



mmi:5.6

API Reference

X Platform, X5 HEVC SDI 1.0.2

23/04/2026

TABLE OF CONTENTS

1	Overview	14
2	alarms (1.5)	16
2.1	Overview	16
2.2	Command Reference	16
2.2.1	GetActiveAlarms	16
2.2.2	GetAlarmsHistory	16
2.2.3	ClearAlarmsHistory	16
2.2.4	GetRegisteredAlarms	16
2.2.5	GetAlarmOverrides	16
2.2.6	GetAllAlarmOverrides	16
2.2.7	SetAlarmOverrides	16
2.2.8	DeleteAlarmOverrides	17
2.3	Type Reference	17
2.3.1	Alarm	17
2.3.2	AlarmGroupId	17
2.3.3	AlarmOverrideConfig	18
2.3.4	AlarmOverrideConfigInfo	18
2.3.5	AlarmOverrideConfigParameters	18
2.3.6	AlarmOverrideDomain	18
2.3.7	AlarmOverrideSeverity	18
2.3.8	AlarmsQuery	19
2.3.9	AlarmsSince	19
2.3.10	CardDomain	19
2.3.11	ClearAlarmsHistory.Request	19
2.3.12	ClearAlarmsHistory.Response	19
2.3.13	ConfigLogin	19
2.3.14	ConfigObject	19
2.3.15	ConfigObjectDomain	20
2.3.16	DeleteAlarmOverrideRequest	20
2.3.17	DeleteAlarmOverrideResponse	20
2.3.18	DeleteAlarmOverrideResult	20
2.3.19	DeleteAlarmOverrides.Request	20
2.3.20	DeleteAlarmOverrides.Response	20
2.3.21	GetActiveAlarms.Request	21
2.3.22	GetActiveAlarms.Response	21
2.3.23	GetAlarmOverrides.Request	21
2.3.24	GetAlarmOverrides.Response	21
2.3.25	GetAlarmsHistory.Request	21
2.3.26	GetAlarmsHistory.Response	21
2.3.27	GetAllAlarmOverrides.Request	21
2.3.28	GetAllAlarmOverrides.Response	21
2.3.29	GetRegisteredAlarms.Request	21
2.3.30	GetRegisteredAlarms.Response	22
2.3.31	Hysteresis	22
2.3.32	RegisteredAlarm	22
2.3.33	RegisteredAlarmOverrides	22
2.3.34	RegisteredAlarms	22
2.3.35	SetAlarmOverride	22
2.3.36	SetAlarmOverrideRequest	22
2.3.37	SetAlarmOverrideResponse	23
2.3.38	SetAlarmOverrideResult	23
2.3.39	SetAlarmOverrides.Request	23
2.3.40	SetAlarmOverrides.Response	23
2.3.41	Severity	23
2.3.42	SystemDomain	23
3	authentication (1.6)	24
3.1	Command Reference	24
3.1.1	BeginSession	24

3.1.2	RefreshSession	24
3.1.3	EndSession	24
3.1.4	GetSessions	24
3.1.5	GetTokenInfo	24
3.1.6	GetUsers	24
3.1.7	CreateUser	24
3.1.8	DeleteUser	25
3.1.9	SetPassword	25
3.1.10	SetGroupMembership	25
3.1.11	SetUserDetails	25
3.1.12	SetUserBlock	25
3.1.13	SetUserPasswordChange	25
3.1.14	GetUserGroups	25
3.1.15	PatchUserGroups	25
3.1.16	SetLDAPSettings	26
3.1.17	GetLDAPSettings	26
3.1.18	GetLDAPGroupMappings	26
3.1.19	SetLDAPGroupMappings	26
3.1.20	TestLdapLogin	26
3.2	Type Reference	26
3.2.1	BeginSession.Request	26
3.2.2	BeginSession.Response	26
3.2.3	CreateUser.Request	26
3.2.4	CreateUser.Response	27
3.2.5	CreateUserRequest	27
3.2.6	Credentials	27
3.2.7	DeleteUser.Request	27
3.2.8	DeleteUser.Response	27
3.2.9	EndSession.Request	27
3.2.10	EndSession.Response	27
3.2.11	GetLDAPGroupMappings.Request	28
3.2.12	GetLDAPGroupMappings.Response	28
3.2.13	GetLDAPSettings.Request	28
3.2.14	GetLDAPSettings.Response	28
3.2.15	GetSessions.Request	28
3.2.16	GetSessions.Response	28
3.2.17	GetSessionsRequest	28
3.2.18	GetSessionsResponse	28
3.2.19	GetTokenInfo.Request	29
3.2.20	GetTokenInfo.Response	29
3.2.21	GetUserGroups.Request	29
3.2.22	GetUserGroups.Response	29
3.2.23	GetUsers.Request	29
3.2.24	GetUsers.Response	29
3.2.25	GroupConfig	29
3.2.26	GroupInfo	29
3.2.27	GroupMembership	30
3.2.28	LDAPBindMethod	30
3.2.29	LDAPGroupMapping	30
3.2.30	LDAPLogin	30
3.2.31	LDAPSettings	30
3.2.32	LDAPUserSearchParams	31
3.2.33	LdapGroupMappingsSetter	31
3.2.34	LdapLoginStepRecord	31
3.2.35	LdapLoginStepResult	32
3.2.36	LocalLogin	32
3.2.37	LoginMethod	32
3.2.38	LoginMethod.auto	32
3.2.39	PasswordExpiration	32
3.2.40	PasswordExpiration.neverExpires	32
3.2.41	PatchUserGroups.Request	32
3.2.42	PatchUserGroups.Response	33

3.2.43	RefreshSession.Request	33
3.2.44	RefreshSession.Response	33
3.2.45	SessionEnd	33
3.2.46	SessionEndReason	33
3.2.47	SessionInfo	33
3.2.48	SessionState	33
3.2.49	SessionState.active	34
3.2.50	SessionStateFilter	34
3.2.51	SetGroupMembership.Request	34
3.2.52	SetGroupMembership.Response	34
3.2.53	SetLDAPGroupMappings.Request	34
3.2.54	SetLDAPGroupMappings.Response	34
3.2.55	SetLDAPSettings.Request	34
3.2.56	SetLDAPSettings.Response	34
3.2.57	SetPassword.Request	35
3.2.58	SetPassword.Response	35
3.2.59	SetUserBlock.Request	35
3.2.60	SetUserBlock.Response	35
3.2.61	SetUserDetails.Request	35
3.2.62	SetUserDetails.Response	35
3.2.63	SetUserPasswordChange.Request	35
3.2.64	SetUserPasswordChange.Response	35
3.2.65	TestLdapLogin.Request	36
3.2.66	TestLdapLogin.Response	36
3.2.67	TestLdapLoginRequest	36
3.2.68	TestLdapLoginResponse	36
3.2.69	TokenInfo	36
3.2.70	UserBlock	36
3.2.71	UserBlockTrigger	37
3.2.72	UserDetails	37
3.2.73	UserInfo	37
3.2.74	UserPasswordExpired	37
3.2.75	UserPasswordExpiredTrigger	38
3.2.76	UserType	38
4	autoLogin (1.0)	39
4.1	Command Reference	39
4.1.1	SetAutoLogin	39
4.1.2	GetAutoLogin	39
4.2	Type Reference	39
4.2.1	AutoLoginConfig	39
4.2.2	GetAutoLogin.Request	39
4.2.3	GetAutoLogin.Response	39
4.2.4	SetAutoLogin.Request	39
4.2.5	SetAutoLogin.Response	39
5	bootPartition (1.0)	40
5.1	Overview	40
5.2	Command Reference	40
5.2.1	GetActivePartition	40
5.2.2	GetBootPartition	40
5.2.3	SetBootPartition	40
5.2.4	IsPartitionBootAllowed	40
5.3	Type Reference	41
5.3.1	GetActivePartition.Request	41
5.3.2	GetActivePartition.Response	41
5.3.3	GetBootPartition.Request	41
5.3.4	GetBootPartition.Response	41
5.3.5	IsPartitionBootAllowed.Request	41
5.3.6	IsPartitionBootAllowed.Response	41
5.3.7	PartitionBootAllowed	41
5.3.8	PartitionId	42

5.3.9	SetBootPartition.Request	42
5.3.10	SetBootPartition.Response	42
6	cards (1.3)	43
6.1	Overview	43
6.2	Command Reference	43
6.2.1	GetChassisInfo	43
6.2.2	GetCardStates	43
6.2.3	GetServiceStates	43
6.3	Type Reference	43
6.3.1	CardApp	43
6.3.2	CardInfo	43
6.3.3	CardLogicalState	44
6.3.4	CardLoginState	44
6.3.5	CardMeta	44
6.3.6	CardPhysicalState	44
6.3.7	CardState	44
6.3.8	Chassis	45
6.3.9	ChassisInfo	45
6.3.10	CurrentLoginAttempt	45
6.3.11	GetCardStates.Request	45
6.3.12	GetCardStates.Response	45
6.3.13	GetChassisInfo.Failure	45
6.3.14	GetChassisInfo.Request	45
6.3.15	GetChassisInfo.Response	46
6.3.16	GetServiceStates.Request	46
6.3.17	GetServiceStates.Response	46
6.3.18	ServiceInfo	46
6.3.19	ServiceLoginState	46
6.3.20	ServiceState	46
6.3.21	SoftwareInfo	46
7	chassis (1.0)	47
7.1	Command Reference	47
7.1.1	GetChassisSettings	47
7.1.2	SetChassisSettings	47
7.2	Type Reference	47
7.2.1	ChassisSettings	47
7.2.2	GetChassisSettings.Request	47
7.2.3	GetChassisSettings.Response	47
7.2.4	SetChassisSettings.Request	47
7.2.5	SetChassisSettings.Response	47
8	chassisModel (1.4)	48
8.1	Overview	48
8.2	Command Reference	48
8.2.1	NodeQuery	48
8.2.2	GetGraph	48
8.3	Type Reference	48
8.3.1	Asi	48
8.3.2	Card	49
8.3.3	ConfigurationDomain	49
8.3.4	ConfigurationId	49
8.3.5	Connector	49
8.3.6	GInfo	49
8.3.7	GetGraph.Request	50
8.3.8	GetGraph.Response	50
8.3.9	Graph	50
8.3.10	Input	50
8.3.11	Ip	50
8.3.12	IpAddr	50
8.3.13	IpInterface	50

8.3.14	IpVlan	50
8.3.15	Node	51
8.3.16	NodeQuery.Request	51
8.3.17	NodeQuery.Response	51
8.3.18	Output	51
8.3.19	PathInfo	51
8.3.20	PossibleTraits	51
8.3.21	Query	52
8.3.22	Slot	52
8.3.23	Traits	52
9	configImportExport (1.3)	54
9.1	Command Reference	54
9.1.1	ExportChassisConfig	54
9.1.2	ImportChassisConfig	54
9.1.3	LiveImportChassisConfig	54
9.2	Type Reference	54
9.2.1	ExportChassisConfig.Request	54
9.2.2	ExportChassisConfig.Response	54
9.2.3	ImportChassisConfig.Request	54
9.2.4	ImportChassisConfig.Response	55
9.2.5	ImportChassisConfigRequest	55
9.2.6	ImportChassisConfigResponse	55
9.2.7	LiveImportChassisConfig.Request	55
9.2.8	LiveImportChassisConfig.Response	55
9.2.9	LiveImportChassisConfigResponse	55
10	configReset (1.0)	56
10.1	Command Reference	56
10.1.1	ResetChassisConfig	56
10.2	Type Reference	56
10.2.1	ResetChassisConfig.Request	56
10.2.2	ResetChassisConfig.Response	56
10.2.3	ResetChassisConfigRequest	56
10.2.4	ResetChassisConfigResponse	56
11	dns (1.0)	58
11.1	Command Reference	58
11.1.1	SetDnsConfig	58
11.1.2	GetDnsConfig	58
11.1.3	ResolveName	58
11.2	Type Reference	58
11.2.1	DnsConfig	58
11.2.2	GetDnsConfig.Request	58
11.2.3	GetDnsConfig.Response	58
11.2.4	IpAddress	59
11.2.5	IpVersion	59
11.2.6	ResolveName.Request	59
11.2.7	ResolveName.Response	59
11.2.8	ResolveNameResult	59
11.2.9	SetDnsConfig.Request	59
11.2.10	SetDnsConfig.Response	59
12	firewallTypes (1.0)	60
12.1	Type Reference	60
12.1.1	IpAddress	60
12.1.2	PaginatedRequest	60
12.1.3	PaginationInfo	60
12.1.4	SocketAddress	60
12.1.5	UUID	60
13	https (1.1)	61

13.1	Command Reference	61
13.1.1	GetCertificate	61
13.1.2	InstallCertificate	61
13.1.3	RemoveCertificate	61
13.1.4	GetCertificateOverview	61
13.1.5	GetCertificateDetails	61
13.1.6	GeneratePrivateKeyAndCsr	61
13.1.7	GetStagingAreaState	61
13.1.8	GetCertificateSigningRequest	62
13.1.9	GetCertificateSigningRequestDetails	62
13.1.10	GetCertificateSigningRequestDownloadRule	62
13.1.11	DeleteStagingArea	62
13.1.12	InstallCertificateFromSignedCSR	62
13.2	Type Reference	63
13.2.1	CertificateInfo	63
13.2.2	CertificateOverview	63
13.2.3	CertificateOverview.validity	63
13.2.4	CertificateSigningRequestAttributes	63
13.2.5	CertificateSigningRequestDownloadRule	64
13.2.6	CertificateSigningRequestInfo	64
13.2.7	DeleteStagingArea.Request	64
13.2.8	DeleteStagingArea.Response	64
13.2.9	DistinguishedNameOverview	64
13.2.10	GeneratePrivateKeyAndCsr.Request	64
13.2.11	GeneratePrivateKeyAndCsr.Response	64
13.2.12	GetCertificate.Request	65
13.2.13	GetCertificate.Response	65
13.2.14	GetCertificateDetails.Request	65
13.2.15	GetCertificateDetails.Response	65
13.2.16	GetCertificateOverview.Request	65
13.2.17	GetCertificateOverview.Response	65
13.2.18	GetCertificateSigningRequest.Request	65
13.2.19	GetCertificateSigningRequest.Response	65
13.2.20	GetCertificateSigningRequestDetails.Request	65
13.2.21	GetCertificateSigningRequestDetails.Response	66
13.2.22	GetCertificateSigningRequestDownloadRule.Request	66
13.2.23	GetCertificateSigningRequestDownloadRule.Response	66
13.2.24	GetStagingAreaState.Request	66
13.2.25	GetStagingAreaState.Response	66
13.2.26	InstallCertificate.Request	66
13.2.27	InstallCertificate.Response	66
13.2.28	InstallCertificateFromSignedCSR.Request	66
13.2.29	InstallCertificateFromSignedCSR.Response	66
13.2.30	InstallCertificateRequest	67
13.2.31	RemoveCertificate.Request	67
13.2.32	RemoveCertificate.Response	67
13.2.33	StagingAreaState	67
13.2.34	SubjectAlternativeName	67
14	imageBank (1.1)	68
14.1	Overview	68
14.2	Command Reference	68
14.2.1	StoreImage	68
14.2.2	RemoveImage	68
14.2.3	ListImages	68
14.2.4	GetSpaceLocal	68
14.3	Type Reference	68
14.3.1	FileId	69
14.3.2	FileMeta	69
14.3.3	GetSpaceLocal.Request	69
14.3.4	GetSpaceLocal.Response	69
14.3.5	ListImages.Request	69

