72.2 Constructor & Destructor Documentation

#### 1.72.2.1 `__init__()`

```
def shallot.FileSearch.SearchCriterionFactory.__init__ (
    self,
    key,
    ctype,
    description )
```

Call this constructor from subclasses.

#### Parameters

<i>key</i>	A short string used as key for this criterion.
<i>ctype</i>	The type of your <code>shallot.FileSearch.SearchCriterion</code> implementation to create.
<i>description</i>	The short description (as shown in menus).

### 1.72.3 Member Function Documentation

#### 1.72.3.1 `get_searchspec_description()`

```
def shallot.FileSearch.SearchCriterionFactory.get_searchspec_description (
    self,
    val )
```

Returns the description of a search configuration (for displaying in the user interface).

Override this method in custom subclasses or leave the default implementation.

## Parameters

<i>val</i>	The search configuration as string list.
------------	--

## 1.72.3.2 is\_visible\_for()

```
def shallot.FileSearch.SearchCriterionFactory.is_visible_for (
    self,
    operation,
    node )
```

Determines if this search criterion should be offered for a node.

Override this method in custom subclasses or leave the default implementation.

## Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>node</i>	The <a href="#">shallot.Filesystem.Node</a> node for which the visibility must be determined.

## 1.72.3.3 register()

```
def shallot.FileSearch.SearchCriterionFactory.register (
    criterionfactory,
    positionGroup = None,
    positionIndex = None ) [static]
```

Registers a custom [shallot.FileSearch.SearchCriterionFactory](#) instance for offering custom file searching filters.

## Parameters

<i>criterionfactory</i>	An instance of <a href="#">shallot.FileSearch.SearchCriterionFactory</a> .
<i>positionGroup</i>	Controls display order. See <a href="#">Position Indexes</a> for details. Use one of the INDEX_* values from <a href="#">shallot.FileSearch.SearchCriterionFactory</a> .
<i>positionIndex</i>	Controls display order. See <a href="#">Position Indexes</a> for details.

## 1.72.4 Member Data Documentation

#### 1.72.4.1 INDEX\_CORE

```
shallot.FileSearch.SearchCriterionFactory.INDEX_CORE = internalstuff.searchcriterionfactory_↵
index_core() [static]
```

Base index value for core (i.e.

very interesting) search criteria.

#### 1.72.4.2 INDEX\_EXOTIC

```
shallot.FileSearch.SearchCriterionFactory.INDEX_EXOTIC = internalstuff.searchcriterionfactory_↵
_index_exotic() [static]
```

Base index value for exotic search criteria.

#### 1.72.4.3 INDEX\_NORMAL

```
shallot.FileSearch.SearchCriterionFactory.INDEX_NORMAL = internalstuff.searchcriterionfactory_↵
_index_normal() [static]
```

Base index value for search criteria with normal importance.

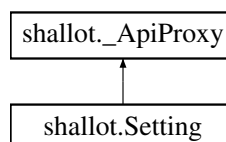
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

### 1.73 shallot.Setting Class Reference

Abstract base class for a scripted Shallot setting (those who have to be stored manually).

Inheritance diagram for shallot.Setting:



#### Classes

- class [GroupInfo](#)

*Enumeration of groups to which a setting can belong.*

## Public Member Functions

- def `__init__` (self, name, description, group, isAdvancedSetting=False, isGlobal=False, isPerFileview=False)
- def `set_value` (self, value, viewindex=None)  
*Called from Shallot core when the value was set.*
- def `get_value` (self, viewindex)  
*Get the currently set value.*
- def `value_description` (self, val)  
*Gets a human readable description text for a value.*

## Static Public Member Functions

- def `register` (setting)  
*Registers a [shallot.Setting](#) instance in Shallot.*

### 1.73.1 Detailed Description

Abstract base class for a scripted Shallot setting (those who have to be stored manually).

See Shallot documentation for details.

### 1.73.2 Constructor & Destructor Documentation

#### 1.73.2.1 `__init__()`

```
def shallot.Setting.__init__ (
    self,
    name,
    description,
    group,
    isAdvancedSetting = False,
    isGlobal = False,
    isPerFileview = False )
```

#### Parameters

<i>name</i>	Internal name (must be unique).
<i>description</i>	Description text.
<i>group</i>	Group. One of <a href="#">shallot.Setting.GroupInfo</a> .
<i>isAdvancedSetting</i>	Is this an advanced setting?
<i>isGlobal</i>	Does this setting regard global aspects (instead of per-directory aspects)?
<i>isPerFileview</i>	Does this setting regard per-fileview aspects (instead of per-mainwindow)?

### 1.73.3 Member Function Documentation

#### 1.73.3.1 `get_value()`

```
def shallot.Setting.get_value (
    self,
    viewindex )
```

Get the currently set value.

Override this method in custom subclasses.

##### Parameters

<i>viewindex</i>	View index. Will be <code>Nothing</code> for per-mainwindow settings. Otherwise a fileview index.
------------------	---

#### 1.73.3.2 `register()`

```
def shallot.Setting.register (
    setting ) [static]
```

Registers a [shallot.Setting](#) instance in Shallot.

##### Parameters

<i>setting</i>	The <a href="#">shallot.Setting</a> setting object.
----------------	---

#### 1.73.3.3 `set_value()`

```
def shallot.Setting.set_value (
    self,
    value,
    viewindex = None )
```

Called from Shallot core when the value was set.

Override this method in custom subclasses.

## Parameters

<i>value</i>	The value.
<i>viewindex</i>	View index. Will be <code>Nothing</code> for per-mainwindow settings. Otherwise a fileview index.

## 1.73.3.4 value\_description()

```
def shallot.Setting.value_description (
    self,
    val )
```

Gets a human readable description text for a value.

Override this method in custom subclasses.

## Parameters

<i>val</i>	The value.
------------	------------

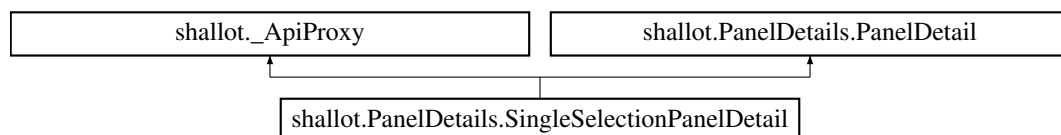
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.74 shallot.PanelDetails.SingleSelectionPanelDetail Class Reference

Abstract base class for a detail panel entry, which occurs when one item is selected.

Inheritance diagram for shallot.PanelDetails.SingleSelectionPanelDetail:



## Public Member Functions

- def `__init__` (self, positionGroup=None, positionIndex=None, valueWidthHint=4)
- def `set_value` (self, detail, node, operation)
 

*This method is called whenever an output must be determined for a certain node.*
- def `link_triggered` (self, node, linktarget)
 

*This method is called whenever the user triggers a link.*

## Static Public Member Functions

- `def register (detail)`  
Registers a [shallot.PanelDetails.PanelDetail](#) in the Shallot core.

## Static Public Attributes

- `INDEX_VERYINTERESTING` = `internalstuff.paneldetail_positionindex_veryinteresting()`  
A integer value to be used as base display index for very interesting details.
- `INDEX_INTERESTING` = `internalstuff.paneldetail_positionindex_interesting()`  
A integer value to be used as base display index for interesting details.
- `INDEX_EXOTIC` = `internalstuff.paneldetail_positionindex_exotic()`  
A integer value to be used as base display index for exotic details.

### 1.74.1 Detailed Description

Abstract base class for a detail panel entry, which occurs when one item is selected.

See [shallot.PanelDetails.PanelDetail.register](#) for registering custom implementations to shallot.

### 1.74.2 Constructor & Destructor Documentation

#### 1.74.2.1 `__init__()`

```
def shallot.PanelDetails.SingleSelectionPanelDetail.__init__ (
    self,
    positionGroup = None,
    positionIndex = None,
    valueWidthHint = 4 )
```

#### Parameters

<i>positionGroup</i>	Controls display order. See <a href="#">Position Indexes</a> for details. Use one of the <code>INDEX_*</code> values from <a href="#">shallot.PanelDetails.PanelDetail</a> .
<i>positionIndex</i>	Controls display order. See <a href="#">Position Indexes</a> for details.
<i>valueWidthHint</i>	A width in centimeters to reserve for printing the values.

