

CONTENTS

- FOREWORD 6
- INTRODUCTION 7
- 1 Scope 8
- 2 Normative references 8
- 3 Terms, definitions and abbreviations 8
 - 3.1 Terms and definitions 8
 - 3.2 Abbreviations 10
- 4 Document conventions 11
- 5 Background technical information 12
 - 5.1 Overview 12
 - 5.2 Target port groups 12
 - 5.3 Relationship between target port groups in SCSI and in this API 13
 - 5.3.1 General 13
 - 5.3.2 Symmetric and asymmetric multipath access 14
 - 5.3.3 Logical unit affinity groups 15
 - 5.3.4 Load balancing 15
 - 5.3.5 Model overview 16
 - 5.4 Client discovery of optional behavior 18
 - 5.4.1 General 18
 - 5.4.2 Discovery of load balancing behavior 18
 - 5.4.3 Client discovery of failover/failback capabilities 20
 - 5.4.4 Client discovery of a driver's OS device file name behavior 20
 - 5.4.5 Client discovery of auto-failback capabilities 20
 - 5.4.6 Client discovery of auto-probing capabilities 21
 - 5.4.7 Client discovery of support for LU assignment to target port groups 21
 - 5.5 Events 21
 - 5.6 API concepts 22
 - 5.6.1 Library and plugins 22
 - 5.6.2 OS-independent implementation 22
 - 5.6.3 Object ID 22
 - 5.6.4 Object ID list 23
- 6 Constants and structures 23
 - 6.1 MP_WCHAR 23
 - 6.2 MP_CHAR 23
 - 6.3 MP_BYTE 23
 - 6.4 MP_BOOL 24
 - 6.5 MP_XBOOL 24
 - 6.6 MP_UINT32 24
 - 6.7 MP_UINT64 24
 - 6.8 MP_STATUS 24
 - 6.9 MP_PATH_STATE 25
 - 6.10 MP_OBJECT_VISIBILITY_FN 26
 - 6.11 MP_OBJECT_PROPERTY_FN 26
 - 6.12 MP_OBJECT_TYPE 27
 - 6.13 MP_OID 27
 - 6.14 MP_OID_LIST 28

6.15	MP_PORT_TRANSPORT_TYPE	28
6.16	MP_ACCESS_STATE_TYPE.....	29
6.17	MP_LOAD_BALANCE_TYPE.....	29
6.18	MP_PROPRIETARY_PROPERTY	30
6.19	MP_PROPRIETARY_LOAD_BALANCE_PROPERTIES	31
6.20	MP_LOGICAL_UNIT_NAME_TYPE	31
6.21	MP_LIBRARY_PROPERTIES.....	32
6.22	MP_AUTOFAILBACK_SUPPORT	32
6.23	MP_AUTOPROBING_SUPPORT	33
6.24	MP_PLUGIN_PROPERTIES.....	33
6.25	MP_DEVICE_PRODUCT_PROPERTIES	35
6.26	MP_MULTIPATH_LOGICAL_UNIT_PROPERTIES.....	36
6.27	MP_PATH_LOGICAL_UNIT_PROPERTIES.....	38
6.28	MP_INITIATOR_PORT_PROPERTIES	39
6.29	MP_TARGET_PORT_PROPERTIES	39
6.30	MP_TARGET_PORT_GROUP_PROPERTIES	40
6.31	MP_TPG_STATE_PAIR.....	40
7	APIs.....	41
7.1	API overview	41
7.2	MP_AssignLogicalUnitToTPG.....	42
7.3	MP_CancelOverridePath	43
7.4	MP_CompareOIDs	44
7.5	MP_DeregisterForObjectPropertyChanges	44
7.6	MP_DeregisterForObjectVisibilityChanges	45
7.7	MP_DeregisterPlugin.....	46
7.8	MP_DisableAutoFailback.....	47
7.9	MP_DisableAutoProbing.....	47
7.10	MP_DisablePath.....	48
7.11	MP_EnableAutoFailback.....	49
7.12	MP_EnableAutoProbing.....	49
7.13	MP_EnablePath	50
7.14	MP_FreeOidList	51
7.15	MP_GetAssociatedPathOidList.....	51
7.16	MP_GetAssociatedPluginOid.....	52
7.17	MP_GetAssociatedTPGOidList.....	52
7.18	MP_GetDeviceProductOidList	53
7.19	MP_GetDeviceProductProperties	54
7.20	MP_GetInitiatorPortOidList.....	55
7.21	MP_GetInitiatorPortProperties	56
7.22	MP_GetLibraryProperties	56
7.23	MP_GetMPLuOidListFromTPG	57
7.24	MP_GetMPLLogicalUnitProperties.....	57
7.25	MP_GetMultipathLus	58
7.26	MP_GetObjectType	59
7.27	MP_GetPathLogicalUnitProperties.....	60
7.28	MP_GetPluginOidList	60
7.29	MP_GetPluginProperties	61
7.30	MP_GetProprietaryLoadBalanceOidList.....	62
7.31	MP_GetProprietaryLoadBalanceProperties.....	62

7.32	MP_GetTargetPortGroupProperties	63
7.33	MP_GetTargetPortOidList.....	64
7.34	MP_GetTargetPortProperties	64
7.35	MP_RegisterForObjectPropertyChanges	65
7.36	MP_RegisterForObjectVisibilityChanges.....	66
7.37	MP_RegisterPlugin.....	67
7.38	MP_SetLogicalUnitLoadBalanceType	68
7.39	MP_SetOverridePath.....	69
7.40	MP_SetPathWeight	70
7.41	MP_SetPluginLoadBalanceType.....	70
7.42	MP_SetFailbackPollingRate	71
7.43	MP_SetProbingPollingRate	72
7.44	MP_SetProprietaryProperties	72
7.45	MP_SetTPGAccess	73
8	Implementation compliance	74
9	Implementations	75
9.1	Backwards compatibility	75
9.2	Client usage notes	75
9.2.1	Reserved fields.....	75
9.2.2	Event notification within a single client	75
9.2.3	Event notification and multi-threading	75
9.3	Library implementation notes.....	75
9.3.1	Multi-threading support.....	75
9.3.2	Event notification and multi-threading	75
9.3.3	Structure packing	75
9.3.4	Calling conventions	76
9.4	Plugin implementation notes.....	76
9.4.1	Reserved fields.....	76
9.4.2	Multi-threading support.....	76
9.4.3	Event notification to different clients	76
9.4.4	Event notification and multi-threading	76
9.4.5	Event overhead conservation.....	76
9.4.6	Function names.....	76
Annex A (informative)	Device names	77
A.1	General	77
A.2	Initiator port osDeviceName	77
A.3	Logical unit osDeviceName	77
Annex B (informative)	Synthesizing target port groups.....	79
Annex C (informative)	Transport layer multipathing.....	80
Annex D (informative)	Coding examples	81
D.1	General	81
D.2	Example of getting library properties	81
D.3	Example of getting plugin properties.....	81
D.4	Example of discovering path LUs associated with an MP LU	82
Annex E (informative)	Library/plugin API.....	84
Bibliography	85

Figure 1 – Asymmetric array example	13
Figure 2 – API Instances corresponding to asymmetric array example	14
Figure 3 – Relationship between various objects in the multipath model	17
Figure 4 – Driver representation of a logical unit with multiple paths	18
Figure 5 – APIs relative to the objects from Figure 1	42
Figure B.1 – Synthetic target port groups	79
Table A.1 – Names for the osDeviceName field	77
Table A.2 – Names for the osDeviceName	78