14.3.6	ListImages.Response	69
14.3.7	OperationResult	69
14.3.8	RemoveImage.Request	69
14.3.9	RemoveImage.Response	70
14.3.10	SpaceLocal	70
14.3.11	StoreImage.Request	70
14.3.12	StoreImage.Response	70
14.3.13	StoredImageMeta	70
14.3.14	UTCDateTimeISO8601	70
14.3.15	Url	70
14.3.16	UserMeta	70
15	license (1.3)	71
15.1	Command Reference	71
15.1.1	GetFeaturesInfo	71
15.1.2	GetLicense	71
15.1.3	GetHardwareId	71
15.1.4	InstallLicense	71
15.2	Type Reference	71
15.2.1	BoardFeatures	71
15.2.2	DemoPeriod	71
15.2.3	Empty	72
15.2.4	FeatureInfo	72
15.2.5	FeatureMap	72
15.2.6	FeaturesInfo	72
15.2.7	GetFeaturesInfo.Request	72
15.2.8	GetFeaturesInfo.Response	72
15.2.9	GetHardwareId.Request	72
15.2.10	GetHardwareId.Response	72
15.2.11	GetLicense.Request	73
15.2.12	GetLicense.Response	73
15.2.13	GetLicense.Response.result	73
15.2.14	InstallLicense.Request	73
15.2.15	InstallLicense.Response	73
15.2.16	InstalledFeatureInfo	73
15.2.17	License	73
15.2.18	Unknown	74
16	linkRates (1.1)	75
16.1	Command Reference	75
16.1.1	GetLinkRates	75
16.2	Type Reference	75
16.2.1	CardRates	75
16.2.2	GetLinkRates.Request	75
16.2.3	GetLinkRates.Response	75
17	maintenanceMmi (1.3)	76
17.1	Overview	76
17.2	Command Reference	76
17.2.1	DeleteConfiguration	76
17.2.2	UnblockLogin	76
17.2.3	RebootSlots	76
17.3	Type Reference	76
17.3.1	DeleteCardConfiguration	76
17.3.2	DeleteConfiguration.Request	76
17.3.3	DeleteConfiguration.Response	77
17.3.4	DeleteConfigurationId	77
17.3.5	DeleteConfigurationRequest	77
17.3.6	DeleteConfigurationResult	77
17.3.7	DeleteServiceConfiguration	77
17.3.8	Partition	77
17.3.9	RebootSlot	77

17.3.10	RebootSlotRequest	78
17.3.11	RebootSlots.Request	78
17.3.12	RebootSlots.Response	78
17.3.13	RebootSlotsRequest	78
17.3.14	RebootSlotsResponse_	78
17.3.15	UnblockLogin.Request	78
17.3.16	UnblockLogin.Response	78
17.3.17	UnblockLoginRequest	78
17.3.18	UnblockLoginResponse	78
18	multiCardUpgrade (1.4)	79
18.1	Overview	79
18.2	Command Reference	79
18.2.1	StartUpgradeJob	79
18.2.2	GetCurrentJobs	79
18.2.3	GetJobsHistory	79
18.2.4	GetJobStatus	79
18.2.5	CheckUpgradeImageCompatibility	79
18.2.6	GetCardsSlots	80
18.2.7	GetCardsStatus	80
18.2.8	GetCardsProductInfo	80
18.2.9	CyclePower	80
18.3	Type Reference	80
18.3.1	CardList	80
18.3.2	CardProductInfoResponse	80
18.3.3	CardStatus	80
18.3.4	CardStatusDump	81
18.3.5	CardStatusResponse	81
18.3.6	CardStep	81
18.3.7	CardStepRecord	81
18.3.8	CardStepState	81
18.3.9	CardUpgradeState	82
18.3.10	CardUpgradeStatus	82
18.3.11	CheckUpgradeImageCompatibility.Request	82
18.3.12	CheckUpgradeImageCompatibility.Response	82
18.3.13	CyclePower.Request	82
18.3.14	CyclePower.Response	82
18.3.15	CyclePowerRequest	82
18.3.16	CyclePowerResponse	83
18.3.17	GetCardsProductInfo.Request	83
18.3.18	GetCardsProductInfo.Response	83
18.3.19	GetCardsSlots.Request	83
18.3.20	GetCardsSlots.Response	83
18.3.21	GetCardsStatus.Request	83
18.3.22	GetCardsStatus.Response	83
18.3.23	GetCurrentJobs.Request	83
18.3.24	GetCurrentJobs.Response	83
18.3.25	GetJobStatus.Request	83
18.3.26	GetJobStatus.Response	83
18.3.27	GetJobStatusResponse	83
18.3.28	GetJobsHistory.Request	84
18.3.29	GetJobsHistory.Response	84
18.3.30	HardwareCompatibility	84
18.3.31	JobFinishedResult	84
18.3.32	JobList	84
18.3.33	JobState	84
18.3.34	JobStatus	84
18.3.35	JobStepRecord	85
18.3.36	JobStepResult	85
18.3.37	SlotConfig	85
18.3.38	SoftwareCompatibility	85
18.3.39	StartUpgradeJob.Request	86

18.3.40	StartUpgradeJob.Response	86
18.3.41	StartUpgradeJobResult	86
18.3.42	UpgradeImageCompatibility	86
18.3.43	UpgradeImageCompatibilityRequest	86
18.3.44	UpgradeImageCompatibilityResult	87
18.3.45	UpgradeJobConfig	87
18.3.46	UpgradeWithoutRebootSupport	87
19	pagination (1.0)	88
19.1	Type Reference	88
19.1.1	DataPageInfo	88
19.1.2	DataSetInfo	88
19.1.3	DataSetSlice	88
19.1.4	PaginatedDataInfo	88
19.1.5	PaginatedQueryResult	88
19.1.6	PaginatedQueryResult.noContent	89
19.1.7	PaginatedQueryResult.processing	89
20	passwordPolicy (1.0)	90
20.1	Command Reference	90
20.1.1	GetPasswordPolicies	90
20.1.2	SetPasswordPolicies	90
20.1.3	DeletePasswordPolicies	90
20.2	Type Reference	90
20.2.1	DeletePasswordPolicies.Request	90
20.2.2	DeletePasswordPolicies.Response	90
20.2.3	GetPasswordPolicies.Request	90
20.2.4	GetPasswordPolicies.Response	90
20.2.5	PasswordContent	90
20.2.6	PasswordExpiration	91
20.2.7	PasswordPolicy	91
20.2.8	SetPasswordPolicies.Request	91
20.2.9	SetPasswordPolicies.Response	91
21	productInfo (1.3)	92
21.1	Overview	92
21.2	Type Reference	92
21.2.1	HardwareInfo	92
21.2.2	ProductInfo	92
21.2.3	SoftwareBundleInfo	92
22	prometheus (1.2)	93
22.1	Overview	93
22.2	Command Reference	93
22.2.1	SetPrometheusSettings	93
22.2.2	GetPrometheusSettings	93
22.2.3	GetPrometheusEndpointNames	93
22.3	Type Reference	93
22.3.1	APITokenAuth	93
22.3.2	AuthenticationVariant	94
22.3.3	BasicAuth	94
22.3.4	GetPrometheusEndpointNames.Request	94
22.3.5	GetPrometheusEndpointNames.Response	94
22.3.6	GetPrometheusSettings.Request	94
22.3.7	GetPrometheusSettings.Response	94
22.3.8	JWTTokenAuth	94
22.3.9	PrometheusEndpointNames	94
22.3.10	PrometheusSettings	95
22.3.11	SetPrometheusSettings.Request	95
22.3.12	SetPrometheusSettings.Response	95
23	remoteConfigExport (1.0)	96

23.1	Command Reference	96
23.1.1	SetRemoteConfigExportSchedule	96
23.1.2	GetRemoteConfigExportSchedule	96
23.1.3	SetRemoteConfigExportInstance	96
23.1.4	GetAllRemoteConfigExportInstances	96
23.1.5	DeleteRemoteConfigExportInstance	96
23.1.6	GetAllRemoteConfigExportStatus	96
23.1.7	PerformRemoteConfigExport	96
23.2	Type Reference	96
23.2.1	DeleteRemoteConfigExportInstance.Request	96
23.2.2	DeleteRemoteConfigExportInstance.Response	96
23.2.3	GetAllRemoteConfigExportInstances.Request	97
23.2.4	GetAllRemoteConfigExportInstances.Response	97
23.2.5	GetAllRemoteConfigExportStatus.Request	97
23.2.6	GetAllRemoteConfigExportStatus.Response	97
23.2.7	GetRemoteConfigExportSchedule.Request	97
23.2.8	GetRemoteConfigExportSchedule.Response	97
23.2.9	Instance	97
23.2.10	Instances	97
23.2.11	LastRun	97
23.2.12	PerformRemoteConfigExport.Request	97
23.2.13	PerformRemoteConfigExport.Response	98
23.2.14	RemoteConfigExportInstance	98
23.2.15	RemoteConfigExportInstanceId	98
23.2.16	RemoteConfigExportLastRun	98
23.2.17	RemoteConfigExportScheduleParameters	98
23.2.18	RemoteConfigExportStatus	98
23.2.19	SFTPRemoteConfigExport	98
23.2.20	Schedule	98
23.2.21	ScheduleParameters	99
23.2.22	SetRemoteConfigExportInstance.Request	99
23.2.23	SetRemoteConfigExportInstance.Response	99
23.2.24	SetRemoteConfigExportSchedule.Request	99
23.2.25	SetRemoteConfigExportSchedule.Response	99
23.2.26	Status	99
23.2.27	Statuses	99
23.2.28	WallClockTimePoint	99
24	security (1.0)	100
24.1	Command Reference	100
24.1.1	GetCardSecurityStatus	100
24.2	Type Reference	100
24.2.1	CardSecurityStatus	100
24.2.2	GetCardSecurityStatus.Request	100
24.2.3	GetCardSecurityStatus.Response	100
25	services (1.4)	101
25.1	Command Reference	101
25.1.1	GetRoutedNodes	101
25.1.2	GetInputServices	101
25.1.3	GetInputServicesPage	101
25.1.4	GetOutputServices	101
25.1.5	GetServiceDependencies	101
25.2	Type Reference	101
25.2.1	ChildSourceInfo	101
25.2.2	FlowSinkInfo	102
25.2.3	FlowSourceInfo	102
25.2.4	FlowSourcePage	102
25.2.5	FlowSourceServiceInfo	102
25.2.6	FlowSourceStorage	102
25.2.7	GetInputServices.Request	102
25.2.8	GetInputServices.Response	102

25.2.9	GetInputServicesPage.Request	103
25.2.10	GetInputServicesPage.Response	103
25.2.11	GetInputServicesPageRequest	103
25.2.12	GetInputServicesRequest	103
25.2.13	GetOutputServices.Request	103
25.2.14	GetOutputServices.Response	103
25.2.15	GetRoutedNodes.Request	103
25.2.16	GetRoutedNodes.Response	103
25.2.17	GetServiceDependencies.Request	103
25.2.18	GetServiceDependencies.Response	104
25.2.19	InputServicePresenceQuery	104
25.2.20	InputServiceQuery	104
25.2.21	ServiceDependencyInfo	104
26 snmp (1.2)		105
26.1	Overview	105
26.2	Command Reference	105
26.2.1	GetTrapDestinations	105
26.2.2	UpdateTrapDestinations	105
26.2.3	GetSnmpConfigV3	106
26.2.4	UpdateSnmpConfigV3	106
26.2.5	GetEngineId	106
26.2.6	GetInformStatistics	106
26.2.7	TestSnmpDestination	106
26.3	Type Reference	106
26.3.1	AuthProtocol	106
26.3.2	EngineIdV3WithStatus	107
26.3.3	GetEngineId.Request	107
26.3.4	GetEngineId.Response	107
26.3.5	GetInformStatistics.Request	107
26.3.6	GetInformStatistics.Response	107
26.3.7	GetSnmpConfigV3.Request	107
26.3.8	GetSnmpConfigV3.Response	107
26.3.9	GetTrapDestinations.Request	107
26.3.10	GetTrapDestinations.Response	107
26.3.11	InformDestinationV2WithStatistics	107
26.3.12	InformDestinationV3WithStatistics	108
26.3.13	InformDestinationV3WithStatus	108
26.3.14	InformStatus	108
26.3.15	InformStatusData	108
26.3.16	PrivConfig	109
26.3.17	PrivProtocol	109
26.3.18	Result	109
26.3.19	Result.ok	109
26.3.20	SecurityConfig	109
26.3.21	SecurityNameMap	109
26.3.22	SecurityNameV3WithStatus	109
26.3.23	SnmpConfigV3	110
26.3.24	SnmpConfigV3VerificationResponse	110
26.3.25	SnmpConfigV3WithStatus	110
26.3.26	SnmpDestinationSelector	110
26.3.27	SnmpDestinationType	110
26.3.28	SnmpInformDestination	111
26.3.29	SnmpInformDestinationV3	111
26.3.30	SnmpTrapDestination	111
26.3.31	SnmpTrapDestinationList	111
26.3.32	SnmpTrapDestinationV3	111
26.3.33	TestSnmpDestination.Request	111
26.3.34	TestSnmpDestination.Response	111
26.3.35	TrapDestinationV3WithStatus	112
26.3.36	UpdateSnmpConfigV3.Request	112
26.3.37	UpdateSnmpConfigV3.Response	112

26.3.38	UpdateTrapDestinations.Request	112
26.3.39	UpdateTrapDestinations.Response	112
27	syslog (1.1)	113
27.1	Overview	113
27.2	Command Reference	113
27.2.1	GetRemoteSyslogDestinations	113
27.2.2	SetRemoteSyslogDestinations	113
27.2.3	TestSyslogDestination	113
27.3	Type Reference	113
27.3.1	GetRemoteSyslogDestinations.Request	113
27.3.2	GetRemoteSyslogDestinations.Response	113
27.3.3	RemoteSyslogDestination	113
27.3.4	RemoteSyslogDestinationList	114
27.3.5	Result	114
27.3.6	Result.ok	114
27.3.7	SetRemoteSyslogDestinations.Request	114
27.3.8	SetRemoteSyslogDestinations.Response	114
27.3.9	SyslogDestinationSelector	114
27.3.10	TestSyslogDestination.Request	114
27.3.11	TestSyslogDestination.Response	114
28	uptime (1.0)	115
28.1	Command Reference	115
28.1.1	GetLoginUptime	115
28.1.2	GetSystemUptime	115
28.2	Type Reference	115
28.2.1	GetLoginUptime.Request	115
28.2.2	GetLoginUptime.Response	115
28.2.3	GetSystemUptime.Request	115
28.2.4	GetSystemUptime.Response	115
28.2.5	LoginUptimeResponse	115
28.2.6	SystemUptimeResponse	115
28.2.7	uptime	116
29	userPolicy (1.0)	117
29.1	Command Reference	117
29.1.1	GetGlobalUserPolicy	117
29.1.2	SetGlobalUserPolicy	117
29.2	Type Reference	117
29.2.1	GetGlobalUserPolicy.Request	117
29.2.2	GetGlobalUserPolicy.Response	117
29.2.3	GlobalUserPolicy	117
29.2.4	SetGlobalUserPolicy.Request	117
29.2.5	SetGlobalUserPolicy.Response	117
29.2.6	UserPolicy	117
30	websocket (1.0)	118
30.1	Command Reference	118
30.1.1	UseToken	118
30.2	Type Reference	118
30.2.1	UseToken.Request	118
30.2.2	UseToken.Response	118

1 Overview

Changelog

5.6

Added

- configReset 1.0

5.5

Changed

- cards from 1.2 to 1.3

5.4

Changed

- configImportExport from 1.2 to 1.3

5.1

Changed

- chassisModel from 1.3 to 1.4

5.0

Removed

- systemTimeSettings are removed from mmi API and moved to timex API (see separate document)
- ledSettings is removed from mmi API
- all previous versions of above APIs up to version 4.7 still exist and can be used

4.7

Added

- remoteConfigExport 1.0

Changed

- prometheus from 1.1 to 1.2
- chassisModel from 1.2 to 1.3

4.6

Added

- uptime 1.0

4.5

Changed

- maintenanceMmi from 1.2 to 1.3
- multiCardUpgrade from 1.3 to 1.4
- productInfo from 1.2 to 1.3

4.4

Changed

- systemTimeSettings from 1.3 to 1.4
- imageBank from 1.0 to 1.1

4.3

Added

- imagebank 1.0

Changed

- chassisModel from 1.1 to 1.2
- prometheus from 1.0 to 1.1
- snmp from 1.1 to 1.2
- syslog from 1.0 to 1.1

4.2

Changed

- alarms from 1.4 to 1.5
- chassisModel from 1.0 to 1.1
- multiCardUpgrade from 1.2 to 1.3
- productInfo from 1.1 to 1.2
- snmp from 1.0 to 1.1

2 alarms (1.5)

2.1 Overview

Changelog :

Version 1.5: - Adds method to get all configured alarm overrides; GetAllAlarmOverrides - AlarmOverrideDomain's config is now split into a variant which takes either a configuration object id and a slot, or a configuration object id and a string. The former is equivalent to v1.4's config domain, whereas the latter is added to allow for specifying config object level overrides on config object ids that are not fixed to a specific slot. The domain name can be fetched, when applicable from the chassis model.