### 1.74.3 Member Function Documentation



### 1.74.3.1 link\_triggered()

```
def shallot.PanelDetails.SingleSelectionPanelDetail.link_triggered (
    self,
    node,
    linktarget )
```

This method is called whenever the user triggers a link.

Override in custom implementations if links are used.

#### Parameters

<i>node</i>	The selected <a href="#">shallot.Filesystem.Node</a> .
<i>linktarget</i>	The link target string, as specified in <a href="#">shallot.PanelDetails.PanelDetailValueElementButton.__init__</a> .

### 1.74.3.2 register()

```
def shallot.PanelDetails.PanelDetail.register (
    detail ) [static], [inherited]
```

Registers a [shallot.PanelDetails.PanelDetail](#) in the Shallot core.

#### Parameters

<i>detail</i>	The <a href="#">shallot.PanelDetails.PanelDetail</a> instance.
---------------	--

### 1.74.3.3 set\_value()

```
def shallot.PanelDetails.SingleSelectionPanelDetail.set_value (
    self,
    detail,
    node,
    operation )
```

This method is called whenever an output must be determined for a certain node.

It must return a list of 2-tuples (one for each row). Those are each name/value pairs with the label of that row and the value (which is a list of [shallot.PanelDetails.PanelDetails.AbstractPanelDetailValueElement](#)). For large waiting times, you should just return a value list only containing a [shallot.PanelDetails.PanelDetailValueElementWaiting](#) for your rows, start an asynchronous execution and use [shallot.PanelDetailInst.set\\_row](#) in the end.

Override this method in custom subclasses.

**Parameters**

<i>detail</i>	A shallot.PanelDetailInst object for printing the details. Only used in advanced cases.
<i>node</i>	The selected <a href="#">shallot.Filesystem.Node</a> .
<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.

**1.74.4 Member Data Documentation****1.74.4.1 INDEX\_EXOTIC**

```
shallot.PanelDetails.PanelDetail.INDEX_EXOTIC = internalstuff.paneldetail_positionindex↔
exotic() [static], [inherited]
```

A integer value to be used as base display index for exotic details.

**1.74.4.2 INDEX\_INTERESTING**

```
shallot.PanelDetails.PanelDetail.INDEX_INTERESTING = internalstuff.paneldetail_positionindex↔
_interesting() [static], [inherited]
```

A integer value to be used as base display index for interesting details.

**1.74.4.3 INDEX\_VERYINTERESTING**

```
shallot.PanelDetails.PanelDetail.INDEX_VERYINTERESTING = internalstuff.paneldetail_positionindex↔
_veryinteresting() [static], [inherited]
```

A integer value to be used as base display index for very interesting details.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

**1.75 shallot.Streaming Class Reference**

[Streaming](#) binary data.

## Classes

- class [BlobReadDataDevice](#)  
*A binary source backed by static binary content in a byte array.*
- class [ReadDataDevice](#)  
*A scripted source of binary content.*
- class [StreamReadDataDevice](#)  
*A binary source backed by a file object.*

### 1.75.1 Detailed Description

[Streaming](#) binary data.

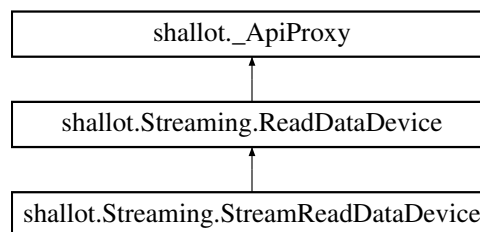
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.76 shallot.Streaming.StreamReadDataDevice Class Reference

A binary source backed by a file object.

Inheritance diagram for shallot.Streaming.StreamReadDataDevice:



## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, stream)
- def [getdata](#) (self)  
*See [shallot.Streaming.ReadDataDevice](#).*
- def [read](#) (self)  
*Reads a chunk of data as Python byte array.*
- def [readall](#) (self)  
*Reads the complete data as Python byte array.*

## Public Attributes

- **stream**

### 1.76.1 Detailed Description

A binary source backed by a file object.

### 1.76.2 Constructor & Destructor Documentation

#### 1.76.2.1 `__init__()`

```
def shallot.Streaming.StreamReadDataDevice.__init__ (
    self,
    stream )
```

##### Parameters

<code>stream</code>	The file object containing the content.
---------------------	---

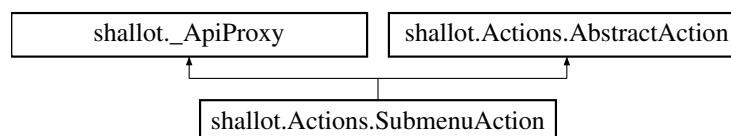
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

## 1.77 shallot.Actions.SubmenuItem Class Reference

Abstract base class for a submenu action, which can be made visible in menus or the toolbar.

Inheritance diagram for shallot.Actions.SubmenuItem:



### Public Member Functions

- `def __init__ (self, text, enabled=True, icon="", defaultActionPrecedence=0)`
- `def set_subitems (self, subitems)`  
*Sets the subitems.*
- `def initialize (self)`  
*Initialize the action.*
- `def set_enabled (self, value)`  
*Sets if the item is enabled.*
- `def set_visible (self, value)`  
*Sets the visibility of this item.*
- `def enabled (self)`  
*Checks if this action is enabled.*
- `def visible (self)`  
*Checks the visibility of this item (non-recursively).*

### 1.77.1 Detailed Description

Abstract base class for a submenu action, which can be made visible in menus or the toolbar.

See [shallot.Actions.AbstractAction](#) for more.

### 1.77.2 Constructor & Destructor Documentation

#### 1.77.2.1 `__init__()`

```
def shallot.Actions.SubmenuItem.__init__ (
    self,
    text,
    enabled = True,
    icon = "",
    defaultActionPrecedence = 0 )
```

##### Parameters

<i>text</i>	A label text.
<i>enabled</i>	If this action is enabled.
<i>icon</i>	Name of an icon.
<i>defaultActionPrecedence</i>	Precedence value for being a default action. This int value must be higher than all others for becoming the default action. See <a href="#">shallot.Actions.DefaultPrecedenceValues</a> .

### 1.77.3 Member Function Documentation

#### 1.77.3.1 `initialize()`

```
def shallot.Actions.AbstractAction.initialize (
    self ) [inherited]
```

Initialize the action.

This should make the time-consuming parts, e.g. for determining a label or enabled state.

Override this method in custom subclasses or leave the default implementation.

#### 1.77.3.2 `set_enabled()`

```
def shallot.Actions.AbstractAction.set_enabled (
    self,
    value ) [inherited]
```

Sets if the item is enabled.

## Parameters

<i>value</i>	The new value.
--------------	----------------

1.77.3.3 `set_subitems()`

```
def shallot.Actions.SubmenuItem.set_subitems (
    self,
    subitems )
```

Sets the subitems.

## Parameters

<i>subitems</i>	A list of <a href="#">shallot.Actions.AbstractAction</a> instances.
-----------------	---

1.77.3.4 `set_visible()`

```
def shallot.Actions.AbstractAction.set_visible (
    self,
    value ) [inherited]
```

Sets the visibility of this item.

## Parameters

<i>value</i>	The new value.
--------------	----------------

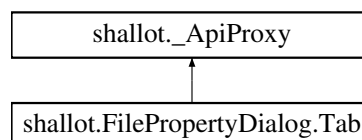
The documentation for this class was generated from the following file:

- `/home/pino/projects/shallot/shallot.py`

1.78 `shallot.FilePropertyDialog.Tab` Class Reference

Abstract base class for a tab in the Properties dialog.

Inheritance diagram for `shallot.FilePropertyDialog.Tab`:



## Classes

- class [PropertyButtonConfig](#)  
Definitions for buttons in a [shallot.FilePropertyDialog.Tab.PropertyConfig](#).
- class [PropertyConfig](#)  
Definitions for properties, which present a piece of information in a [shallot.FilePropertyDialog.Tab](#).
- class [PropertyType](#)  
A type of a single property.

## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, title, properties)
- def [refresh](#) (self)  
Reloads the content in the dialog (typically called after the external situation has changed so the dialog shows out-dated information).
- def [selected\\_index\\_for\\_property](#) (self, index)  
For a [shallot.FilePropertyDialog.Tab.PropertyType.StringMap](#) property, this returns the index of the row selected by the user in the dialog (or -1 for no selection).
- def [nodes](#) (self)  
Returns a list of [shallot.Filesystem.Node](#) containing the nodes to show (typically one).
- def [update\\_widget](#) (self, i, operation)  
Provides the actual content.