Affects RPC methods SetAlarmOverrides, GetAlarmOverrides and GetAllAlarmOverrides where config domain is concerned.

2.2 Command Reference

2.2.1 GetActiveAlarms

- message **GetActiveAlarms.Request**
- message **GetActiveAlarms.Response**

2.2.2 GetAlarmsHistory

- message **GetAlarmsHistory.Request**
- message **GetAlarmsHistory.Response**

2.2.3 ClearAlarmsHistory

- message **ClearAlarmsHistory.Request**
- message **ClearAlarmsHistory.Response**

2.2.4 GetRegisteredAlarms

- message **GetRegisteredAlarms.Request**
- message **GetRegisteredAlarms.Response**

2.2.5 GetAlarmOverrides

- message **GetAlarmOverrides.Request**
- message **GetAlarmOverrides.Response**

2.2.6 GetAllAlarmOverrides

- message **GetAllAlarmOverrides.Request**
- message **GetAllAlarmOverrides.Response**

2.2.7 SetAlarmOverrides

- message **SetAlarmOverrides.Request**
- message **SetAlarmOverrides.Response**

2.2.8 DeleteAlarmOverrides

- message `DeleteAlarmOverrides.Request`
- message `DeleteAlarmOverrides.Response`

2.3 Type Reference

2.3.1 Alarm

Description of an alarm

struct

alarmId	string ID of the alarm, as described in the alarm documentation
alarmName	string Human readable name of the alarm
alarmDescription	string Human readable general description of the alarm
severity	Severity How critical the alarm is
details	string Human readable information about the instance on the alarm
configObjectId	firewallTypes.UUID UUID of the configuration object
configObjectType	string Type of the configuration object
configObjectDescription	string Human readable information about what has been configured
configObjectSlot	int The slot which is currently running the configuration
configObjectLogin	ConfigLogin (private) Login of the owner of the configuration
configObjectHandler	string (private) Configuration editor
configObjectLabel	string
timeSet	bigint When this alarm was raised
timeClear	optional bigint When this alarm was cleared, if ever
position	int Index used when requesting paginated alarms
filtered	bool Not used. Always false.

2.3.2 AlarmGroupId

struct

slot	int
tag	string

2.3.3 AlarmOverrideConfig

For alarm with id 'alarmId', severity should map to 'severity' in domain 'domain'

struct

alarmId	string
parameters	AlarmOverrideConfigParameters
info	AlarmOverrideConfigInfo

2.3.4 AlarmOverrideConfigInfo

struct

alarmName	string
alarmDescription	string
alarmDomain	optional string

2.3.5 AlarmOverrideConfigParameters

struct

domain	AlarmOverrideDomain
severity	AlarmOverrideSeverity
setDelay	int
clearDelay	int

2.3.6 AlarmOverrideDomain

Specifies domain for a AlarmOverrideConfig

variant

system	SystemDomain
card	CardDomain
config	ConfigObject

2.3.7 AlarmOverrideSeverity

enum

FILTERED
NOTICE
WARNING
MAJOR
CRITICAL

2.3.8 AlarmsQuery

struct

since	optional AlarmsSince
configObjectTypes	optional list of string
count	optional int
enableRootCauseAnalysis	bool

2.3.9 AlarmsSince

struct

sessionId	firewallTypes.UUID
position	int

2.3.10 CardDomain

Used to specify slot-wide alarm override domain. Slot must be between 1 and 14, inclusive.

struct

slot	int
------	------------

2.3.11 ClearAlarmsHistory.Request

empty **struct**

2.3.12 ClearAlarmsHistory.Response

struct

ok	bool
----	-------------

2.3.13 ConfigLogin

variant

slot	int
service	string

2.3.14 ConfigObject

struct

configObjectId	UUID
domainType	ConfigObjectDomain

2.3.15 ConfigObjectDomain

Used to specify alarm override for a specific configuration.

Per configuration alarm overrides are specified either by slot or by domain name, depending on the product. The chassis model API provides information about which domain is the appropriate one for a given configuration. When the config object overrides is defined with a domain name for the configuration. If the chassis model specifies no such domain, the default 'slot' should be used.

Slot must be between 1 and 14. Null-UUID is not allowed.

variant

slot	int
domain	string

2.3.16 DeleteAlarmOverrideRequest

Delete a group of overrides

struct

overrides	set of UUID
-----------	-------------

2.3.17 DeleteAlarmOverrideResponse

struct

ok	bool
status	map from UUID to DeleteAlarmOverrideResult

2.3.18 DeleteAlarmOverrideResult

Result codes for each entry in DeleteAlarmoverrideRequest

OK: Request is valid UNKNOWN: Unspecified error INVALID_UUID_ERROR: Entry identified by UUID was not found DATABASE_ERROR: Database interaction failure

enum

OK
UNKNOWN
INVALID_UUID_ERROR
DATABASE_ERROR

2.3.19 DeleteAlarmOverrides.Request

DeleteAlarmOverrideRequest

2.3.20 DeleteAlarmOverrides.Response

DeleteAlarmOverrideResponse

2.3.21 GetActiveAlarms.Request

struct

query **optional AlarmsQuery**

2.3.22 GetActiveAlarms.Response

struct

data **list of Alarm**
sessionId **firewallTypes.UUID**

2.3.23 GetAlarmOverrides.Request

struct

slice **pagination.DataSetSlice**

2.3.24 GetAlarmOverrides.Response

RegisteredAlarmOverrides

2.3.25 GetAlarmsHistory.Request

struct

query **optional AlarmsQuery**

2.3.26 GetAlarmsHistory.Response

struct

data **list of Alarm**
sessionId **firewallTypes.UUID**

2.3.27 GetAllAlarmOverrides.Request

empty **struct**

2.3.28 GetAllAlarmOverrides.Response

struct

overrides **map from UUID to AlarmOverrideConfig**

2.3.29 GetRegisteredAlarms.Request

empty **struct**

2.3.30 GetRegisteredAlarms.Response

RegisteredAlarms

2.3.31 Hysteresis

struct

setDelay	int
clearDelay	int

2.3.32 RegisteredAlarm

struct

id	string
name	string
description	string
domain	optional string
severity	AlarmOverrideSeverity

2.3.33 RegisteredAlarmOverrides

struct

result	pagination.PaginatedQueryResult
overrides	map from UUID to AlarmOverrideConfig

2.3.34 RegisteredAlarms

struct

alarms	map from AlarmGroupId to list of RegisteredAlarm
--------	--

2.3.35 SetAlarmOverride

struct

alarmId	string
domain	AlarmOverrideDomain
severity	optional AlarmOverrideSeverity
hysteresis	optional Hysteresis

2.3.36 SetAlarmOverrideRequest

Request to set or update a group of overrides. Returns ok

struct

overrides	map from UUID to SetAlarmOverride
-----------	-----------------------------------

2.3.37 SetAlarmOverrideResponse

struct

ok	bool
status	map from UUID to SetAlarmOverrideResult

2.3.38 SetAlarmOverrideResult

Result codes for each entry in SetAlarmOverrideRequest

OK: Request is valid UNKNOWN: Unspecified error DATABASE_ERROR: Database interaction failure ALARM_ID_ERROR: Specified alarm ID was not found OVERRIDE_PARAMETERS_ERROR: Invalid domain parameters or hysteresis parameters MULTIPLE_DEFINITION_ERROR: An override for alarm ID with same domain is already registered

enum

OK
UNKNOWN
DATABASE_ERROR
ALARM_ID_ERROR
OVERRIDE_PARAMETERS_ERROR
MULTIPLE_DEFINITION_ERROR
INVALID_UUID_ERROR

2.3.39 SetAlarmOverrides.Request

SetAlarmOverrideRequest

2.3.40 SetAlarmOverrides.Response

SetAlarmOverrideResponse

2.3.41 Severity

enum

EVENT
NOTICE
WARNING
MAJOR
CRITICAL

2.3.42 SystemDomain

Used to specify system-wide alarm override domain

empty struct

3 authentication (1.6)

3.1 Command Reference

3.1.1 BeginSession

Authenticates the user and initiates a new session.

- message **BeginSession.Request**
- message **BeginSession.Response**

3.1.2 RefreshSession

Extends the current session and issues a new access token.

- message **RefreshSession.Request**
- message **RefreshSession.Response**

3.1.3 EndSession

Terminates the current session.

- message **EndSession.Request**
- message **EndSession.Response**

3.1.4 GetSessions

- message **GetSessions.Request**
- message **GetSessions.Response**

3.1.5 GetTokenInfo

Retrieves basic information about a token.

This command is mainly useful for figuring out how much time a token has to live and if it needs to be refreshed.

- message **GetTokenInfo.Request**
- message **GetTokenInfo.Response**

3.1.6 GetUsers

Retrieves a list of all users in the system that are visible to the current user.

- message **GetUsers.Request**
- message **GetUsers.Response**

3.1.7 CreateUser

Creates a new user in the system.

- message **CreateUser.Request**
- message **CreateUser.Response**

3.1.8 DeleteUser

Deletes a user from the system.

The user will be automatically logged out.

- message **DeleteUser.Request**
- message **DeleteUser.Response**

3.1.9 SetPassword

Changes user's password.

The user will be automatically logged out.

- message **SetPassword.Request**
- message **SetPassword.Response**

3.1.10 SetGroupMembership

Modifies user's access level by changing his group membership.

The user will be automatically logged out.

- message **SetGroupMembership.Request**
- message **SetGroupMembership.Response**

3.1.11 SetUserDetails

- message **SetUserDetails.Request**
- message **SetUserDetails.Response**

3.1.12 SetUserBlock

Blocks or unblocks a user. Blocked users can't log in, i.e. BeginSession call will always fail for them.

- message **SetUserBlock.Request**
- message **SetUserBlock.Response**

3.1.13 SetUserPasswordChange

Forces a user to change password at the next login.

- message **SetUserPasswordChange.Request**
- message **SetUserPasswordChange.Response**

3.1.14 GetUserGroups

Retrieves a list of all user groups in the system that are visible to the current user.

- message **GetUserGroups.Request**
- message **GetUserGroups.Response**

3.1.15 PatchUserGroups

- message **PatchUserGroups.Request**
- message **PatchUserGroups.Response**

3.1.16 SetLDAPSettings

- message **SetLDAPSettings.Request**
- message **SetLDAPSettings.Response**

3.1.17 GetLDAPSettings

- message **GetLDAPSettings.Request**
- message **GetLDAPSettings.Response**

3.1.18 GetLDAPGroupMappings

- message **GetLDAPGroupMappings.Request**
- message **GetLDAPGroupMappings.Response**

3.1.19 SetLDAPGroupMappings

- message **SetLDAPGroupMappings.Request**
- message **SetLDAPGroupMappings.Response**

3.1.20 TestLdapLogin

For given username/password and for given LDAP-settings and group mappings, this method:

- tests if a user will get authenticated and authorized by LDAP server
- provides detailed information about how the procedure went, indicating problems and their possible causes
- message **TestLdapLogin.Request**
- message **TestLdapLogin.Response**

3.2 Type Reference

3.2.1 BeginSession.Request

The variant specifies login method and supplies necessary credentials.

struct

auth	LoginMethod
	The authentication method and corresponding credentials

3.2.2 BeginSession.Response

Contains access token which can be used as a proof of identity for this session.

struct

accessToken	string
-------------	---------------

3.2.3 CreateUser.Request

CreateUserRequest

3.2.4 CreateUser.Response

Contains an unique ID which can be used to refer to this user.

struct

userId	firewallTypes.UUID
--------	---------------------------

3.2.5 CreateUserRequest

struct

username	string Desired user name
password	string Desired password
groups	GroupMembership Define permissions by specifying which groups the user belongs to.
details	UserDetails
forcePasswordChange	bool If set to true then a new user will be forced to change a password on first login.

3.2.6 Credentials

struct

username	string
password	string

3.2.7 DeleteUser.Request

Specifies the user account which is to be deleted.

struct

userId	firewallTypes.UUID Unique ID of the user.
--------	---

3.2.8 DeleteUser.Response

empty **struct**

3.2.9 EndSession.Request

empty **struct**

3.2.10 EndSession.Response

empty **struct**

3.2.11 GetLDAPGroupMappings.Request

empty **struct**

3.2.12 GetLDAPGroupMappings.Response

struct

settings **list of LDAPGroupMapping**

3.2.13 GetLDAPSettings.Request

empty **struct**

3.2.14 GetLDAPSettings.Response

struct

settings **LDAPSettings**

3.2.15 GetSessions.Request

GetSessionsRequest

3.2.16 GetSessions.Response

GetSessionsResponse

3.2.17 GetSessionsRequest

struct

filter **optional SessionStateFilter**
If not set then result will contain both active and ended sessions. Setting this allows getting only active or only ended sessions.

slice **optional pagination.DataSetSlice**
If not set then all sessions matching a filter will be returned. If set then a slice of sessions matching a filter will be returned.

3.2.18 GetSessionsResponse

struct

result **pagination.PaginatedQueryResult**
Meta information about returned data and about whole dataset.

data **map from UUID to SessionInfo**
Actual data, contains all sessions or requested page/slice of sessions.