## Static Public Member Functions

- def [register](#) (tabclass, positionGroup=None, positionIndex=None)  
Registers a [shallot.FilePropertyDialog.Tab](#) implementation for making its content available in the Properties dialog.

## Public Attributes

- [properties](#)
- [btnid2btn](#)

## Static Public Attributes

- [INDEX\\_CORE](#) = [internalstuff.filepropertydialogtab\\_index\\_core\(\)](#)  
An integer value to be used as base value for specifying order indexes of core (i.e.
- [INDEX\\_VERYIMPORTANT](#) = [internalstuff.filepropertydialogtab\\_index\\_veryimportant\(\)](#)  
An integer value to be used as base value for specifying order indexes of very important properties.
- [INDEX\\_IMPORTANT](#) = [internalstuff.filepropertydialogtab\\_index\\_important\(\)](#)  
An integer value to be used as base value for specifying order indexes of important properties.
- [INDEX\\_NORMAL](#) = [internalstuff.filepropertydialogtab\\_index\\_normal\(\)](#)  
An integer value to be used as base value for specifying order indexes of properties with normal importance.
- [INDEX\\_EXOTIC](#) = [internalstuff.filepropertydialogtab\\_index\\_exotic\(\)](#)  
An integer value to be used as base value for specifying order indexes of exotic (i.e.

### 1.78.1 Detailed Description

Abstract base class for a tab in the Properties dialog.

It provides content and can also offer user interactions. See [shallot.FilePropertyDialog.Tab.register](#) for registering custom implementations to shallot.

### 1.78.2 Constructor & Destructor Documentation

#### 1.78.2.1 `__init__()`

```
def shallot.FilePropertyDialog.Tab.__init__ (
    self,
    title,
    properties )
```

##### Parameters

<i>title</i>	The title text for this tab.
<i>properties</i>	A list of <a href="#">shallot.FilePropertyDialog.Tab.PropertyConfig</a> instances. Each instances specifies one piece of information you want to provide.

### 1.78.3 Member Function Documentation

#### 1.78.3.1 `register()`

```
def shallot.FilePropertyDialog.Tab.register (
    tabclass,
    positionGroup = None,
    positionIndex = None ) [static]
```

Registers a [shallot.FilePropertyDialog.Tab](#) implementation for making its content available in the Properties dialog.

##### Parameters

<i>positionGroup</i>	Controls display order. See <a href="#">Position Indexes</a> for details. Use one of the <code>INDEX_*</code> values from <a href="#">shallot.FilePropertyDialog.Tab</a> .
<i>positionIndex</i>	Controls display order. See <a href="#">Position Indexes</a> for details.
<i>tabclass</i>	A subclass of <a href="#">shallot.FilePropertyDialog.Tab</a> .



1.78.3.2 `selected_index_for_property()`

```
def shallot.FilePropertyDialog.Tab.selected_index_for_property (
    self,
    index )
```

For a [shallot.FilePropertyDialog.Tab.PropertyType.StringMap](#) property, this returns the index of the row selected by the user in the dialog (or -1 for no selection).

It is not allowed to call it for other properties.

1.78.3.3 `update_widget()`

```
def shallot.FilePropertyDialog.Tab.update_widget (
    self,
    i,
    operation )
```

Provides the actual content.

Returns a value depending on the type choice in [shallot.FilePropertyDialog.Tab.PropertyConfig](#).

Override this method in custom subclasses.

## Parameters

<i>i</i>	The index (in the same order as the <code>properties</code> constructor parameter).
<i>operation</i>	A <a href="#">shallot.Operations.Operation</a> instance.

## 1.78.4 Member Data Documentation

1.78.4.1 `INDEX_CORE`

```
shallot.FilePropertyDialog.Tab.INDEX_CORE = internalstuff.filepropertydialogtab_index_core()
[static]
```

An integer value to be used as base value for specifying order indexes of core (i.e. maximum important) properties.

1.78.4.2 `INDEX_EXOTIC`

```
shallot.FilePropertyDialog.Tab.INDEX_EXOTIC = internalstuff.filepropertydialogtab_index_↔
exotic() [static]
```

An integer value to be used as base value for specifying order indexes of exotic (i.e. less important) properties.

### 1.78.4.3 INDEX\_IMPORTANT

```
shallot.FilePropertyDialog.Tab.INDEX_IMPORTANT = internalstuff.filepropertydialogtab_index_↵
important() [static]
```

An integer value to be used as base value for specifying order indexes of important properties.

### 1.78.4.4 INDEX\_NORMAL

```
shallot.FilePropertyDialog.Tab.INDEX_NORMAL = internalstuff.filepropertydialogtab_index_↵
normal() [static]
```

An integer value to be used as base value for specifying order indexes of properties with normal importance.

### 1.78.4.5 INDEX\_VERYIMPORTANT

```
shallot.FilePropertyDialog.Tab.INDEX_VERYIMPORTANT = internalstuff.filepropertydialogtab_↵
index_veryimportant() [static]
```

An integer value to be used as base value for specifying order indexes of very important properties.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.79 shallot.FilePropertyDialog.TabPropertyIconTextBanner Class Reference

Represents values for image/text banners as used for [shallot.FilePropertyDialog.Tab.PropertyType.IconTextBanner](#).

### Public Member Functions

- `def __init__(self)`
- `def add_text(self, text)`  
*Adds text to the banner.*
- `def add_icon(self, iconname, size=1.0)`  
*Adds an icon to the banner.*

### Public Attributes

- `items`

### 1.79.1 Detailed Description

Represents values for image/text banners as used for [shallot.FilePropertyDialog.Tab.PropertyType.IconTextBanner](#).

### 1.79.2 Member Function Documentation

#### 1.79.2.1 add\_icon()

```
def shallot.FilePropertyDialog.TabPropertyIconTextBanner.add_icon (
    self,
    iconname,
    size = 1.0 )
```

Adds an icon to the banner.

##### Parameters

<i>iconname</i>	The name of the icon to append.
<i>size</i>	Icon size (not in pixels but relative to the default).

#### 1.79.2.2 add\_text()

```
def shallot.FilePropertyDialog.TabPropertyIconTextBanner.add_text (
    self,
    text )
```

Adds text to the banner.

##### Parameters

<i>text</i>	The text to append.
-------------	---------------------

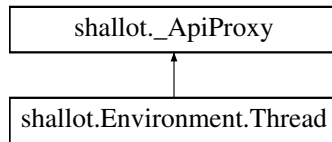
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.80 shallot.Environment.Thread Class Reference

A thread executes code asynchronously.

Inheritance diagram for shallot.Environment.Thread:



## Public Member Functions

- `def __init__(self)`
- `def run(self)`  
*Contains the code to execute in this thread.*
- `def start(self)`  
*Starts the execution of this thread and calls 'run' in that thread.*

## Static Public Member Functions

- `def execute_threaded(fct, args, kwargs)`  
*Executes a function in a new [shallot.Environment.Thread](#) (and returns that).*

### 1.80.1 Detailed Description

A thread executes code asynchronously.

See also the Python documentation about threading. Although this thread class is not related to the Python thread class (in terms of object orientation), you should understand the general pitfalls which come with multithreading and you might find the Python synchronization tools useful for avoiding those pitfalls. Note: It is strictly forbidden to 'park' a thread and wait for some external event most of the time. Use a [shallot.Environment.Timer](#) for recurring tasks.

### 1.80.2 Member Function Documentation

#### 1.80.2.1 execute\_threaded()

```
def shallot.Environment.Thread.execute_threaded (
    fct,
    args,
    kwargs ) [static]
```

Executes a function in a new [shallot.Environment.Thread](#) (and returns that).

This avoids subclassing and makes the code more compact. 'args' and 'kwargs' are additional parameters which become the parameters in the actual call of 'fct'.

#### Parameters

<i>fct</i>	The function to execute.
------------	--------------------------

### 1.80.2.2 run()

```
def shallot.Environment.Thread.run (
    self )
```

Contains the code to execute in this thread.

Override this method in custom subclasses.

### 1.80.2.3 start()

```
def shallot.Environment.Thread.start (
    self )
```

Starts the execution of this thread and calls 'run' in that thread.