3.2.19 GetTokenInfo.Request

struct

accessToken	string Raw access token (without the 'Bearer' part)
-------------	---

3.2.20 GetTokenInfo.Response

TokenInfo

3.2.21 GetUserGroups.Request

empty **struct**

3.2.22 GetUserGroups.Response

struct

data	list of GroupInfo
------	---------------------------------

3.2.23 GetUsers.Request

empty **struct**

3.2.24 GetUsers.Response

struct

data	list of UserInfo
------	--------------------------------

3.2.25 GroupConfig

struct

passwordPolicy	optional UUID If set then this is ID of PasswordPolicy (see passwordPolicy module) which will be enforced on users who belong to this group.
----------------	--

guiInactivityTimeout	optional int If set then GUI will automatically log out user after this number of seconds of inactivity in a browser.
----------------------	---

3.2.26 GroupInfo

Contains information about one user group

struct

id	firewallTypes.UUID Unique ID the group which can be used to refer it.
----	---

name	string
config	GroupConfig

3.2.27 GroupMembership

Lists groups user is member of.

struct

main	firewallTypes.UUID User's main group
------	---

3.2.28 LDAPBindMethod

enum

UNKNOWN	
SIMPLE	
SASL	

3.2.29 LDAPGroupMapping

struct

info	GroupInfo
ldapMapping	optional string

3.2.30 LDAPLogin

Type and credentials for LDAP login procedure.

struct

username	string
password	string

3.2.31 LDAPSettings

.

struct

serverAddress	optional string If this optional has a value it means that LDAP support is enabled. Server's address must begin with 'ldap://' or ldaps://. If address begins with ldap:// we will use LDAP with StartTLS. If address begins with ldaps:// we will use LDAPS.
	optional string

rootDN	NOT USED •
managerCredentials	optional Credentials
userSearchParams	optional LDAPUserSearchParams
bindMethod	LDAPBindMethod
rootCACertificate	optional string

3.2.32 LDAPUserSearchParams

•

struct

searchBase	string The DN of the entry at which to start the search. The field normally looks something like "ou=people". •
searchFilter	string Determines the query to be run to identify the user record. It must contain string which will be replaced with username provided when making a call to BeginSession. •
attributeName	string Group membership attribute of a user.

3.2.33 LdapGroupMappingsSetter

struct

groupId	firewallTypes.UUID UUID of a "local" group, see GetUserGroups method
ldapMapping	optional string DN of a LDAP group which will match a "local" group specified by groupId

3.2.34 LdapLoginStepRecord

Information about execution of a step of LDAP login process.

struct

name	string Human readable short description of a step
result	LdapLoginStepResult When step finished this will be SUCCESS or FAILURE, if step got into undefined then this will be UNKNOWN and indicates internal problem
reports	list of string Human readable additional information reported by a step

3.2.35 LdapLoginStepResult

enum

UNKNOWN

SUCCESS

FAILURE

WARNING

3.2.36 LocalLogin

Type and credentials for local login procedure.

struct

username

string

password

string

3.2.37 LoginMethod

variant

local

LocalLogin

ldap

LDAPLogin

auto

LoginMethod.auto

3.2.38 LoginMethod.auto

empty struct

3.2.39 PasswordExpiration

variant

neverExpires

PasswordExpiration.neverExpires

expiresIn

int

expired

UserPasswordExpired

3.2.40 PasswordExpiration.neverExpires

empty struct

3.2.41 PatchUserGroups.Request

struct

data

map from UUID to GroupConfig

3.2.42 PatchUserGroups.Response

empty **struct**

3.2.43 RefreshSession.Request

empty **struct**

3.2.44 RefreshSession.Response

Contains the new extended access token for this session.

struct

accessToken	string
-------------	---------------

3.2.45 SessionEnd

struct

endTime	bigint
endReason	SessionEndReason
details	optional string

3.2.46 SessionEndReason

enum

UserEndedSession	User has explicitly ended his/her own session by calling End-Session. E.g. by clicking "logout" button in GUI.
SystemEndedSession	System ended session for some reason. E.g. user was blocked, or user's password was changed.
Expired	Session's validity period expired.

3.2.47 SessionInfo

struct

userId	UUID
sessionId	UUID
userName	string
userType	UserType
startTime	bigint
clientId	string
permissions	set of string
state	SessionState
sshSlot	optional int

3.2.48 SessionState

variant

active	SessionState.active
ended	SessionEnd

3.2.49 SessionState.active

empty **struct**

3.2.50 SessionStateFilter

enum

GetActive
GetEnded

3.2.51 SetGroupMembership.Request

Replaces the current group membership with the provided structure.

struct

userId	firewallTypes.UUID Unique ID of the user.
groups	GroupMembership

3.2.52 SetGroupMembership.Response

empty **struct**

3.2.53 SetLDAPGroupMappings.Request

struct

mappings	list of LdapGroupMappingsSetter
----------	--

3.2.54 SetLDAPGroupMappings.Response

struct

ok	bool
----	-------------

3.2.55 SetLDAPSettings.Request

struct

settings	LDAPSettings
----------	---------------------

3.2.56 SetLDAPSettings.Response

struct

ok **bool**

3.2.57 SetPassword.Request

Replaces the current password with the new one.

struct

userId	firewallTypes.UUID Unique ID of the user.
password	string

3.2.58 SetPassword.Response

empty **struct**

3.2.59 SetUserBlock.Request

struct

userId	UUID
block	bool

3.2.60 SetUserBlock.Response

empty **struct**

3.2.61 SetUserDetails.Request

struct

userId	firewallTypes.UUID
details	UserDetails

3.2.62 SetUserDetails.Response

empty **struct**

3.2.63 SetUserPasswordChange.Request

struct

userId	UUID
--------	-------------

3.2.64 SetUserPasswordChange.Response

empty **struct**

3.2.65 TestLdapLogin.Request

TestLdapLoginRequest

3.2.66 TestLdapLogin.Response

TestLdapLoginResponse

3.2.67 TestLdapLoginRequest

struct

username	string As if passed to BeginSession
password	string As if passed to BeginSession
ldapSettings	LDAPSettings The same object that would be passed to SetLDAPSettings
groupMappings	list of LdapGroupMappingsSetter The same object that would be passed to SetLDAPGroupMappings

3.2.68 TestLdapLoginResponse

struct

ok	bool
stepRecord	list of LdapLoginStepRecord

3.2.69 TokenInfo

General information about an access token.

@param username User name associated with this token.

@param expires Time in seconds this token will remain valid for.

@param valid Indicates whether this token is valid or not.

struct

username	string
expires	int
valid	bool

3.2.70 UserBlock

struct

trigger	UserBlockTrigger What caused a user to be blocked.
details	optional string Might provide additional information why user was blocked.

when

bigint

UTC time of when user was blocked. In seconds since beginning of unix epoch.

3.2.71 UserBlockTrigger

enum

Unspecified
RpcRequest
PasswordExpired

3.2.72 UserDetails

struct

name **string**
email **string**

3.2.73 UserInfo

User account information

struct

id **firewallTypes.UUID**
Unique ID which can be used to refer to this user.

username **string**
Account username.

groups **GroupMembership**
Describes which groups the user belongs to.

details **UserDetails**

block **optional UserBlock**
If has a value it means a user is blocked, i.e. is not allowed to log in.

passwordExpiration **PasswordExpiration**

3.2.74 UserPasswordExpired

struct

trigger **UserPasswordExpiredTrigger**
What caused password expiration.

details **optional string**
Might provide additional information.

when **bigint**
UTC time when password expiration was detected. In seconds since beginning of unix epoch.

3.2.75 UserPasswordExpiredTrigger

enum

Unspecified

RpcRequest

PasswordTooOld

UserCreation

3.2.76 UserType

enum

Local

Ldap

Auto

4 autoLogin (1.0)

4.1 Command Reference

4.1.1 SetAutoLogin

- message **SetAutoLogin.Request**
- message **SetAutoLogin.Response**

4.1.2 GetAutoLogin

- message **GetAutoLogin.Request**
- message **GetAutoLogin.Response**

4.2 Type Reference

4.2.1 AutoLoginConfig

struct

userGroupId

optional UUID

If this has a value then "auto login" feature will be enabled.
Empty optional disables the feature.

4.2.2 GetAutoLogin.Request

empty struct

4.2.3 GetAutoLogin.Response

AutoLoginConfig

4.2.4 SetAutoLogin.Request

AutoLoginConfig

4.2.5 SetAutoLogin.Response

empty struct

5 bootPartition (1.0)

5.1 Overview

All cards in X platform have two software partitions – Partition_0 and Partition_1. Only one of them is in use at any given moment and that one is called “active”. The other partition is called “inactive”.

When software image is installed during upgrade procedure it is installed on “inactive” partition and then usually we swap active and inactive partitions (the inactive partition is set as “boot” partition which means it will become “active” on next boot) and we reboot a card.

It is also possible to perform upgrade without reboot. In such case the software image is installed on “inactive” partiton and no further actions occur. It is responsibility of a user to possibly change a boot partition and reboot at a desired moment in time.

5.2 Command Reference

5.2.1 GetActivePartition

Returns ID of the partition from which the system is currently booted from (i.e. active partition).

- message **GetActivePartition.Request**
- message **GetActivePartition.Response**

5.2.2 GetBootPartition

Returns ID of the partition which will be used next time the system boots.

- message **GetBootPartition.Request**
- message **GetBootPartition.Response**

5.2.3 SetBootPartition

Sets which partition will be used on next system boot. Not supported on X5.

- message **SetBootPartition.Request**
- message **SetBootPartition.Response**

5.2.4 IsPartitionBootAllowed

Checks if it's possible to use a given partition as boot partition.

It is not possible to use a partition for booting system in some situations, e.g.:

- if SW-image on selected partition is not compatible with SW-image on active (currently running) partition – in such case selecting a given partition as boot might make card now work correctly after reboot
- old image on inactive partition (or no image at all)
- ongoing upgrade (then the state of inactive partition is udedinfed, files are being installed there)

Some cards might allow to use SW-image even if it's not entirely compatible with SW-image on the active partiton, but in such case a user should be warned about a danger. In such case this method will return YesWithConsent.

- message **IsPartitionBootAllowed.Request**
- message **IsPartitionBootAllowed.Response**

5.3 Type Reference

5.3.1 GetActivePartition.Request

struct

slot	int
------	-----

5.3.2 GetActivePartition.Response

struct

partitionId	optional PartitionId
-------------	----------------------

5.3.3 GetBootPartition.Request

struct

slot	int
------	-----

5.3.4 GetBootPartition.Response

struct

partitionId	optional PartitionId
-------------	----------------------

5.3.5 IsPartitionBootAllowed.Request

struct

slot	int
partitionId	PartitionId

5.3.6 IsPartitionBootAllowed.Response

struct

status	PartitionBootAllowed
info	string

5.3.7 PartitionBootAllowed

enum

Yes
No
YesWithConsent

5.3.8 PartitionId

enum

Partition_0	
Partition_1	

5.3.9 SetBootPartition.Request

struct

slot	int
partitionId	PartitionId

5.3.10 SetBootPartition.Response

struct

status	bool
info	string

6 cards (1.3)

6.1 Overview

Changelog

1.3

Added

- added X5 to Chassis enum

6.2 Command Reference

6.2.1 GetChassisInfo

- message **GetChassisInfo.Request**
- message **GetChassisInfo.Response**
- message **GetChassisInfo.Failure**

6.2.2 GetCardStates

Returns list of entries for each card which is either physically present and/or for which system has logical state information.

- message **GetCardStates.Request**
- message **GetCardStates.Response**

6.2.3 GetServiceStates

Returns list of services

- message **GetServiceStates.Request**
- message **GetServiceStates.Response**

6.3 Type Reference

6.3.1 CardApp

struct

description	string
icon	string
url	string
cat	string

6.3.2 CardInfo

struct

name	string
serial	string
slot	int

uiAccessUrl	string
accessUrl	string
meta	CardMeta
features	list of string
apps	map from string to CardApp

6.3.3 CardLogicalState

struct

login	optional CurrentLoginAttempt If has a value it means that a card is trying to login.
configSwInfo	optional SoftwareInfo If has a value it means that the system has a configuration for a card. Value defines sw-info of a card which was the one migrating and using the configuration.

6.3.4 CardLoginState

enum

NOT_LOGGED_IN
 LOGGED_IN
 BLOCKED

6.3.5 CardMeta

struct

width	int
height	int
pictureBaseUrl	string

6.3.6 CardPhysicalState

struct

present **bool**

6.3.7 CardState

struct

slot	int
physical	CardPhysicalState
logical	optional CardLogicalState

6.3.8 Chassis

enum

X10_1RU
X20_2RU
X5

6.3.9 ChassisInfo

struct

chassisType	Chassis
cards	map from UUID to CardInfo
services	map from UUID to ServiceInfo

6.3.10 CurrentLoginAttempt

Describes a situation of a card which tries to log in.

struct

loginState	CardLoginState Current state of login procedure.
swInfo	optional SoftwareInfo Information about card's software provided by a card when it is trying to log in.
diagnosticMessage	optional string Human readable string which might be provided by the system to give a hint about problems if they occur.

6.3.11 GetCardStates.Request

empty **struct**

6.3.12 GetCardStates.Response

struct

cards	list of CardState
-------	---------------------------------

6.3.13 GetChassisInfo.Failure

empty **struct**

6.3.14 GetChassisInfo.Request

empty **struct**

6.3.15 GetChassisInfo.Response

struct

data **ChassisInfo**

6.3.16 GetServiceStates.Request

empty **struct**

6.3.17 GetServiceStates.Response

struct

services **list of ServiceState**

6.3.18 ServiceInfo

struct

name	string
slot	int
features	list of string
apps	map from string to CardApp

6.3.19 ServiceLoginState

enum

NOT_LOGGED_IN	
LOGGED_IN	

6.3.20 ServiceState

struct

name	string
displayName	string
status	ServiceLoginState
currentSlot	optional int
providerSlots	list of int

6.3.21 SoftwareInfo

struct

softwareId	string
displayName	string
ver	string
compatibleConfigSwIds	list of string

7 chassis (1.0)

7.1 Command Reference

7.1.1 GetChassisSettings

- message **GetChassisSettings.Request**
- message **GetChassisSettings.Response**

7.1.2 SetChassisSettings

- message **SetChassisSettings.Request**
- message **SetChassisSettings.Response**

7.2 Type Reference

7.2.1 ChassisSettings

struct

name **string**

7.2.2 GetChassisSettings.Request

empty **struct**

7.2.3 GetChassisSettings.Response

ChassisSettings

7.2.4 SetChassisSettings.Request

ChassisSettings

7.2.5 SetChassisSettings.Response

empty **struct**

8 chassisModel (1.4)

8.1 Overview

Changelog

1.4

Changed

- change in ConfigurationDomain struct
 - a field domain was renamed to domainId
 - a field displayName was added

1.3

Added

- Added optional configLabel member to ConfigurationId trait
- Added Output trait

1.2

Added

- IpAddr and IpVlan traits

Changed

- added physicalLabel to Connector trait

8.2 Command Reference

8.2.1 NodeQuery

Runs a single query on the chassis model and returns the matching nodes.

- message **NodeQuery.Request**
- message **NodeQuery.Response**

8.2.2 GetGraph

Fetches the entire graph of the chassis model.

- message **GetGraph.Request**
- message **GetGraph.Response**

8.3 Type Reference

8.3.1 Asi

A node with this trait ...

empty **struct**

8.3.2 Card

A node with this trait represents a card within the chassis.

struct

slot	int slot number of the card this trait represents
------	---

8.3.3 ConfigurationDomain

A node with this traits specifies a configuration object's domain (see ConfigurationId). Used for setting alarm overrides where config object domain is a required parameter.

struct

domainId	string identifier of the configuration domain
displayName	string human friendly name of a domain which can be presented to users

8.3.4 ConfigurationId

A node with this trait represents a configuration object within the chassis.

struct

configId	UUID configuration object id
configLabel	optional string

8.3.5 Connector

A node with this trait represents some kind of physical connector.

struct

physicalLabel	string
---------------	---------------

8.3.6 GInfo

Graph info type. Describes the relations (edges) for all nodes in the chassis model.

struct

parent	optional UUID optional parent of this node
children	set of UUID set of children this node has

8.3.7 GetGraph.Request

empty **struct**

8.3.8 GetGraph.Response

struct

graph **Graph**

8.3.9 Graph

Describes the entire chassis model graph, for every slot.

struct

nodes	map from Slot to map from UUID to Node all nodes, per slot
relations	map from Slot to map from UUID to GInfo all relations, per slot

8.3.10 Input

A node with this trait is some kind of input.

empty **struct**

8.3.11 Ip

A node with this trait ...

empty **struct**

8.3.12 IpAddr

A node with this trait marks is something with an ip address and a port.

struct

ipAddress	string
port	int

8.3.13 IpInterface

A node with this trait is some kind of interface.

empty **struct**

8.3.14 IpVlan

A node with this trait marks a VLAN and contains its number.

struct

vlan **int**

8.3.15 Node

A node in the chassis model.

Each node represents some physical, logical and/or configured entity in the chassis. Nodes have properties and may have traits. A node's traits describe what the node represents. For instance, an ip-input may be represented by a node in the chassis model with the traits Ip and Input set.

struct

id	UUID the id of this node in the chassis model. A node's id is guaranteed to be unique within a slot, but not across the entire chassis.
display_name	string a human readable name for this node
traits	PossibleTraits struct containing traits of node

8.3.16 NodeQuery.Request

Query

8.3.17 NodeQuery.Response

struct

result	map from Slot to list of Node
exception	optional string

8.3.18 Output

Node with this traits is some kind of output

empty **struct**

8.3.19 PathInfo

struct

traits	PossibleTraits
distance_to_root	int

8.3.20 PossibleTraits

struct

card	optional Card optional card trait
	optional Connector

connector	optional connector trait
ipInterface	optional IpInterface optional interface trait
input	optional Input optional input trait
asi	optional Asi optional asi trait
ip	optional Ip optional ip trait
configId	optional ConfigurationId optional configuration object id trait
configDomain	optional ConfigurationDomain optional configuration object alarm domain name trait. For alarm overrides specified for a configuration object that is not fixed to a particular slot.
ipAddr	optional IpAddr optional ip address trait
ipVlan	optional IpVlan optional vlan trait
output	optional Output optional output trait

8.3.21 Query

Chassis model query type. This type is used to “match” against nodes in the chassis model and is used with RPCs to be able to act on a specific set of nodes in the chassis model.

struct

slots	optional set of int which slots to include in the query. If empty, all slots will be included in the query.
traits	set of Traits

8.3.22 Slot

int

8.3.23 Traits

Enum listing all available traits which a node in the chassis model may exhibit

enum

CARD	
CONNECTOR	
IP_INTERFACE	
INPUT	
ASI	
IP	
CONFIG_DOMAIN	

CONFIG_ID	
IP_ADDR	
IP_VLAN	
OUTPUT	

9 configImportExport (1.3)

9.1 Command Reference

9.1.1 ExportChassisConfig

Exports complete chassis configuration. Returned 'configPayload' can be later used to perform chassis configuration import.

- message **ExportChassisConfig.Request**
- message **ExportChassisConfig.Response**

9.1.2 ImportChassisConfig

Imports provided chassis configuration data (which should be 'configPayload' of previously exported chassis configuration) and if successful will try to use it as configuration for all cards.

On success a chassis will be power cycled and then all cards will try to use new configuration. On failure, if a problem was related to provided data an error is reported and nothing happens with the chassis. However, in case of unexpected internal error we will try to recover from it to ensure that the current configuration is intact and this might require to power cycle the chassis after recovering from an error. Failure object provides information whether a chassis was power cycled on error.

- message **ImportChassisConfig.Request**
- message **ImportChassisConfig.Response**

9.1.3 LiveImportChassisConfig

Imports provided chassis configuration data (which should be 'configPayload' of previously exported chassis configuration) and if successful will try to use it as configuration for all cards in seamless way without interrupting their operation. Temporarily disabled due to instability. Use ImportChassisConfig instead.

- message **LiveImportChassisConfig.Request**
- message **LiveImportChassisConfig.Response**

9.2 Type Reference

9.2.1 ExportChassisConfig.Request

empty **struct**

9.2.2 ExportChassisConfig.Response

struct

configPayload **string**

9.2.3 ImportChassisConfig.Request

ImportChassisConfigRequest

9.2.4 ImportChassisConfig.Response

ImportChassisConfigResponse

9.2.5 ImportChassisConfigRequest

struct

configPayload	string
preserveHardwareConfig	bool If set to true then IP interfaces configuration of cards will not be imported and current IP interface config will be preserved.
preserveUsers	bool If set to true then users settings will not be imported and current users settings will be preserved.

9.2.6 ImportChassisConfigResponse

struct

configImportSuccess	bool If true the import went fine and chassis will try to use new configuration for all cards. If false then the chassis should remain using its current configuration.
resetRequested	bool If true then a chassis will go down any moment. This will always happen if import was successful. This can also happen if import failed because of internal error.
errorMessage	optional string In case of failure this might contain a human readable hint to why import failed.

9.2.7 LiveImportChassisConfig.Request

ImportChassisConfigRequest

9.2.8 LiveImportChassisConfig.Response

LiveImportChassisConfigResponse

9.2.9 LiveImportChassisConfigResponse

struct

configImportSuccess	bool If true the import went fine and chassis will try to use new configuration for all cards. If false then the chassis should remain using its current configuration.
errorMessage	optional string In case of failure this might contain a human readable hint to why import failed.

10 configReset (1.0)

10.1 Command Reference

10.1.1 ResetChassisConfig

- message **ResetChassisConfig.Request**
- message **ResetChassisConfig.Response**

10.2 Type Reference

10.2.1 ResetChassisConfig.Request

ResetChassisConfigRequest

10.2.2 ResetChassisConfig.Response

ResetChassisConfigResponse

10.2.3 ResetChassisConfigRequest

Reset configuration of the chassis.

This request will perform a chassis reboot if the request is successful.

struct

resetManagementIpConfig	bool Reset the management port (control port) IP interface configuration on the MMI(s) back to the factory default (192.168.1.100). Otherwise, the current configuration is kept.
resetUserConfig	bool Delete all the current user login configuration. This will remove all current users and passwords, and will be reset back to only a single default user, without any password.
resetOtherConfig	bool Delete all other configuration for all cards (everything but the management IP and the user login configuration). Deletion of config includes for example configured input and output flows, coder services and profiles, biss keys, etc.

10.2.4 ResetChassisConfigResponse

Response from a reset request

struct

success	bool If true, the reset request was successful and the chassis is ready to reboot and then reset the configuration when starting up again. If false, something went wrong, and the the chassis should remain using its current configuration and not reboot.
	optional string

errorMessage

In case of failure this might contain a human readable hint to why reset failed.

11 dns (1.0)

11.1 Command Reference

11.1.1 SetDnsConfig

Set DNS configuration on MMI card. Not supported on X5, use ipGateway:x.xx/ipCardSettings/SetCardSetting instead for DNS configuration.

- message **SetDnsConfig.Request**
- message **SetDnsConfig.Response**

11.1.2 GetDnsConfig

Get DNS configuration on MMI card. Not supported on X5, use ipGateway:x.xx/ipCardSettings/GetCardSetting instead.

- message **GetDnsConfig.Request**
- message **GetDnsConfig.Response**

11.1.3 ResolveName

Request MMI to try to resolve a given name to IP address. This can be useful for testing MMI's DNS settings and MMI's network environment.

- message **ResolveName.Request**
- message **ResolveName.Response**

11.2 Type Reference

11.2.1 DnsConfig

•

struct

nameResolutionEnabled	bool Enables or disables name resolution on MMI card.
-----------------------	---

nameServers	list of string IP addresses of name servers to be used if name resolution is enabled. Maximum 3 servers are allowed, passing more will yield an error response. Passing a string which is not a valid IP address will yield an error response.
-------------	--

11.2.2 GetDnsConfig.Request

empty **struct**

11.2.3 GetDnsConfig.Response

DnsConfig

11.2.4 IPAddress

struct

ipVersion	IpVersion
address	string

11.2.5 IpVersion

enum

v4
v6

11.2.6 ResolveName.Request

struct

name	string
------	---------------

11.2.7 ResolveName.Response

ResolveNameResult

11.2.8 ResolveNameResult

struct

didPerform	bool If 'true' then DNS server was contacted for name resolution. If 'false' then either a given name is not a valid host/domain name or DNS server is not configured (see SetDnsConfig method) or not reachable.
addresses	list of IPAddress If DNS resolution was performed then this contains a list of resolved IP addresses (DNS server might provide more than one) and if the list is empty it means that DNS server couldn't find any IP address for a given name.

11.2.9 SetDnsConfig.Request

DnsConfig

11.2.10 SetDnsConfig.Response

empty **struct**

12 firewallTypes (1.0)

12.1 Type Reference

12.1.1 IPAddress

string

12.1.2 PaginatedRequest

Instructs the server to return only a subset of the data found on the server

@param pageSize number of entries returned per request

@param pageNumber what page of the set to returned, determined by the slice [pageSize*pageNumber

struct

pageSize	int
pageNumber	int

12.1.3 PaginationInfo

struct

totalEntries	int	the total number of entries on the server, before matching
--------------	------------	--

12.1.4 SocketAddress

Tuple with IP address and port

struct

address	IPAddress	IP address, either IPv4 or IPv6
port	int	

12.1.5 UUID

string

13 https (1.1)

13.1 Command Reference

13.1.1 GetCertificate

Returns information about SSL certificate used by an MMI's card's web server for communication over HTTPS protocol.

- message **GetCertificate.Request**
- message **GetCertificate.Response**

13.1.2 InstallCertificate

Installs a certificate and a matching private key provided by a user and restarts MMI card's web server. If certificate/key are invalid then a failure is returned and current configuration is not changed.

- message **InstallCertificate.Request**
- message **InstallCertificate.Response**

13.1.3 RemoveCertificate

Removes current certificate and private key. A card will generate a new self signed certificate and restart MMI card's web server.

- message **RemoveCertificate.Request**
- message **RemoveCertificate.Response**

13.1.4 GetCertificateOverview

Returns basic information about a given PEM encoded SSL certificate.

- message **GetCertificateOverview.Request**
- message **GetCertificateOverview.Response**

13.1.5 GetCertificateDetails

Returns a human readable text representation of all information in a given PEM encoded SSL certificate.

- message **GetCertificateDetails.Request**
- message **GetCertificateDetails.Response**

13.1.6 GeneratePrivateKeyAndCsr

Request a card to generate a private key and CSR using given CSR attributes. Changes state of staging area to WAITING_FOR_CERTIFICATE. If already in WAITING_FOR_CERTIFICATE state returns error.

- message **GeneratePrivateKeyAndCsr.Request**
- message **GeneratePrivateKeyAndCsr.Response**

13.1.7 GetStagingAreaState

- message **GetStagingAreaState.Request**
- message **GetStagingAreaState.Response**

13.1.8 GetCertificateSigningRequest

Tries to download CSR file. Returns PEM encoded CSR.

Following rules are evaluated in the given order, the first matching rule is used:

- if staging area is in WAITING_FOR_CERTIFICATE state then CSR from staging area is returned
- if staging area is EMPTY and active area contains CSR file then CSR from active area is returned
- if active area doesn't contain CSR file then error is returned – this will be the case if webserver uses auto-generated-self-signed certificate or user provided “full pair” of private key and certificate

Which one will be used can be checked by checking value of CertificateSigningRequestDownloadRule.

- message **GetCertificateSigningRequest.Request**
- message **GetCertificateSigningRequest.Response**

13.1.9 GetCertificateSigningRequestDetails

Tries to get CSR file and parse it into human readable information. To determine which CSR file is used (if any) it follows the same rules as GetCertificateSigningRequest. This is a helper function which parses PEM encoded result of GetCertificateSigningRequest. Useful for debugging or simple UIs.

- message **GetCertificateSigningRequestDetails.Request**
- message **GetCertificateSigningRequestDetails.Response**

13.1.10 GetCertificateSigningRequestDownloadRule

Allows to check which CSR file will be downloaded if any.

- message **GetCertificateSigningRequestDownloadRule.Request**
- message **GetCertificateSigningRequestDownloadRule.Response**

13.1.11 DeleteStagingArea

Deletes content of staging area and changes its state to EMPTY.

- message **DeleteStagingArea.Request**
- message **DeleteStagingArea.Response**

13.1.12 InstallCertificateFromSignedCSR

Tries to install a certificate created previously by calling GeneratePrivateKeyAndCsr method and make webserver use the new certificate.

A certificate should match a private key in staging area (if staging area is not EMPTY) or private key currently used by webserver (if staging area is EMPTY). The second case is for renewing certificates.

If a given certificate is not valid format, it doesn't match a private key in staging area or causes any kind of problem then this returns an error.