Returns immediately.

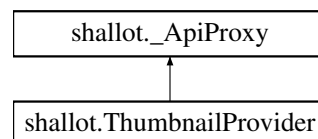
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.81 shallot.ThumbnailProvider Class Reference

Abstract base class for a [thumbnail](#) provider.

Inheritance diagram for shallot.ThumbnailProvider:



### Public Member Functions

- def **\_\_init\_\_** (self)
- def [get\\_thumbnail](#) (self, operation, node, contenttype, width, height)  
*Returns the thumbnail.*

## Static Public Member Functions

- `def register` (thumbnailprovider, positionGroup=None, positionIndex=None)  
*Registers a custom [shallot.ThumbnailProvider](#) instance for offering custom thumbnails.*

## Static Public Attributes

- `INDEX_CORE` = `internalstuff.thumbnailprovider_index_core()`  
*Base index value for core (i.e.*
- `INDEX_NORMAL` = `internalstuff.thumbnailprovider_index_normal()`  
*Base index value for thumbnail providers with normal importance.*
- `INDEX_EXOTIC` = `internalstuff.thumbnailprovider_index_exotic()`  
*Base index value for exotic thumbnail providers.*
- `INDEX_FALLBACK` = `internalstuff.thumbnailprovider_index_fallback()`  
*Base index value for fallback thumbnail providers (executed after everything else failed).*

### 1.81.1 Detailed Description

Abstract base class for a [thumbnail](#) provider.

It can implement new ways of generating thumbnail images for files. Implement this class and register an instance with [shallot.ThumbnailProvider.register](#).

### 1.81.2 Member Function Documentation

#### 1.81.2.1 `get_thumbnail()`

```
def shallot.ThumbnailProvider.get_thumbnail (
    self,
    operation,
    node,
    contenttype,
    width,
    height )
```

Returns the thumbnail.

It must be a bytestring containing the image in a well known format (png, jpeg, svg, ...).

Override this method in custom subclasses.

#### Parameters

<i>operation</i>	The <a href="#">shallot.Operations.Operation</a> operation object.
<i>node</i>	The <a href="#">shallot.Filesystem.Node</a> filesystem node to get a thumbnail for.
<i>contenttype</i>	The content type of the original file.
<i>width</i>	The requested thumbnail width in pixels.
<i>height</i>	The requested thumbnail height in pixels.

### 1.81.2.2 register()

```
def shallot.ThumbnailProvider.register (
    thumbnailprovider,
    positionGroup = None,
    positionIndex = None ) [static]
```

Registers a custom [shallot.ThumbnailProvider](#) instance for offering custom thumbnails.

#### Parameters

<i>thumbnailprovider</i>	An instance of <a href="#">shallot.ThumbnailProvider</a> .
<i>positionGroup</i>	Controls execution order. See <a href="#">Position Indexes</a> for details. Use one of the INDEX_* values from <a href="#">shallot.ThumbnailProvider</a> .
<i>positionIndex</i>	Controls execution order. See <a href="#">Position Indexes</a> for details.

## 1.81.3 Member Data Documentation

### 1.81.3.1 INDEX\_CORE

```
shallot.ThumbnailProvider.INDEX_CORE = internalstuff.thumbnailprovider_index_core() [static]
```

Base index value for core (i.e.

very high priority) thumbnail providers.

### 1.81.3.2 INDEX\_EXOTIC

```
shallot.ThumbnailProvider.INDEX_EXOTIC = internalstuff.thumbnailprovider_index_exotic() [static]
```

Base index value for exotic thumbnail providers.

### 1.81.3.3 INDEX\_FALLBACK

```
shallot.ThumbnailProvider.INDEX_FALLBACK = internalstuff.thumbnailprovider_index_fallback()
[static]
```

Base index value for fallback thumbnail providers (executed after everything else failed).

### 1.81.3.4 INDEX\_NORMAL

```
shallot.ThumbnailProvider.INDEX_NORMAL = internalstuff.thumbnailprovider_index_normal() [static]
```

Base index value for thumbnail providers with normal importance.

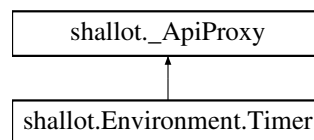
The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py

## 1.82 shallot.Environment.Timer Class Reference

A timers executes some code recurringly in some time interval.

Inheritance diagram for shallot.Environment.Timer:



### Public Member Functions

- def `__init__`(self)
- def `run`(self)  
*Contains the code to execute.*
- def `start`(self, interval)  
*Starts the timer, which executes 'run' each 'interval' milliseconds.*
- def `stop`(self)  
*Stops the timer.*
- def `is_started`(self)  
*Returns if the timer is currently active.*

### Static Public Member Functions

- def `start_in_timer`(fct, interval, args, kwargs)  
*Starts a function in a `shallot.Environment.Timer` (and returns that).*

### 1.82.1 Detailed Description

A timers executes some code recurringly in some time interval.

### 1.82.2 Member Function Documentation



### 1.82.2.1 run()

```
def shallot.Environment.Timer.run (
    self )
```

Contains the code to execute.

Override this method in custom subclasses.

### 1.82.2.2 start()

```
def shallot.Environment.Timer.start (
    self,
    interval )
```

Starts the timer, which executes 'run' each 'interval' milliseconds.

Returns immediately.

#### Parameters

<i>interval</i>	The interval in milliseconds.
-----------------	-------------------------------

### 1.82.2.3 start\_in\_timer()

```
def shallot.Environment.Timer.start_in_timer (
    fct,
    interval,
    args,
    kwargs ) [static]
```

Starts a function in a [shallot.Environment.Timer](#) (and returns that).

This avoids subclassing and makes the code more compact. 'args' and 'kwargs' are additional parameters which become the parameters in the actual call of 'fct'.

#### Parameters

<i>fct</i>	The function to execute.
<i>interval</i>	The interval in milliseconds.

The documentation for this class was generated from the following file:

- /home/pino/projects/shallot/shallot.py



# Index

- `__init__`
  - `shallot::Actions::ActionAction`, 21
  - `shallot::Actions::SubmenuAction`, 147
  - `shallot::ConfigurationValue`, 35
  - `shallot::DetailColumn`, 40
  - `shallot::Exceptions::ScriptedException`, 132
  - `shallot::FilePropertyDialog::Tab`, 150
  - `shallot::FilePropertyDialog::Tab::PropertyButton`↔  
Config, 127
  - `shallot::FilePropertyDialog::Tab::PropertyConfig`, 128
  - `shallot::FileSearch::SearchCriterion`, 134
  - `shallot::FileSearch::SearchCriterionFactory`, 136
  - `shallot::IntlStringMap`, 96
  - `shallot::IntlStringMap::IntlString`, 95
  - `shallot::Operations::FilesystemOperationProgress`↔  
Monitor, 74
  - `shallot::PanelDetails::MultiSelectionPanelDetail`, 104
  - `shallot::PanelDetails::PanelDetailValueElement`↔  
Button, 121
  - `shallot::PanelDetails::PanelDetailValueElement`↔  
Icon, 122
  - `shallot::PanelDetails::PanelDetailValueElement`↔  
String, 123
  - `shallot::PanelDetails::SingleSelectionPanelDetail`, 142
  - `shallot::Setting`, 139
  - `shallot::Streaming::BlobReadDataDevice`, 27
  - `shallot::Streaming::StreamReadDataDevice`, 146
- `action`
  - `shallot::Actions::ActionAction`, 22
- `add_bookmark`
  - `shallot::Bookmarks`, 29
- `add_changed_eurl`
  - `shallot::Actions::ExecutionInfo`, 53
- `add_detail`
  - `shallot::Filesystem::Node`, 107
- `add_icon`
  - `shallot::FilePropertyDialog::TabPropertyIconText`↔  
Banner, 153
- `add_item`
  - `shallot::Filesystem::NodeList`, 108
- `add_text`
  - `shallot::FilePropertyDialog::TabPropertyIconText`↔  
Banner, 153
- `apply_value`
  - `shallot::DetailColumn`, 40
- `as_string`
  - `shallot::Eurl`, 46
- `basename`
  - `shallot::Eurl`, 46
- `begin_iterative_adding`
  - `shallot::Filesystem::NodeList`, 108
- `Behavior`
  - `shallot::ConfigurationValue::Category`, 34
  - `shallot::Setting::GroupInfo`, 77
- `BuiltinSpecialOpen`
  - `shallot::Actions::DefaultPrecedenceValues`, 38
- `can_copy_item`
  - `shallot::Operations::FilesystemOperation`, 63
- `can_create_directory`
  - `shallot::Filesystem::Handler`, 79
  - `shallot::Operations::FilesystemOperation`, 63
- `can_create_file`
  - `shallot::Filesystem::Handler`, 80
  - `shallot::Operations::FilesystemOperation`, 64
- `can_create_link`
  - `shallot::Filesystem::Handler`, 80
  - `shallot::Operations::FilesystemOperation`, 64
- `can_delete_item`
  - `shallot::Filesystem::Handler`, 81
  - `shallot::Operations::FilesystemOperation`, 64
- `can_get_filecontent`
  - `shallot::Filesystem::Handler`, 81
  - `shallot::Operations::FilesystemOperation`, 65
- `can_move_item`
  - `shallot::Operations::FilesystemOperation`, 65
- `can_rename_item`
  - `shallot::Filesystem::Handler`, 81
- `Cancel`
  - `shallot::Actions::ExecutionUserFeedback::`↔  
MessageBoxButton, 102
- `change_bookmark`
  - `shallot::Bookmarks`, 30
- `change_bookmark_tags`
  - `shallot::Bookmarks`, 30
- `changed`
  - `shallot::Operations::FilesystemOperationProgress`↔  
Monitor, 74
- `compute_value`
  - `shallot::DetailColumn`, 41
- `configure`
  - `shallot::FileSearch::SearchCriterion`, 134
- `configure_item`
  - `shallot::Filesystem::Handler`, 82
- `Continue`