If successful and staging area is not EMPTY this will:

- will move private key, CSR and certificate to active area
- (!!) will make webserver use the private key and provided certificate
- clean staging area and change its state to EMPTY

If successful and staging area is EMPTY this will:

- make webserver use the new certificate
- message **InstallCertificateFromSignedCSR.Request**
- message **InstallCertificateFromSignedCSR.Response**

13.2 Type Reference

13.2.1 CertificateInfo

•

struct

isUserInstalled	bool If false then the currently used certificate has been automatically generated and self signed by a card. If true then the currently used certificate has been installed by a user using InstallCertificate RPC method.
certificate	string PEM encoded certificate
overview	CertificateOverview Basic information extracted from certificate

13.2.2 CertificateOverview

struct

subject	DistinguishedNameOverview
issuer	DistinguishedNameOverview
validity	CertificateOverview.validity

13.2.3 CertificateOverview.validity

struct

notBefore	bigint
notAfter	bigint

13.2.4 CertificateSigningRequestAttributes

struct

country	string
stateOrProvince	string
locality	string
organization	string
organizationalUnit	string
commonName	string
subjectAlternativeNames	optional list of SubjectAlternativeName

13.2.5 CertificateSigningRequestDownloadRule

enum

NOT_AVAILABLE
 STAGING
 ACTIVE

13.2.6 CertificateSigningRequestInfo

struct

certificateSigningRequest **string**
 PEM encoded CSR, can be provided to signing authority

13.2.7 DeleteStagingArea.Request

struct

slot **int**

13.2.8 DeleteStagingArea.Response

struct

status **bool**

13.2.9 DistinguishedNameOverview

Basic information about 'distinguished name'.

struct

commonName **optional string**
 organization **optional string**
 organizationalUnit **optional string**

13.2.10 GeneratePrivateKeyAndCsr.Request

struct

slot **int**
 attributes **CertificateSigningRequestAttributes**

13.2.11 GeneratePrivateKeyAndCsr.Response

CertificateSigningRequestInfo

13.2.12 GetCertificate.Request**struct**slot int**13.2.13 GetCertificate.Response****CertificateInfo****13.2.14 GetCertificateDetails.Request****struct**certificate string**13.2.15 GetCertificateDetails.Response****struct**readableText string**13.2.16 GetCertificateOverview.Request****struct**certificate string**13.2.17 GetCertificateOverview.Response****struct**overview CertificateOverview**13.2.18 GetCertificateSigningRequest.Request****struct**slot int**13.2.19 GetCertificateSigningRequest.Response****CertificateSigningRequestInfo****13.2.20 GetCertificateSigningRequestDetails.Request****CertificateSigningRequestInfo**

13.2.21 GetCertificateSigningRequestDetails.Response

struct

readableText **string**

13.2.22 GetCertificateSigningRequestDownloadRule.Request

struct

slot **int**

13.2.23 GetCertificateSigningRequestDownloadRule.Response

struct

rule **CertificateSigningRequestDownloadRule**

13.2.24 GetStagingAreaState.Request

struct

slot **int**

13.2.25 GetStagingAreaState.Response

struct

state **StagingAreaState**

13.2.26 InstallCertificate.Request

InstallCertificateRequest

13.2.27 InstallCertificate.Response

empty **struct**

13.2.28 InstallCertificateFromSignedCSR.Request

struct

slot **int**

certificate **string**
PEM encoded certificate

13.2.29 InstallCertificateFromSignedCSR.Response

empty **struct**

13.2.30 InstallCertificateRequest

struct

slot	int
certificate	string PEM encoded certificate
privateKey	string PEM encoded private key matching a public key in the certificate

13.2.31 RemoveCertificate.Request

struct

slot	int
------	------------

13.2.32 RemoveCertificate.Response

empty **struct**

13.2.33 StagingAreaState

enum

EMPTY
WAITING_FOR_CERTIFICATE

13.2.34 SubjectAlternativeName

variant

dns	string
ip	string

14 imageBank (1.1)

14.1 Overview

Changelog

1.1

Added

- added `InsufficientSpace` to `OperationResult` to improve error messages in UI when the underlying storage implementation cannot comply with the `StoreImage` request
- new RPC method `GetSpaceLocal` - allows a sanity check if an upgrade image is not too big

1.0

Introduced the concept of pre-uploading upgrade images for use with upgrades.

Example use-case:

- Upload an upgrade image
- Invoke `StoreImage` RPC with 'url' set to scheme "upload://" and the uploaded filename
- Invoke `ListImages` to get a list of identifiers for uploaded image
- Invoke `multiCardUpgrade/StartUpgradeJob` RPC with 'swUrl' set to "imagebank://identifier"

14.2 Command Reference

14.2.1 StoreImage

- message `StoreImage.Request`
- message `StoreImage.Response`

14.2.2 RemoveImage

- message `RemoveImage.Request`
- message `RemoveImage.Response`

14.2.3 ListImages

- message `ListImages.Request`
- message `ListImages.Response`

14.2.4 GetSpaceLocal

Allows a sanity check if an upgrade image is not too big.

If `available` field of the response is smaller than the size of the upgrade image you will want to upload and store in the image bank then trying to store such image will fail. This allows to avoid possibly long lasting operation which will fail.

- message `GetSpaceLocal.Request`
- message `GetSpaceLocal.Response`

14.3 Type Reference

14.3.1 FileId

string

14.3.2 FileMeta

struct

created	UTCDateTimeISO8601
fileName	string
hash	string

14.3.3 GetSpaceLocal.Request

empty struct

14.3.4 GetSpaceLocal.Response

struct

bytes	optional SpaceLocal
-------	---------------------

14.3.5 ListImages.Request

empty struct

14.3.6 ListImages.Response

struct

images	map from FileId to StoredImageMeta
--------	------------------------------------

14.3.7 OperationResult

enum

Unknown
Ok
Failed
NotFound
InvalidUrl
InvalidImage
InsufficientSpace

14.3.8 RemoveImage.Request

struct

id	FileId
----	--------

14.3.9 RemoveImage.Response

struct

result **OperationResult**

14.3.10 SpaceLocal

struct

used	bigint
size	bigint
available	bigint

14.3.11 StoreImage.Request

struct

url **Url**

14.3.12 StoreImage.Response

struct

result **OperationResult**

14.3.13 StoredImageMeta

struct

file	FileMeta
manifest	string
upgradeParams	optional string

14.3.14 UTCDateTimeISO8601

string

14.3.15 Url

string

14.3.16 UserMeta

map from **string** to **string**

15 license (1.3)

15.1 Command Reference

15.1.1 GetFeaturesInfo

Returns information on supported features known to this card

- message **GetFeaturesInfo.Request**
- message **GetFeaturesInfo.Response**

15.1.2 GetLicense

Returns the status type of license, and – where applicable – the feature set of license installed license on the board. The response is of either type Unknown, Empty or License.

- Unknown: license server is not ready
- Empty: no license installed
- License: the feature set of the installed license
- message **GetLicense.Request**
- message **GetLicense.Response**

15.1.3 GetHardwareId

Returns the unique identifier for the board

- message **GetHardwareId.Request**
- message **GetHardwareId.Response**

15.1.4 InstallLicense

Installs a license on the board. Returns true if the license was successfully installed and false otherwise.

Request parameter license must be a signed, base64 encoded license file

- message **InstallLicense.Request**
- message **InstallLicense.Response**

15.2 Type Reference

15.2.1 BoardFeatures

struct

features **map** from **string** to **FeatureInfo**

15.2.2 DemoPeriod

Time during which a license is valid.

struct

string

startDate	ISO 6801 date/time
endDate	string ISO 6801 date/time
isActive	bool

15.2.3 Empty

empty **struct**

15.2.4 FeatureInfo

struct

id	string
label	string
orderingCode	optional string If not set then license can't be bought, e.g. it might be installed by default.
value	bool

15.2.5 FeatureMap

map from **string** to **InstalledFeatureInfo**

15.2.6 FeaturesInfo

struct

boards **map** from **string** to **BoardFeatures**

15.2.7 GetFeaturesInfo.Request

empty **struct**

15.2.8 GetFeaturesInfo.Response

FeaturesInfo

15.2.9 GetHardwareId.Request

empty **struct**

15.2.10 GetHardwareId.Response

struct

hardwareId **string**

15.2.11 GetLicense.Request

empty **struct**

15.2.12 GetLicense.Response

struct

result **GetLicense.Response.result**

15.2.13 GetLicense.Response.result

variant

unknown	Unknown
empty	Empty
license	License

15.2.14 InstallLicense.Request

struct

license **string**

15.2.15 InstallLicense.Response

struct

success **bool**

15.2.16 InstalledFeatureInfo

Information about a feature installed on a card.

struct

integerValue	optional int If this has a value it means that this license/feature unlocks this many "something".
demo	optional DemoPeriod If this has a value it means that this is time limited demo li- cense/feature.

15.2.17 License

struct

features **FeatureMap**

15.2.18 Unknown

empty **struct**

16 linkRates (1.1)

16.1 Command Reference

16.1.1 GetLinkRates

Returns a list of link rates for each card present in the chassis.

- message **GetLinkRates.Request**
- message **GetLinkRates.Response**

16.2 Type Reference

16.2.1 CardRates

Link rates for a single card.

struct

slot	int
rx	bigint
tx	bigint

16.2.2 GetLinkRates.Request

empty **struct**

16.2.3 GetLinkRates.Response

struct

data	list of CardRates
------	--------------------------

17 maintenanceMmi (1.3)

17.1 Overview

Changelog

1.3

Added

- new RPC method UnblockLogin

17.2 Command Reference

17.2.1 DeleteConfiguration

Delete configuration of a given card and let this card try to log in with default configuration.

If system doesn't have a configuration for a given card or the cards is in a state which doesn't allow deleting configuration then operation fails.

Limitations:

- currently only supported for cards in BLOCKED state and cards which are physically not present
- currently NOT supported for MMI cards or X5
- message **DeleteConfiguration.Request**
- message **DeleteConfiguration.Response**

17.2.2 UnblockLogin

Moves the card out of the BLOCKED state. The card will retry to login with its current configuration and if that fails the card will be brought back to BLOCKED state. Not supported on X5.

- message **UnblockLogin.Request**
- message **UnblockLogin.Response**

17.2.3 RebootSlots

- message **RebootSlots.Request**
- message **RebootSlots.Response**

17.3 Type Reference

17.3.1 DeleteCardConfiguration

struct

slot int

17.3.2 DeleteConfiguration.Request

DeleteConfigurationRequest

17.3.3 DeleteConfiguration.Response

DeleteConfigurationResult

17.3.4 DeleteConfigurationId

variant

card	DeleteCardConfiguration
service	DeleteServiceConfiguration

17.3.5 DeleteConfigurationRequest

struct

id	DeleteConfigurationId
----	-----------------------

17.3.6 DeleteConfigurationResult

struct

success	bool
failureReason	optional string

17.3.7 DeleteServiceConfiguration

struct

name	string
------	--------

17.3.8 Partition

enum

Active
Inactive

17.3.9 RebootSlot

Reboot slot response for a slot

struct

slot	int Slot number
accepted	bool Bool for whether the slot is attempting to reboot
report	optional string Optional string containing a reason or comment on whether the slot is rebooting or not

17.3.10 RebootSlotRequest

struct

slot	int
partition	optional Partition

17.3.11 RebootSlots.Request

RebootSlotsRequest

17.3.12 RebootSlots.Response

RebootSlotsResponse_

17.3.13 RebootSlotsRequest

struct

slots	set of RebootSlotRequest Slots to reboot
-------	---

17.3.14 RebootSlotsResponse_

struct

slots	list of RebootSlot List of response for slots we received a reply from
-------	---

17.3.15 UnblockLogin.Request

UnblockLoginRequest

17.3.16 UnblockLogin.Response

UnblockLoginResponse

17.3.17 UnblockLoginRequest

struct

slot	int
------	-----

17.3.18 UnblockLoginResponse

struct

success	bool
failureReason	optional string

18 multiCardUpgrade (1.4)

18.1 Overview

Changelog

1.4

Changed

- Bumped productInfo from version 1.2 to version 1.3

18.2 Command Reference

18.2.1 StartUpgradeJob

Requests a start of a new upgrade job which will try to upgrade specified cards with a specified software. Response contains information whether an upgrade job has been started or not.

Several conditions must be met so an upgrade job can be started:

- currently only one upgrade job can be run at a time
- all cards must be in a state which allow their upgrade
- message **StartUpgradeJob.Request**
- message **StartUpgradeJob.Response**

18.2.2 GetCurrentJobs

Returns a list of IDs of currently running upgrade jobs.

- message **GetCurrentJobs.Request**
- message **GetCurrentJobs.Response**

18.2.3 GetJobsHistory

Returns a list of IDs of all ever started upgrade jobs which are not running anymore. Contains successful and failed jobs.

- message **GetJobsHistory.Request**
- message **GetJobsHistory.Response**

18.2.4 GetJobStatus

Returns information about specified upgrade jobs.

For each requested job, if a job is currently running or is found in jobs history then information is provided. Otherwise a job is marked as "unknown".

- message **GetJobStatus.Request**
- message **GetJobStatus.Response**

18.2.5 CheckUpgradeImageCompatibility

Given a product manifest extracted from an upgrade image, checks which cards in chassis can use an upgrade image and under what conditions.

- message **CheckUpgradeImageCompatibility.Request**
- message **CheckUpgradeImageCompatibility.Response**

18.2.6 GetCardsSlots

Returns a list of all slots of all cards in chassis.

- message **GetCardsSlots.Request**
- message **GetCardsSlots.Response**

18.2.7 GetCardsStatus

Returns upgrade status of specified cards.

- message **GetCardsStatus.Request**
- message **GetCardsStatus.Response**

18.2.8 GetCardsProductInfo

Returns product information of specified cards.

- message **GetCardsProductInfo.Request**
- message **GetCardsProductInfo.Response**

18.2.9 CyclePower

Not supported on X5. Use the card reboot functionality instead.

- message **CyclePower.Request**
- message **CyclePower.Response**

18.3 Type Reference

18.3.1 CardList

struct

slots	list of int
-------	-------------

18.3.2 CardProductInfoResponse

struct

cards	map from int to productInfo.ProductInfo map of card's slot to product info
cardMissingSlots	list of int

18.3.3 CardStatus

struct

slot	int
upgradeState	CardUpgradeState

supportedProtocols **set of string**

18.3.4 CardStatusDump

Dump of status data received from a card. This is mostly for human reading.

struct

upgradeId	string
upgradeState	string
partition	int
taskRecord	list of string
latestDebug	list of string

18.3.5 CardStatusResponse

struct

cards	list of CardStatus
cardMissingSlots	list of int

18.3.6 CardStep

A card during an upgrade will go via certain steps. Upgrade pipeline defines these steps that a card can reach and information about steps' state is provided for each card. Initial state of each step is NOT_REACHED.

enum

UPGRADE_INITIATED	
SOFTWARE_INSTALLED	
REBOOT_INITIATED	

18.3.7 CardStepRecord

struct

stepType	CardStep
state	CardStepState
msg	string optional human readable information regarding current state of a card step
timeStamp	optional bigint time of the latest change of step's state (seconds since beginning of unix epoch)

18.3.8 CardStepState

enum

NOT_REACHED	
-------------	--

SUCCESS	
FAILURE	
SKIPPED	

18.3.9 CardUpgradeState

enum

UNKNOWN	
NOT_RUNNING	
RUNNING	

18.3.10 CardUpgradeStatus

Information about execution of upgrade on a single card.

struct

slot	int Slot of the card being upgraded
swUrl	string URL of SW package file that a card should use to upgrade itself
steps	list of CardStepRecord The steps of the upgrade and their state
cardDump	CardStatusDump

18.3.11 CheckUpgradeImageCompatibility.Request

UpgradeImageCompatibilityRequest

18.3.12 CheckUpgradeImageCompatibility.Response

UpgradeImageCompatibilityResult

18.3.13 CyclePower.Request

CyclePowerRequest

18.3.14 CyclePower.Response

CyclePowerResponse

18.3.15 CyclePowerRequest

empty **struct**

18.3.16 CyclePowerResponseempty **struct****18.3.17 GetCardsProductInfo.Request****CardList****18.3.18 GetCardsProductInfo.Response****CardProductInfoResponse****18.3.19 GetCardsSlots.Request**empty **struct****18.3.20 GetCardsSlots.Response****CardList****18.3.21 GetCardsStatus.Request****CardList****18.3.22 GetCardsStatus.Response****CardStatusResponse****18.3.23 GetCurrentJobs.Request**empty **struct****18.3.24 GetCurrentJobs.Response****JobList****18.3.25 GetJobStatus.Request****JobList****18.3.26 GetJobStatus.Response****GetJobStatusResponse****18.3.27 GetJobStatusResponse****struct**

jobs	list of JobStatus
unknown	list of UUID

18.3.28 GetJobsHistory.Request

empty **struct**

18.3.29 GetJobsHistory.Response

JobList

18.3.30 HardwareCompatibility

enum

CANT_USE	Hardware mismatch. Upgrade image can't be used to upgrade a card.
COMPATIBLE	Card's hardware matches hardware target of an upgrade image. Upgrade image can be used.

18.3.31 JobFinishedResult

struct

success	bool
errorInfo	optional string

18.3.32 JobList

struct

jobs	list of UUID
------	---------------------

18.3.33 JobState

enum

RUNNING	job is still running
FINISHED_UNEXPECTED_FAILURE	job finished, job was interrupted unexpectedly during one of steps, job is in undefined state
FINISHED_PRELIMINARY_FAILURE	job finished running, job detected a stop condition in preliminary phase, nothing was done to any card
FINISHED_EXECUTED	job finished running, all necessary job's steps were executed, user can check each card's upgrade results

18.3.34 JobStatus

Information about a multi card upgrade job. Contains current state and history of execution.

struct

jobId	UUID
executorSlot	int slot of a card executing this job

jobConfig	UpgradeJobConfig
state	JobState machine readable current state of a job
currentStep	string human readable name of current step of execution of a job
stepRecord	list of JobStepRecord human readable history of job's steps
timeStarted	bigint time when job was started (seconds since beginning of unix epoch)
timeFinished	optional bigint if has a value then it's time when job was finished (seconds since beginning of unix epoch)
cards	list of CardUpgradeStatus

18.3.35 JobStepRecord

Information about execution of a step of upgrade process.

struct

name	string human readable short description of a step
startTime	bigint timestamp (seconds since beginning of unix epoch)
reports	list of string human readable additional information reported by a step
result	JobStepResult when step finished this will be SUCCESS or FAILURE, if step got into undefined or hanging state or poof-dissapeared then this will be UNKNOWN

18.3.36 JobStepResult

enum

UNKNOWN
SUCCESS
FAILURE

18.3.37 SlotConfig

struct

slot	int
reboot	bool

18.3.38 SoftwareCompatibility

enum

CANT_USE	Software mismatch. Upgrade image can't be used to upgrade a card.
SAME_PRODUCT	Card's software-id matches functionality in an upgrade image (the same product). Upgrade image can be used.
PRODUCT_CHANGE_COMPATIBLE	Software mismatch - different but compatible products. Upgrade image can be used to change functionality of a card (change product) to functionality in upgrade image. Card will be operational after software installation and reboot. This is possible only when allowProductChange is enabled when starting an upgrade job.
PRODUCT_CHANGE_INCOMPATIBLE	Software mismatch - different and INCOMPATIBLE. Upgrade image can be used to change functionality of a card (change product) to functionality in upgrade image. HOWEVER, card WILL NOT be operational after software installation and reboot - it will be required to manually delete existing configuration to let card start with default config. This is possible only when allowProductChange is enabled when starting an upgrade job.

18.3.39 StartUpgradeJob.Request

UpgradeJobConfig

18.3.40 StartUpgradeJob.Response

StartUpgradeJobResult

18.3.41 StartUpgradeJobResult

struct

started	bool
rejectionReason	optional string
jobId	optional UUID

18.3.42 UpgradeImageCompatibility

Whether an upgrade image can be used to upgrade a card. An upgrade image can be used if both software and hardware parts allow usage of an image.

struct

software	SoftwareCompatibility
hardware	HardwareCompatibility
supportUpgradeWithoutReboot	UpgradeWithoutRebootSupport

18.3.43 UpgradeImageCompatibilityRequest

struct

manifest	string
	Base64 encoded content of product.manifest file extracted from an upgrade image.

slots	optional list of int If not defined then compatibility check will be performed for all cards in chassis. If defined it will be performed only for specified cards.
-------	--

18.3.44 UpgradeImageCompatibilityResult

struct

cards	map from int to UpgradeImageCompatibility Result of compatibility check of cards against an upgrade image. Key is card's slot. If compatibility check was limited to specified cards and some requested cards' slots are not in this map it means that these cards are not present in chassis or the system couldn't acquire information from these cards required to determine compatibility.
-------	--

18.3.45 UpgradeJobConfig

struct

slots	list of SlotConfig List of slots of cards which will be upgraded
swUrl	string URL pointing to software package file which should be used to upgrade cards
allowProductChange	bool bababa

18.3.46 UpgradeWithoutRebootSupport

enum

NOT_SUPPORTED	Card does not support upgrade without reboot, i.e. it must reboot as part of upgrade procedure.
SUPPORTED	Card supports upgrade without reboot. Upgrade procedure can install upgrade without rebooting the card.

19 pagination (1.0)

19.1 Type Reference

19.1.1 DataPageInfo

struct

startIndex	int
hasReachedEnd	bool

19.1.2 DataSetInfo

struct

token	optional UUID Used to inform about changes in dataset. If backend can detect changes in dataset then this should have a value. If this value changes it means that dataset has changed.
size	optional int Total size of full dataset. Should have a value if backend can know the size of full dataset.

19.1.3 DataSetSlice

struct

startIndex	int
elementCount	int

19.1.4 PaginatedDataInfo

struct

datasetInfo	DataSetInfo
pageInfo	DataPageInfo

19.1.5 PaginatedQueryResult

variant

ok	PaginatedDataInfo Requested data can be addressed and you get it now.
processing	PaginatedQueryResult.processing Requested data can be addressed but you can't get it now, system started to retrieve the data so you will be able to get it soon.
noContent	PaginatedQueryResult.noContent Requested data can be addressed but you can't get it now, system is in a state in which it's currently impossible to retrieve data.

19.1.6 PaginatedQueryResult.noContent

empty **struct**

19.1.7 PaginatedQueryResult.processing

empty **struct**

20 passwordPolicy (1.0)

20.1 Command Reference

20.1.1 GetPasswordPolicies

- message **GetPasswordPolicies.Request**
- message **GetPasswordPolicies.Response**

20.1.2 SetPasswordPolicies

- message **SetPasswordPolicies.Request**
- message **SetPasswordPolicies.Response**

20.1.3 DeletePasswordPolicies

- message **DeletePasswordPolicies.Request**
- message **DeletePasswordPolicies.Response**

20.2 Type Reference

20.2.1 DeletePasswordPolicies.Request

struct

policyIds **set of UUID**

20.2.2 DeletePasswordPolicies.Response

empty **struct**

20.2.3 GetPasswordPolicies.Request

empty **struct**

20.2.4 GetPasswordPolicies.Response

struct

data **map from UUID to PasswordPolicy**

20.2.5 PasswordContent

struct

minimumLength **int**
If zero then empty passwords are allowed. Must not be negative number.

mustContainNonAlphaNumeric **bool**
Must contain a character which is not a letter or a digit.

mustContainUpperCase	bool
mustContainLowerCase	bool
mustContainNumber	bool

20.2.6 PasswordExpiration

struct

period	int How long until a newly created password becomes invalid and needs to be changed. In seconds. Can't be negative.
blockUser	bool If set to true a user will be blocked when password expires (effectively user must change password before it expires or contact admin to get unblocked and then be able to change password).

20.2.7 PasswordPolicy

struct

name	string Human friendly name or description of a policy.
content	optional PasswordContent If set it will force new passwords to meet given criteria. This will be enforced when users change passwords or when new users are created.
expiration	optional PasswordExpiration If set this then system will set expiration date for each password under this policy. Password's expiration date can be refreshed by user if he/she changes a password.

20.2.8 SetPasswordPolicies.Request

struct

data	map from UUID to PasswordPolicy
------	--

20.2.9 SetPasswordPolicies.Response

empty **struct**

21 productInfo (1.3)

21.1 Overview

Changelog

1.3

Changed

- Updated to use `articleNumber` and `moduleHardwareID` as `hardwareInfo`. Replaces previous `hardwareInfo`

21.2 Type Reference

21.2.1 HardwareInfo

struct

<code>articleNumber</code>	<code>string</code>
<code>moduleHardwareID</code>	<code>string</code>

21.2.2 ProductInfo

Provides basic information about what software bundle is running and what is the hardware/board.

struct

<code>activeSoftwareBundle</code>	<code>SoftwareBundleInfo</code>
<code>inactiveSoftwareBundle</code>	<code>optional SoftwareBundleInfo</code>
<code>hardware</code>	<code>HardwareInfo</code>

21.2.3 SoftwareBundleInfo

struct

<code>id</code>	<code>string</code>
<code>name</code>	<code>string</code>
<code>displayName</code>	<code>string</code>
<code>ver</code>	<code>string</code>
<code>supportUpgradeWithoutReboot</code>	<code>bool</code>
<code>partitionId</code>	<code>optional bootPartition.PartitionId</code>
<code>active</code>	<code>bool</code>
<code>hasBooted</code>	<code>bool</code>
<code>lastInstallTime</code>	<code>string</code>
<code>lastBootTime</code>	<code>string</code>

22 prometheus (1.2)

22.1 Overview

Changelog

1.1

Added

- new RPC method `GetPrometheusEndpointNames`

1.2

Added

- `BasicAuth` and `APIToken` authentication options

22.2 Command Reference

22.2.1 SetPrometheusSettings

Changes current Prometheus settings.

- message `SetPrometheusSettings.Request`
- message `SetPrometheusSettings.Response`

22.2.2 GetPrometheusSettings

Returns current Prometheus settings.

- message `GetPrometheusSettings.Request`
- message `GetPrometheusSettings.Response`

22.2.3 GetPrometheusEndpointNames

Returns available Prometheus endpoints.

Allows to construct a complete list of scrape URLs for a given AppearTV unit. Scrape URLs have a form like: `https://[UNIT-CONTROL-INTERFACE-IP]/prometheus/[ENDPOINT-NAME]/metrics`

- message `GetPrometheusEndpointNames.Request`
- message `GetPrometheusEndpointNames.Response`

22.3 Type Reference

22.3.1 APITokenAuth

APIToken/APIKey.

@param token String that must be present in HTTP requests under the "Authorization" header with Token must have between 1 and 2048 characters. Token must match with the one set on the promethe

struct

token `string`

22.3.2 AuthenticationVariant

variant

jwt_token	JWTTokenAuth
basic	BasicAuth
api_token	APITokenAuth

22.3.3 BasicAuth

```
HTTP Basic Auth (https://en.wikipedia.org/wiki/Basic_access_authentication).
@param username Username to be set.
@param password Password to be set.
Username and password must have between 1 and 255 characters.
Username cannot contain a colon (:).
```

struct

username	string
password	string

22.3.4 GetPrometheusEndpointNames.Request

empty **struct**

22.3.5 GetPrometheusEndpointNames.Response

PrometheusEndpointNames

22.3.6 GetPrometheusSettings.Request

empty **struct**

22.3.7 GetPrometheusSettings.Response

PrometheusSettings

22.3.8 JWTTokenAuth

```
JWT token method (BeginSession, RefreshSession).
```

empty **struct**

22.3.9 PrometheusEndpointNames

struct

endpointNames	set of string
---------------	----------------------

22.3.10 PrometheusSettings

struct

enable	bool
authentication	AuthenticationVariant

22.3.11 SetPrometheusSettings.Request

PrometheusSettings

22.3.12 SetPrometheusSettings.Response

empty struct

23 remoteConfigExport (1.0)

23.1 Command Reference

23.1.1 SetRemoteConfigExportSchedule

- message `SetRemoteConfigExportSchedule.Request`
- message `SetRemoteConfigExportSchedule.Response`

23.1.2 GetRemoteConfigExportSchedule

- message `GetRemoteConfigExportSchedule.Request`
- message `GetRemoteConfigExportSchedule.Response`

23.1.3 SetRemoteConfigExportInstance

- message `SetRemoteConfigExportInstance.Request`
- message `SetRemoteConfigExportInstance.Response`

23.1.4 GetAllRemoteConfigExportInstances

- message `GetAllRemoteConfigExportInstances.Request`
- message `GetAllRemoteConfigExportInstances.Response`

23.1.5 DeleteRemoteConfigExportInstance

- message `DeleteRemoteConfigExportInstance.Request`
- message `DeleteRemoteConfigExportInstance.Response`

23.1.6 GetAllRemoteConfigExportStatus

- message `GetAllRemoteConfigExportStatus.Request`
- message `GetAllRemoteConfigExportStatus.Response`

23.1.7 PerformRemoteConfigExport

- message `PerformRemoteConfigExport.Request`
- message `PerformRemoteConfigExport.Response`

23.2 Type Reference

23.2.1 DeleteRemoteConfigExportInstance.Request

struct

id `RemoteConfigExportInstanceId`

23.2.2 DeleteRemoteConfigExportInstance.Response

empty struct

23.2.3 GetAllRemoteConfigExportInstances.Request

empty **struct**

23.2.4 GetAllRemoteConfigExportInstances.Response

struct

instances **Instances**

23.2.5 GetAllRemoteConfigExportStatus.Request

empty **struct**

23.2.6 GetAllRemoteConfigExportStatus.Response

struct

status **Statuses**

23.2.7 GetRemoteConfigExportSchedule.Request

empty **struct**

23.2.8 GetRemoteConfigExportSchedule.Response

struct

schedule **Schedule**

23.2.9 Instance

variant

sftp **SFTPRemoteConfigExport**

23.2.10 Instances

map from **RemoteConfigExportInstanceId** to **Instance**

23.2.11 LastRun

WallClockTimePoint

23.2.12 PerformRemoteConfigExport.Request

struct

requested **set** of **RemoteConfigExportInstanceId**

23.2.13 PerformRemoteConfigExport.Response

struct

accepted **set** of **RemoteConfigExportInstanceld**

23.2.14 RemoteConfigExportInstance

struct

instance **Instance**

23.2.15 RemoteConfigExportInstanceld

UUID

23.2.16 RemoteConfigExportLastRun

struct

lastRun **LastRun**

23.2.17 RemoteConfigExportScheduleParameters

struct

params **ScheduleParameters**

23.2.18 RemoteConfigExportStatus

struct

status **Statuses**

23.2.19 SFTPRemoteConfigExport

struct

enabled	bool
name	string
hostname	string
port	int
username	string
password	string
destinationDirectory	string
baseFilename	string

23.2.20 Schedule

struct

parameters	ScheduleParameters
lastRun	optional LastRun

23.2.21 ScheduleParameters

struct

enable	bool
timeToRun	WallClockTimePoint

23.2.22 SetRemoteConfigExportInstance.Request

struct

id	RemoteConfigExportInstanceId
data	Instance

23.2.23 SetRemoteConfigExportInstance.Response

empty **struct**

23.2.24 SetRemoteConfigExportSchedule.Request

struct

schedule	ScheduleParameters
----------	---------------------------

23.2.25 SetRemoteConfigExportSchedule.Response

empty **struct**

23.2.26 Status

struct

time	WallClockTimePoint
success	bool
errorMessage	optional string

23.2.27 Statuses

map from **RemoteConfigExportInstanceId** to **Status**

23.2.28 WallClockTimePoint

string

24 security (1.0)

24.1 Command Reference

24.1.1 GetCardSecurityStatus

- message **GetCardSecurityStatus.Request**
- message **GetCardSecurityStatus.Response**

24.2 Type Reference

24.2.1 CardSecurityStatus

struct

passwordInstalled **bool**

24.2.2 GetCardSecurityStatus.Request

struct

slot **int**

24.2.3 GetCardSecurityStatus.Response

struct

status **optional CardSecurityStatus**

25 services (1.4)

25.1 Command Reference

25.1.1 GetRoutedNodes

Returns service information for the queried list of service node IDs if these nodes have been routed.