- shallot::Actions::ExecutionUserFeedback::↵  
    - MessageBoxButton, 102
- copy\_item
  - shallot::Operations::FilesystemOperation, 65
- copy\_items
  - shallot::Operations::FilesystemOperation, 66
- create
  - shallot::Eurl, 47
- create\_directory
  - shallot::Filesystem::Handler, 82
  - shallot::Operations::FilesystemOperation, 66
- create\_file
  - shallot::Filesystem::Handler, 84
  - shallot::Operations::FilesystemOperation, 66
- create\_link
  - shallot::Filesystem::Handler, 84
  - shallot::Operations::FilesystemOperation, 67
- create\_node
  - shallot::Filesystem, 58
- shallot::Setting::GroupInfo, 77
- delete\_directory\_if\_empty
  - shallot::Operations::FilesystemOperation, 67
- delete\_item
  - shallot::Filesystem::Handler, 85
  - shallot::Operations::FilesystemOperation, 67
- determine\_value
  - shallot::DetailColumn, 41
- Directory
  - shallot::Filesystem::NodeType, 109
- end\_iterative\_adding
  - shallot::Filesystem::NodeList, 108
- enwrap\_with\_outer\_url
  - shallot::Eurl, 47
- execute\_threaded
  - shallot::Environment::Thread, 154
- ExternalTools
  - shallot::ConfigurationValue::Category, 34
- FIRSTTYPE
  - shallot::Filesystem::NodeType, 110
- fetch\_container\_file
  - shallot::Operations::Operation, 114
- fetch\_file
  - shallot::Operations::Operation, 114
- File
  - shallot::Filesystem::NodeType, 109
- FileHandling
  - shallot::Setting::GroupInfo, 77
- find\_by\_name
  - shallot::DetailColumn, 41
- find\_nodes\_for\_eurl
  - shallot::Filesystem, 59
- from\_string
  - shallot::Eurl, 47
- GUI
  - shallot::ConfigurationValue::Category, 34
    - shallot::Setting::GroupInfo, 77
  - get\_actions
    - shallot::Filesystem::Handler, 85
  - get\_custom\_data
    - shallot::Operations::Operation, 114
  - get\_customattributes
    - shallot::Filesystem::Handler, 85
  - get\_detail\_from\_cache
    - shallot::Operations::Operation, 115
  - get\_extendedattribute
    - shallot::Filesystem::Handler, 86
    - shallot::Operations::FilesystemOperation, 68
  - get\_extendedattribute\_size
    - shallot::Filesystem::Handler, 86
    - shallot::Operations::FilesystemOperation, 68
  - get\_file\_content
    - shallot::Filesystem::Handler, 87
    - shallot::Operations::FilesystemOperation, 68
  - get\_linktarget
    - shallot::Filesystem::Handler, 87
    - shallot::Operations::FilesystemOperation, 69
  - get\_mimetype
    - shallot::Filesystem::Handler, 87
  - get\_mtime
    - shallot::Filesystem::Handler, 88
  - get\_or\_create\_node
    - shallot::Filesystem, 59
  - get\_searchspec\_description
    - shallot::FileSearch::SearchCriterionFactory, 136
  - get\_size
    - shallot::Filesystem::Handler, 88
  - get\_thumbnail
    - shallot::ThumbnailProvider, 156
  - get\_type
    - shallot::Filesystem::Handler, 89
    - shallot::Operations::FilesystemOperation, 69
  - get\_value
    - shallot::Setting, 140
  - getdata
    - shallot::Streaming::ReadDataDevice, 130
  - Global
    - shallot::Setting::GroupInfo, 77
  - has\_inner\_urls
    - shallot::Eurl, 48
  - has\_parent\_segment
    - shallot::Eurl, 48
  - hostname
    - shallot::Eurl, 48
  - INDEX\_CORE
    - shallot::DetailColumn, 42
    - shallot::FilePropertyDialog::Tab, 151
    - shallot::FileSearch::SearchCriterionFactory, 137
    - shallot::ThumbnailProvider, 157
  - INDEX\_EXOTIC
    - shallot::DetailColumn, 42
    - shallot::FilePropertyDialog::Tab, 151

- shallot::FileSearch::SearchCriterionFactory, [138](#)
  - shallot::PanelDetails::MultiSelectionPanelDetail, [105](#)
  - shallot::PanelDetails::PanelDetail, [118](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail, [144](#)
  - shallot::ThumbnailProvider, [157](#)
- INDEX\_FALLBACK
  - shallot::ThumbnailProvider, [157](#)
- INDEX\_IMPORTANT
  - shallot::FilePropertyDialog::Tab, [151](#)
- INDEX\_INTERESTING
  - shallot::DetailColumn, [42](#)
  - shallot::PanelDetails::MultiSelectionPanelDetail, [105](#)
  - shallot::PanelDetails::PanelDetail, [118](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail, [144](#)
- INDEX\_NORMAL
  - shallot::FilePropertyDialog::Tab, [152](#)
  - shallot::FileSearch::SearchCriterionFactory, [138](#)
  - shallot::ThumbnailProvider, [157](#)
- INDEX\_VERYIMPORTANT
  - shallot::FilePropertyDialog::Tab, [152](#)
- INDEX\_VERYINTERESTING
  - shallot::PanelDetails::MultiSelectionPanelDetail, [106](#)
  - shallot::PanelDetails::PanelDetail, [118](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail, [144](#)
- IconTextBanner
  - shallot::FilePropertyDialog::Tab::PropertyType, [129](#)
- id
  - shallot::Bookmarks::Bookmark, [28](#)
- increment\_transferred\_bytes
  - shallot::Operations::HandlerTransfer, [93](#)
- Indirect
  - shallot::Operations::FilesystemOperationStep::↔ ConflictResolution, [36](#)
- initialize
  - shallot::Actions::AbstractAction, [19](#)
  - shallot::Actions::ActionAction, [22](#)
  - shallot::Actions::SubmenuAction, [147](#)
- initialize\_sync
  - shallot::Actions::ActionAction, [22](#)
- is\_debug\_build
  - shallot::Environment, [44](#)
- is\_prefix\_of
  - shallot::Eurl, [48](#)
- is\_visible\_for
  - shallot::FileSearch::SearchCriterionFactory, [137](#)
- isExceptionClass
  - shallot::Exceptions::ArgumentException, [26](#)
  - shallot::Exceptions::IOException, [97](#)
  - shallot::Exceptions::ProgramException, [126](#)
  - shallot::Exceptions::RuntimeException, [131](#)
  - shallot::Exceptions::ScriptedException, [133](#)
- itemlist
  - shallot::Filesystem::Handler, [89](#)
  - shallot::Operations::FilesystemOperation, [69](#)
- itemlist\_by\_type
  - shallot::Operations::FilesystemOperation, [70](#)
- jump\_to\_eurl
  - shallot::MainWindow, [100](#)
- LASTPHYSICALTYPE
  - shallot::Filesystem::NodeType, [110](#)
- Link
  - shallot::Filesystem::NodeType, [110](#)
- link\_triggered
  - shallot::PanelDetails::MultiSelectionPanelDetail, [104](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail, [142](#)
- list\_extendedattributes
  - shallot::Filesystem::Handler, [90](#)
  - shallot::Operations::FilesystemOperation, [70](#)
- log\_debug
  - shallot::Logging, [99](#)
- log\_error
  - shallot::Logging, [99](#)
- log\_info
  - shallot::Logging, [99](#)
- log\_warning
  - shallot::Logging, [99](#)
- map
  - shallot::IntlStringMap, [96](#)
- match
  - shallot::FileSearch::SearchCriterion, [134](#)
- MergeDirectories
  - shallot::Operations::FilesystemOperationStep::↔ ConflictResolution, [37](#)
- messagebox
  - shallot::Actions::ExecutionUserFeedback, [55](#)
- move\_bookmark\_down
  - shallot::Bookmarks, [30](#)
- move\_bookmark\_to\_folder
  - shallot::Bookmarks, [32](#)
- move\_bookmark\_up
  - shallot::Bookmarks, [32](#)
- move\_item
  - shallot::Operations::FilesystemOperation, [70](#)
- move\_items
  - shallot::Operations::FilesystemOperation, [71](#)
- NONE
  - shallot::Filesystem::NodeType, [110](#)
- No
  - shallot::Actions::ExecutionUserFeedback::↔ MessageBoxButton, [102](#)
- OK
  - shallot::Actions::ExecutionUserFeedback::↔ MessageBoxButton, [102](#)

- open\_items
  - shallot::MainWindow, [101](#)
- OpenFile
  - shallot::Actions::DefaultPrecedenceValues, [38](#)
- outer\_url
  - shallot::Eurl, [49](#)
- outer\_url\_is\_root\_directory
  - shallot::Eurl, [49](#)
- outermost\_inner\_eurl
  - shallot::Eurl, [49](#)
- OverwriteDestination
  - shallot::Operations::FilesystemOperationStep::↔  
ConflictResolution, [37](#)
- parent\_segment
  - shallot::Eurl, [49](#)
- path
  - shallot::Eurl, [49](#)
- refresh
  - shallot::Filesystem, [59](#)
- register
  - shallot::Actions, [25](#)
  - shallot::FilePropertyDialog::Tab, [150](#)
  - shallot::FileSearch::SearchCriterionFactory, [137](#)
  - shallot::Filesystem::Handler, [90](#)
  - shallot::PanelDetails::MultiSelectionPanelDetail,  
[104](#)
  - shallot::PanelDetails::PanelDetail, [117](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail,  
[143](#)
  - shallot::Setting, [140](#)
  - shallot::ThumbnailProvider, [157](#)
- register\_as\_transferrable
  - shallot::DetailColumn, [42](#)
- remove\_bookmark
  - shallot::Bookmarks, [32](#)
- remove\_extendedattribute
  - shallot::Filesystem::Handler, [90](#)
  - shallot::Operations::FilesystemOperation, [71](#)
- rename\_item
  - shallot::Filesystem::Handler, [91](#)
- RenameDestinationBefore
  - shallot::Operations::FilesystemOperationStep::↔  
ConflictResolution, [37](#)
- resolve\_conflicts
  - shallot::Operations::FilesystemOperationProgress↔  
Monitor, [74](#)
- resolve\_eurl\_link
  - shallot::Operations::FilesystemOperation, [71](#)
- resolve\_eurl\_link\_nonrecursive
  - shallot::Operations::FilesystemOperation, [72](#)
- resolve\_node\_link
  - shallot::Operations::FilesystemOperation, [72](#)
- resolve\_node\_link\_nonrecursive
  - shallot::Operations::FilesystemOperation, [72](#)
- respect\_cancel
  - shallot::Actions::ExecutionInfo, [53](#)
- Retry
  - shallot::Actions::ExecutionUserFeedback::↔  
MessageBoxButton, [102](#)
- root
  - shallot::Eurl, [50](#)
- rootnode
  - shallot::Filesystem, [61](#)
- run
  - shallot::Environment::Thread, [155](#)
  - shallot::Environment::Timer, [158](#)
- scheme
  - shallot::Eurl, [50](#)
- selected\_index\_for\_property
  - shallot::FilePropertyDialog::Tab, [150](#)
- set\_custom\_data
  - shallot::Operations::Operation, [115](#)
- set\_customattribute
  - shallot::Filesystem::Handler, [91](#)
- set\_details
  - shallot::Actions::ExecutionInfo, [53](#)
- set\_enabled
  - shallot::Actions::AbstractAction, [19](#)
  - shallot::Actions::ActionAction, [23](#)
  - shallot::Actions::SubmenuAction, [147](#)
- set\_extendedattribute
  - shallot::Filesystem::Handler, [92](#)
  - shallot::Operations::FilesystemOperation, [72](#)
- set\_items
  - shallot::Filesystem::NodeList, [108](#)
- set\_mtime
  - shallot::Filesystem::Handler, [92](#)
- set\_progress
  - shallot::Actions::ExecutionInfo, [54](#)
- set\_progress\_indeterminate
  - shallot::Actions::ExecutionInfo, [54](#)
- set\_row
  - shallot::PanelDetails::PanelDetailInst, [119](#)
- set\_subitems
  - shallot::Actions::SubmenuAction, [148](#)
- set\_value
  - shallot::ConfigurationValue, [35](#)
  - shallot::PanelDetails::MultiSelectionPanelDetail,  
[105](#)
  - shallot::PanelDetails::SingleSelectionPanelDetail,  
[143](#)
  - shallot::Setting, [140](#)
- set\_visible
  - shallot::Actions::AbstractAction, [20](#)
  - shallot::Actions::ActionAction, [23](#)
  - shallot::Actions::SubmenuAction, [148](#)
- shallot, [16](#)
- shallot.\_ApiProxy, [17](#)
- shallot.Actions, [23](#)
- shallot.Actions.AbstractAction, [18](#)
- shallot.Actions.ActionAction, [21](#)
- shallot.Actions.ByRegExpPredicate, [33](#)
- shallot.Actions.DefaultPrecedenceValues, [38](#)
- shallot.Actions.DontResolveLinksPredicate, [43](#)
- shallot.Actions.ExecutionInfo, [51](#)