- message **GetRoutedNodes.Request**
- message **GetRoutedNodes.Response**

25.1.2 GetInputServices

Returns service information for the queried list of service node IDs. By default returns only services which are currently present on inputs. Can be provided a query object which can change default behavior.

- message **GetInputServices.Request**
- message **GetInputServices.Response**

25.1.3 GetInputServicesPage

- message **GetInputServicesPage.Request**
- message **GetInputServicesPage.Response**

25.1.4 GetOutputServices

Returns service information for output services

- message **GetOutputServices.Request**
- message **GetOutputServices.Response**

25.1.5 GetServiceDependencies

Returns service dependency information

- message **GetServiceDependencies.Request**
- message **GetServiceDependencies.Response**

25.2 Type Reference

25.2.1 ChildSourceInfo

struct

name	string
nodeType	string
body	string
owner	string
isPresent	bool

25.2.2 FlowSinkInfo

struct

name	string
label	string
nodeType	string
body	string
sources	map from UUID to FlowSourceServiceInfo
owner	string

25.2.3 FlowSourceInfo

struct

name	string
label	string
nodeType	string
body	string
owner	string
isPresent	bool
sources	map from UUID to ChildSourceInfo

25.2.4 FlowSourcePage

struct

result	pagination.PaginatedQueryResult
data	map from UUID to FlowSourceInfo

25.2.5 FlowSourceServiceInfo

ChildSourceInfo

25.2.6 FlowSourceStorage

map from UUID to FlowSourceInfo

25.2.7 GetInputServices.Request

GetInputServicesRequest

25.2.8 GetInputServices.Response

struct

data	FlowSourceStorage
------	-------------------

25.2.9 GetInputServicesPage.Request

GetInputServicesPageRequest

25.2.10 GetInputServicesPage.Response

FlowSourcePage

25.2.11 GetInputServicesPageRequest

struct

slice	pagination.DataSetSlice
query	optional InputServiceQuery

25.2.12 GetInputServicesRequest

struct

query	optional InputServiceQuery
-------	----------------------------

25.2.13 GetOutputServices.Request

empty struct

25.2.14 GetOutputServices.Response

struct

data	map from UUID to FlowSinkInfo
------	-------------------------------

25.2.15 GetRoutedNodes.Request

struct

uuids	list of UUID
-------	--------------

25.2.16 GetRoutedNodes.Response

struct

data	FlowSourceStorage
------	-------------------

25.2.17 GetServiceDependencies.Request

empty struct

25.2.18 GetServiceDependencies.Response

struct

data **list** of **ServiceDependencyInfo**

25.2.19 InputServicePresenceQuery

enum

ALL

ONLY_PRESENT

ONLY_MISSING

25.2.20 InputServiceQuery

struct

servicePresence **InputServicePresenceQuery**

25.2.21 ServiceDependencyInfo

struct

source **UUID**

dependency **UUID**

26 snmp (1.2)

26.1 Overview

Tips

- RPC methods related to SNMPv3 have typically V3 at the end of their names (unless they deal with concepts which only exist in SNMPv3, then suffix might be skipped in the name, e.g. GetEngineId)
- RPC methods related to SNMPv2c don't have anything added at the end of their names

Changelog

1.2

Added

- user specified engine ID (SNMPv3)
 - added new optional field engineId to SnmpConfigV3, in case of nullopt it will be reset to a default value
- INFORMs support (SNMPv2c and SNMPv3)
 - added informDestinations fields to SnmpConfigV3 and 'SnmpTrapDestinationList' - now both these structs have two lists, one for trap destinations and one for inform destinations
 - new RPC method GetInformStatistics which provides diagnostic data about sending INFORMs to each INFORM destination
- new RPC method TestSnmpDestination

Changed

- renamed fields with lists of trap destinations in SNMPv2c and SNMPv3 config
 - data field in SnmpTrapDestinationList which refers to SNMPv2 Trap destination is renamed to trapDestinations
 - destinations in SnmpConfigV3 which refers to SNMPv3 Trap destination is renamed to trapDestinations

1.1

Added

- SNMPv3 support
 - new RPC methods: GetSnmpConfigV3, UpdateSnmpConfigV3, GetEngineId

26.2 Command Reference

26.2.1 GetTrapDestinations

Get SNMPv2c settings.

- message **GetTrapDestinations.Request**
- message **GetTrapDestinations.Response**

26.2.2 UpdateTrapDestinations

Update SNMPv2c settings.

- message **UpdateTrapDestinations.Request**

- message **UpdateTrapDestinations.Response**

26.2.3 GetSnmpConfigV3

Get SNMPv3 settings and verification status for each setting (for each destination and each security name).

- message **GetSnmpConfigV3.Request**
- message **GetSnmpConfigV3.Response**

26.2.4 UpdateSnmpConfigV3

Update SNMPv3 settings.

Response contains result of basic validation and if new settings has been saved. However, we can encounter runtime problems when trying to put settings to use. To verify if settings are corret and don't cause problems use please RPC call GetSnmpConfigV3 which contains runtime status.

- message **UpdateSnmpConfigV3.Request**
- message **UpdateSnmpConfigV3.Response**

26.2.5 GetEngineId

Get SNMPv3 security EngineID.

- message **GetEngineId.Request**
- message **GetEngineId.Response**

26.2.6 GetInformStatistics

Provides diagnostic data about sending SNMP INFORMs to all destinations.

- message **GetInformStatistics.Request**
- message **GetInformStatistics.Response**

26.2.7 TestSnmpDestination

Sends a test alarm to the first configured destination of given type which matches a given IP address. Fails if can't find a matching destination.

- message **TestSnmpDestination.Request**
- message **TestSnmpDestination.Response**

26.3 Type Reference

26.3.1 AuthProtocol

enum

HmacMD5	
HmacSHA1	
HmacSHA224	
HmacSHA256	
HmacSHA384	
HmacSHA512	

26.3.2 EngineIdV3WithStatus

struct

engineId	string
status	Result Configuration verification status. If we encountered problems with setting engineId failReason string with details is returned.

26.3.3 GetEngineId.Request

empty **struct**

26.3.4 GetEngineId.Response

struct

engineId	string
----------	--------

26.3.5 GetInformStatistics.Request

empty **struct**

26.3.6 GetInformStatistics.Response

InformStatusData

26.3.7 GetSnmpConfigV3.Request

empty **struct**

26.3.8 GetSnmpConfigV3.Response

SnmpConfigV3WithStatus

26.3.9 GetTrapDestinations.Request

empty **struct**

26.3.10 GetTrapDestinations.Response

SnmpTrapDestinationList

26.3.11 InformDestinationV2WithStatistics

struct

config	SnmpInformDestination
statistics	InformStatus

26.3.12 InformDestinationV3WithStatistics

struct

config	SnmpInformDestinationV3
statistics	InformStatus

26.3.13 InformDestinationV3WithStatus

struct

config	SnmpInformDestinationV3 Single SNMPV3 configuration object
status	Result Configuration verification status. If we are not able to establish a session using provided data failReason string with details is returned.

26.3.14 InformStatus

struct

attempted	int number of alarm events we tried to send as INFORMs
delivered	int number of alarm events successfully sent with confirmed reception
unsent	int number of alarm events we didn't manage to send, e.g. host not reachable
undelivered	int number of alarm events we managed to send but we never received reception confirmation (even after retries)
sec_errors	int number of alarm events we managed to send and got some response but we were unable to read the response
retries	int a metric telling about network and/or SNMP receiver configuration problems - total number of times we tried to send an INFORM again because we didn't get reception confirmation, ideally should be zero

26.3.15 InformStatusData

struct

V2Statistics	list of InformDestinationV2WithStatistics
V3Statistics	list of InformDestinationV3WithStatistics

26.3.16 PrivConfig

struct

privProtocol	PrivProtocol
privPassPhrase	string

26.3.17 PrivProtocol

enum

AES
DES

26.3.18 Result

variant

ok	Result.ok
failReason	string

26.3.19 Result.ok

empty **struct**

26.3.20 SecurityConfig

struct

authProtocol	AuthProtocol
authPassPhrase	string
priv	optional PrivConfig

26.3.21 SecurityNameMap

User-based Security Model (USM) settings for SNMPv3.
Collection of security settings per securityName (user).
<https://tools.ietf.org/html/rfc3414>

map from **string** to **optional SecurityConfig**

26.3.22 SecurityNameV3WithStatus

struct

securityName	string
security	optional SecurityConfig
status	Result Configuration verification status. If we encountered problems with security settings failReason string with details is returned.

26.3.23 SnmpConfigV3

struct

trapDestinations	list of SnmpTrapDestinationV3
informDestinations	list of SnmpInformDestinationV3
securityNames	SecurityNameMap
engineId	optional string

26.3.24 SnmpConfigV3VerificationResponse

struct

overallAccepted	bool If true configuration passed verification stage and has been saved into database. Previous configuration has been overwritten.
trapDestinations	list of TrapDestinationV3WithStatus Verification status about each SNMPV3 trap destination provided in request.
informDestinations	list of InformDestinationV3WithStatus Verification status about each SNMPV3 inform destination provided in request.
securityNames	list of SecurityNameV3WithStatus Verification status about each securityName provided in request.
engineId	optional EngineIdV3WithStatus

26.3.25 SnmpConfigV3WithStatus

struct

trapDestinations	list of TrapDestinationV3WithStatus
informDestinations	list of InformDestinationV3WithStatus
securityNames	list of SecurityNameV3WithStatus
engineId	optional EngineIdV3WithStatus

26.3.26 SnmpDestinationSelector

struct

destinationType	SnmpDestinationType
ipAddress	string

26.3.27 SnmpDestinationType

enum

TrapV2	
TrapV3	
InformV2	

InformV3	
----------	--

26.3.28 SnmpInformDestination

struct; extending **SnmpTrapDestination**

ackTimeout	int
retries	int

26.3.29 SnmpInformDestinationV3

struct; extending **SnmpTrapDestinationV3**

ackTimeout	int
retries	int
remoteEngineId	optional string

26.3.30 SnmpTrapDestination

struct

ipAddress	string
community	string
isEnabled	bool

26.3.31 SnmpTrapDestinationList

struct

trapDestinations	list of SnmpTrapDestination
informDestinations	list of SnmpInformDestination

26.3.32 SnmpTrapDestinationV3

struct

ipAddress	string
isEnabled	bool
securityName	string

Specifies which security settings should be used for this destination. Must be a security name used in SecurityNameMap.

26.3.33 TestSnmpDestination.Request

SnmpDestinationSelector

26.3.34 TestSnmpDestination.Response

Result

26.3.35 TrapDestinationV3WithStatus

struct

config	SnmpTrapDestinationV3 Single SNMPV3 configuration object
status	Result Configuration verification status. If we are not able to establish a session using provided data failReason string with details is returned.

26.3.36 UpdateSnmpConfigV3.Request

SnmpConfigV3

26.3.37 UpdateSnmpConfigV3.Response

SnmpConfigV3VerificationResponse

26.3.38 UpdateTrapDestinations.Request

SnmpTrapDestinationList

26.3.39 UpdateTrapDestinations.Response

empty **struct**

27 syslog (1.1)

27.1 Overview

Changelog

1.1

Added

- new RPC method `TestSyslogDestination`

27.2 Command Reference

27.2.1 `GetRemoteSyslogDestinations`

- message `GetRemoteSyslogDestinations.Request`
- message `GetRemoteSyslogDestinations.Response`

27.2.2 `SetRemoteSyslogDestinations`

This is for configuring sending of alarms to remote syslog servers.

- message `SetRemoteSyslogDestinations.Request`
- message `SetRemoteSyslogDestinations.Response`

27.2.3 `TestSyslogDestination`

Sends a test alarm to the first configured destination which matches a given IP address. Fails if can't find a matching destination.

- message `TestSyslogDestination.Request`
- message `TestSyslogDestination.Response`

27.3 Type Reference

27.3.1 `GetRemoteSyslogDestinations.Request`

empty `struct`

27.3.2 `GetRemoteSyslogDestinations.Response`

`RemoteSyslogDestinationList`

27.3.3 `RemoteSyslogDestination`

`struct`

<code>address</code>	<code>string</code> IP address or host/domain name of Syslog server
<code>isEnabled</code>	<code>bool</code> Alarms are logged only if this is set to true. Allows pausing logging to Syslog server without removing configuration.

27.3.4 RemoteSyslogDestinationList

struct

data **list** of **RemoteSyslogDestination**

27.3.5 Result

variant

ok	Result.ok
failReason	string

27.3.6 Result.ok

empty **struct**

27.3.7 SetRemoteSyslogDestinations.Request

RemoteSyslogDestinationList

27.3.8 SetRemoteSyslogDestinations.Response

empty **struct**

27.3.9 SyslogDestinationSelector

struct

ipAddress **string**

27.3.10 TestSyslogDestination.Request

SyslogDestinationSelector

27.3.11 TestSyslogDestination.Response

Result

28 uptime (1.0)

28.1 Command Reference

28.1.1 GetLoginUptime

Returns login uptime of all cards.

- message **GetLoginUptime.Request**
- message **GetLoginUptime.Response**

28.1.2 GetSystemUptime

Returns system uptime of all cards.

- message **GetSystemUptime.Request**
- message **GetSystemUptime.Response**

28.2 Type Reference

28.2.1 GetLoginUptime.Request

empty **struct**

28.2.2 GetLoginUptime.Response

LoginUptimeResponse

28.2.3 GetSystemUptime.Request

empty **struct**

28.2.4 GetSystemUptime.Response

SystemUptimeResponse

28.2.5 LoginUptimeResponse

struct

loginUptime

map from **int** to **uptime**
map of card's slot to login uptime.

28.2.6 SystemUptimeResponse

struct

systemUptime

map from **int** to **uptime**
map of card's slot to system uptime.

28.2.7 uptime

The uptime given in seconds (s).

bigint

29 userPolicy (1.0)

29.1 Command Reference

29.1.1 GetGlobalUserPolicy

- message **GetGlobalUserPolicy.Request**
- message **GetGlobalUserPolicy.Response**

29.1.2 SetGlobalUserPolicy

- message **SetGlobalUserPolicy.Request**
- message **SetGlobalUserPolicy.Response**

29.2 Type Reference

29.2.1 GetGlobalUserPolicy.Request

empty **struct**

29.2.2 GetGlobalUserPolicy.Response

UserPolicy

29.2.3 GlobalUserPolicy

struct

policy **UserPolicy**

29.2.4 SetGlobalUserPolicy.Request

UserPolicy

29.2.5 SetGlobalUserPolicy.Response

empty **struct**

29.2.6 UserPolicy

struct

minUsernameLength **int**

30 websocket (1.0)

30.1 Command Reference

30.1.1 UseToken

Make the current websocket connection use a given access token (obtained from BeginSession/RefreshSession) for all further RPC requests. This method is typically used when you obtain an access token via HTTP and want to use it also for websocket connection. If BeginSession is called via websocket calling this method is not needed. Only works with websocket. Fails with http.

- message **UseToken.Request**
- message **UseToken.Response**

30.2 Type Reference

30.2.1 UseToken.Request

struct

accessToken

string

JWT access token (without the 'Bearer' part) previously received from BeginSession / RefreshSession

30.2.2 UseToken.Response

empty **struct**