- shallot.Actions.ExecutionUserFeedback, [54](#)
- shallot.Actions.ExecutionUserFeedback.MessageBox↔  
Button, [101](#)
- shallot.Actions.HideOnCurrentDirectoryLevelPredicate, [94](#)
- shallot.Actions.HideOnSelectionLevelPredicate, [94](#)
- shallot.Actions.KeyShortcutPredicate, [98](#)
- shallot.Actions.OnDirectoriesPredicate, [111](#)
- shallot.Actions.OnFilesPredicate, [111](#)
- shallot.Actions.OnLinksPredicate, [112](#)
- shallot.Actions.OnSingleEntrySelectionPredicate, [112](#)
- shallot.Actions.PositionIndexPredicate, [124](#)
- shallot.Actions.Predicate, [125](#)
- shallot.Actions.SubmenuItemAction, [146](#)
- shallot.Bookmarks, [29](#)
- shallot.Bookmarks.Bookmark, [27](#)
- shallot.ConfigurationValue, [34](#)
- shallot.ConfigurationValue.Category, [33](#)
- shallot.DetailColumn, [39](#)
- shallot.Environment, [43](#)
- shallot.Environment.Thread, [153](#)
- shallot.Environment.Timer, [158](#)
- shallot.Eurl, [45](#)
- shallot.Exceptions, [51](#)
- shallot.Exceptions.ArgumentException, [25](#)
- shallot.Exceptions.IOException, [97](#)
- shallot.Exceptions.ProgramException, [125](#)
- shallot.Exceptions.RuntimeException, [130](#)
- shallot.Exceptions.ScriptedException, [131](#)
- shallot.FilePropertyDialog, [56](#)
- shallot.FilePropertyDialog.Tab, [148](#)
- shallot.FilePropertyDialog.Tab.PropertyButtonConfig, [126](#)
- shallot.FilePropertyDialog.Tab.PropertyConfig, [127](#)
- shallot.FilePropertyDialog.Tab.PropertyType, [128](#)
- shallot.FilePropertyDialog.TabPropertyIconTextBanner, [152](#)
- shallot.FileSearch, [57](#)
- shallot.FileSearch.SearchCriterion, [133](#)
- shallot.FileSearch.SearchCriterionFactory, [135](#)
- shallot.Filesystem, [57](#)
- shallot.Filesystem.Handler, [78](#)
- shallot.Filesystem.Node, [106](#)
- shallot.Filesystem.NodeList, [107](#)
- shallot.Filesystem.NodeType, [109](#)
- shallot.IntlStringMap, [95](#)
- shallot.IntlStringMap.IntlString, [95](#)
- shallot.Logging, [98](#)
- shallot.MainWindow, [100](#)
- shallot.Operations, [116](#)
- shallot.Operations.FilesystemOperation, [61](#)
- shallot.Operations.FilesystemOperationProgress↔  
Monitor, [73](#)
- shallot.Operations.FilesystemOperationStep, [75](#)
- shallot.Operations.FilesystemOperationStep.Conflict↔  
Resolution, [36](#)
- shallot.Operations.HandlerTransfer, [93](#)
- shallot.Operations.Operation, [113](#)
- shallot.PanelDetails, [119](#)
- shallot.PanelDetails.AbstractPanelDetailValueElement, [20](#)
- shallot.PanelDetails.MultiSelectionPanelDetail, [103](#)
- shallot.PanelDetails.PanelDetail, [117](#)
- shallot.PanelDetails.PanelDetailInst, [118](#)
- shallot.PanelDetails.PanelDetailValueElementButton, [120](#)
- shallot.PanelDetails.PanelDetailValueElementIcon, [121](#)
- shallot.PanelDetails.PanelDetailValueElementString, [122](#)
- shallot.PanelDetails.PanelDetailValueElementWaiting, [123](#)
- shallot.PanelDetails.SingleSelectionPanelDetail, [141](#)
- shallot.Setting, [138](#)
- shallot.Setting.GroupInfo, [76](#)
- shallot.Streaming, [144](#)
- shallot.Streaming.BlobReadDataDevice, [26](#)
- shallot.Streaming.ReadDataDevice, [129](#)
- shallot.Streaming.StreamReadDataDevice, [145](#)
- shallot.ThumbnailProvider, [155](#)
- shallot::Actions
  - register, [25](#)
- shallot::Actions::AbstractAction
  - initialize, [19](#)
  - set\_enabled, [19](#)
  - set\_visible, [20](#)
- shallot::Actions::ActionAction
  - \_\_init\_\_, [21](#)
  - action, [22](#)
  - initialize, [22](#)
  - initialize\_sync, [22](#)
  - set\_enabled, [23](#)
  - set\_visible, [23](#)
- shallot::Actions::DefaultPrecedenceValues
  - BuiltinSpecialOpen, [38](#)
  - OpenFile, [38](#)
- shallot::Actions::ExecutionInfo
  - add\_changed\_eurl, [53](#)
  - respect\_cancel, [53](#)
  - set\_details, [53](#)
  - set\_progress, [54](#)
  - set\_progress\_indeterminate, [54](#)
- shallot::Actions::ExecutionUserFeedback
  - messagebox, [55](#)
  - simple\_inputbox, [55](#)
  - simple\_messagebox, [56](#)
- shallot::Actions::ExecutionUserFeedback::Message↔  
BoxButton
  - Cancel, [102](#)
  - Continue, [102](#)
  - No, [102](#)
  - OK, [102](#)
  - Retry, [102](#)
  - Yes, [102](#)
- shallot::Actions::SubmenuItemAction
  - \_\_init\_\_, [147](#)
  - initialize, [147](#)

- set\_enabled, [147](#)
  - set\_subitems, [148](#)
  - set\_visible, [148](#)
- shallot::Bookmarks
  - add\_bookmark, [29](#)
  - change\_bookmark, [30](#)
  - change\_bookmark\_tags, [30](#)
  - move\_bookmark\_down, [30](#)
  - move\_bookmark\_to\_folder, [32](#)
  - move\_bookmark\_up, [32](#)
  - remove\_bookmark, [32](#)
- shallot::Bookmarks::Bookmark
  - id, [28](#)
  - tags, [28](#)
- shallot::ConfigurationValue
  - \_\_init\_\_, [35](#)
  - set\_value, [35](#)
- shallot::ConfigurationValue::Category
  - Behavior, [34](#)
  - ExternalTools, [34](#)
  - GUI, [34](#)
- shallot::DetailColumn
  - \_\_init\_\_, [40](#)
  - apply\_value, [40](#)
  - compute\_value, [41](#)
  - determine\_value, [41](#)
  - find\_by\_name, [41](#)
  - INDEX\_CORE, [42](#)
  - INDEX\_EXOTIC, [42](#)
  - INDEX\_INTERESTING, [42](#)
  - register\_as\_transferrable, [42](#)
- shallot::Environment
  - is\_debug\_build, [44](#)
  - shallot\_builtinplugin\_dir, [44](#)
  - shallot\_data\_dir, [44](#)
  - shallot\_language, [44](#)
  - shallot\_systemplugin\_dir, [45](#)
  - shallot\_userplugin\_dir, [45](#)
- shallot::Environment::Thread
  - execute\_threaded, [154](#)
  - run, [155](#)
  - start, [155](#)
- shallot::Environment::Timer
  - run, [158](#)
  - start, [159](#)
  - start\_in\_timer, [159](#)
- shallot::Eurl
  - as\_string, [46](#)
  - basename, [46](#)
  - create, [47](#)
  - enwrap\_with\_outer\_url, [47](#)
  - from\_string, [47](#)
  - has\_inner\_urls, [48](#)
  - has\_parent\_segment, [48](#)
  - hostname, [48](#)
  - is\_prefix\_of, [48](#)
  - outer\_url, [49](#)
  - outer\_url\_is\_root\_directory, [49](#)
  - outermost\_inner\_eurl, [49](#)
  - parent\_segment, [49](#)
  - path, [49](#)
  - root, [50](#)
  - scheme, [50](#)
  - with\_appended\_segment, [50](#)
  - with\_appended\_segments, [50](#)
- shallot::Exceptions::ArgumentException
  - isExceptionClass, [26](#)
- shallot::Exceptions::IOException
  - isExceptionClass, [97](#)
- shallot::Exceptions::ProgramException
  - isExceptionClass, [126](#)
- shallot::Exceptions::RuntimeException
  - isExceptionClass, [131](#)
- shallot::Exceptions::ScriptedException
  - \_\_init\_\_, [132](#)
  - isExceptionClass, [133](#)
- shallot::FilePropertyDialog::Tab
  - \_\_init\_\_, [150](#)
  - INDEX\_CORE, [151](#)
  - INDEX\_EXOTIC, [151](#)
  - INDEX\_IMPORTANT, [151](#)
  - INDEX\_NORMAL, [152](#)
  - INDEX\_VERYIMPORTANT, [152](#)
  - register, [150](#)
  - selected\_index\_for\_property, [150](#)
  - update\_widget, [151](#)
- shallot::FilePropertyDialog::Tab::PropertyButtonConfig
  - \_\_init\_\_, [127](#)
- shallot::FilePropertyDialog::Tab::PropertyConfig
  - \_\_init\_\_, [128](#)
- shallot::FilePropertyDialog::Tab::PropertyType
  - IconTextBanner, [129](#)
  - String, [129](#)
  - StringMap, [129](#)
- shallot::FilePropertyDialog::TabPropertyIconTextBanner
  - add\_icon, [153](#)
  - add\_text, [153](#)
- shallot::FileSearch::SearchCriterion
  - \_\_init\_\_, [134](#)
  - configure, [134](#)
  - match, [134](#)
- shallot::FileSearch::SearchCriterionFactory
  - \_\_init\_\_, [136](#)
  - get\_searchspec\_description, [136](#)
  - INDEX\_CORE, [137](#)
  - INDEX\_EXOTIC, [138](#)
  - INDEX\_NORMAL, [138](#)
  - is\_visible\_for, [137](#)
  - register, [137](#)
- shallot::Filesystem
  - create\_node, [58](#)
  - find\_nodes\_for\_eurl, [59](#)
  - get\_or\_create\_node, [59](#)
  - refresh, [59](#)
  - rootnode, [61](#)
  - try\_get\_nodes\_for\_eurl, [61](#)



- shallot::Filesystem::Handler
  - can\_create\_directory, 79
  - can\_create\_file, 80
  - can\_create\_link, 80
  - can\_delete\_item, 81
  - can\_get\_filecontent, 81
  - can\_rename\_item, 81
  - configure\_item, 82
  - create\_directory, 82
  - create\_file, 84
  - create\_link, 84
  - delete\_item, 85
  - get\_actions, 85
  - get\_customattributes, 85
  - get\_extendedattribute, 86
  - get\_extendedattribute\_size, 86
  - get\_file\_content, 87
  - get\_linktarget, 87
  - get\_mimetype, 87
  - get\_mtime, 88
  - get\_size, 88
  - get\_type, 89
  - itemlist, 89
  - list\_extendedattributes, 90
  - register, 90
  - remove\_extendedattribute, 90
  - rename\_item, 91
  - set\_customattribute, 91
  - set\_extendedattribute, 92
  - set\_mtime, 92
- shallot::Filesystem::Node
  - add\_detail, 107
- shallot::Filesystem::NodeList
  - add\_item, 108
  - begin\_iterative\_adding, 108
  - end\_iterative\_adding, 108
  - set\_items, 108
- shallot::Filesystem::NodeType
  - Directory, 109
  - FIRSTTYPE, 110
  - File, 109
  - LASTPHYSICALTYPE, 110
  - Link, 110
  - NONE, 110
  - Unknown, 110
- shallot::IntlStringMap
  - \_\_init\_\_, 96
  - map, 96
- shallot::IntlStringMap::IntlString
  - \_\_init\_\_, 95
- shallot::Logging
  - log\_debug, 99
  - log\_error, 99
  - log\_info, 99
  - log\_warning, 99
- shallot::MainWindow
  - jump\_to\_eurl, 100
  - open\_items, 101
- shallot::Operations::FilesystemOperation
  - can\_copy\_item, 63
  - can\_create\_directory, 63
  - can\_create\_file, 64
  - can\_create\_link, 64
  - can\_delete\_item, 64
  - can\_get\_filecontent, 65
  - can\_move\_item, 65
  - copy\_item, 65
  - copy\_items, 66
  - create\_directory, 66
  - create\_file, 66
  - create\_link, 67
  - delete\_directory\_if\_empty, 67
  - delete\_item, 67
  - get\_extendedattribute, 68
  - get\_extendedattribute\_size, 68
  - get\_file\_content, 68
  - get\_linktarget, 69
  - get\_type, 69
  - itemlist, 69
  - itemlist\_by\_type, 70
  - list\_extendedattributes, 70
  - move\_item, 70
  - move\_items, 71
  - remove\_extendedattribute, 71
  - resolve\_eurl\_link, 71
  - resolve\_eurl\_link\_nonrecursive, 72
  - resolve\_node\_link, 72
  - resolve\_node\_link\_nonrecursive, 72
  - set\_extendedattribute, 72
- shallot::Operations::FilesystemOperationProgress↔
  - Monitor
    - \_\_init\_\_, 74
    - changed, 74
    - resolve\_conflicts, 74
- shallot::Operations::FilesystemOperationStep::↔
  - ConflictResolution
    - Indirect, 36
    - MergeDirectories, 37
    - OverwriteDestination, 37
    - RenameDestinationBefore, 37
    - Skip, 37
    - Unresolved, 37
    - UseDifferentDestinationName, 37
- shallot::Operations::HandlerTransfer
  - increment\_transferred\_bytes, 93
- shallot::Operations::Operation
  - fetch\_container\_file, 114
  - fetch\_file, 114
  - get\_custom\_data, 114
  - get\_detail\_from\_cache, 115
  - set\_custom\_data, 115
  - store\_detail\_in\_cache, 115
- shallot::PanelDetails::MultiSelectionPanelDetail
  - \_\_init\_\_, 104
  - INDEX\_EXOTIC, 105
  - INDEX\_INTERESTING, 105

- INDEX\_VERYINTERESTING, [106](#)
- link\_triggered, [104](#)
- register, [104](#)
- set\_value, [105](#)
- shallot::PanelDetails::PanelDetail
  - INDEX\_EXOTIC, [118](#)
  - INDEX\_INTERESTING, [118](#)
  - INDEX\_VERYINTERESTING, [118](#)
  - register, [117](#)
- shallot::PanelDetails::PanelDetailInst
  - set\_row, [119](#)
- shallot::PanelDetails::PanelDetailValueElementButton
  - \_\_init\_\_, [121](#)
- shallot::PanelDetails::PanelDetailValueElementIcon
  - \_\_init\_\_, [122](#)
- shallot::PanelDetails::PanelDetailValueElementString
  - \_\_init\_\_, [123](#)
- shallot::PanelDetails::SingleSelectionPanelDetail
  - \_\_init\_\_, [142](#)
  - INDEX\_EXOTIC, [144](#)
  - INDEX\_INTERESTING, [144](#)
  - INDEX\_VERYINTERESTING, [144](#)
  - link\_triggered, [142](#)
  - register, [143](#)
  - set\_value, [143](#)
- shallot::Setting
  - \_\_init\_\_, [139](#)
  - get\_value, [140](#)
  - register, [140](#)
  - set\_value, [140](#)
  - value\_description, [141](#)
- shallot::Setting::GroupInfo
  - Behavior, [77](#)
  - DataView, [77](#)
  - FileHandling, [77](#)
  - GUI, [77](#)
  - Global, [77](#)
  - Special, [77](#)
- shallot::Streaming::BlobReadDataDevice
  - \_\_init\_\_, [27](#)
- shallot::Streaming::ReadDataDevice
  - getdata, [130](#)
- shallot::Streaming::StreamReadDataDevice
  - \_\_init\_\_, [146](#)
- shallot::ThumbnailProvider
  - get\_thumbnail, [156](#)
  - INDEX\_CORE, [157](#)
  - INDEX\_EXOTIC, [157](#)
  - INDEX\_FALLBACK, [157](#)
  - INDEX\_NORMAL, [157](#)
  - register, [157](#)
- shallot\_builtplugin\_dir
  - shallot::Environment, [44](#)
- shallot\_data\_dir
  - shallot::Environment, [44](#)
- shallot\_language
  - shallot::Environment, [44](#)
- shallot\_systemplugin\_dir
  - shallot::Environment, [45](#)
- shallot\_userplugin\_dir
  - shallot::Environment, [45](#)
- simple\_inputbox
  - shallot::Actions::ExecutionUserFeedback, [55](#)
- simple\_messagebox
  - shallot::Actions::ExecutionUserFeedback, [56](#)
- Skip
  - shallot::Operations::FilesystemOperationStep::↵  
ConflictResolution, [37](#)
- Special
  - shallot::Setting::GroupInfo, [77](#)
- start
  - shallot::Environment::Thread, [155](#)
  - shallot::Environment::Timer, [159](#)
- start\_in\_timer
  - shallot::Environment::Timer, [159](#)
- store\_detail\_in\_cache
  - shallot::Operations::Operation, [115](#)
- String
  - shallot::FilePropertyDialog::Tab::PropertyType,  
[129](#)
- StringMap
  - shallot::FilePropertyDialog::Tab::PropertyType,  
[129](#)
- tags
  - shallot::Bookmarks::Bookmark, [28](#)
- try\_get\_nodes\_for\_eurl
  - shallot::Filesystem, [61](#)
- Unknown
  - shallot::Filesystem::NodeType, [110](#)
- Unresolved
  - shallot::Operations::FilesystemOperationStep::↵  
ConflictResolution, [37](#)
- update\_widget
  - shallot::FilePropertyDialog::Tab, [151](#)
- UseDifferentDestinationName
  - shallot::Operations::FilesystemOperationStep::↵  
ConflictResolution, [37](#)
- value\_description
  - shallot::Setting, [141](#)
- with\_appended\_segment
  - shallot::Eurl, [50](#)
- with\_appended\_segments
  - shallot::Eurl, [50](#)
- Yes
  - shallot::Actions::ExecutionUserFeedback::↵  
MessageBoxButton, [102](#)