

# Content index

<b>Request and Response</b> .....	11
Request types.....	12
Xmlrpc.....	12
Jsonrpc.....	12
Simple.....	12
Response types.....	12
Xmlrpc.....	13
Json.....	13
Jsonrpc.....	13
Xml.....	13
Simple.....	13
Debug.....	14
Test.....	14
<b>Errors</b> .....	15
Error codes.....	16
SUCCESS ( code: 1000 ).....	16
SUCCESS_PENDING ( code: 1001 ).....	16
UNKNOWN_COMMAND ( code: 2000 ).....	16
COMMAND_SYNTAX_ERROR ( code: 2001 ).....	16
COMMAND_USE_ERROR ( code: 2002 ).....	16
PARAM_MISSING ( code: 2003 ).....	16
PARAM_VALUE_RANGE ( code: 2004 ).....	16
PARAM_VALUE_SYNTAX ( code: 2005 ).....	16
DELETED_COMMAND ( code: 2100 ).....	17
UNIMPLEMENTED_COMMAND ( code: 2101 ).....	17
BILLING_FAILURE ( code: 2104 ).....	17
BILLING_NOT_ALLOWED ( code: 2105 ).....	17
AUTH_ERROR_USER ( code: 2200 ).....	17
AUTH_ERROR_OBJECT ( code: 2201 ).....	17
OBJECT_EXISTS ( code: 2302 ).....	17
OBJECT_NOT_EXISTS ( code: 2303 ).....	18
OBJECT_STATUS_PROHIBITS_OP ( code: 2304 ).....	18
OBJECT_ASSOCIATION_PROHIBITS_OP ( code: 2305 ).....	19
PARAMETER_VALUE_POLICY_ERROR ( code: 2306 ).....	19
UNIMPLEMENTED_OBJECT_SERVICE ( code: 2307 ).....	19
COMMAND_FAILED ( code: 2400 ).....	20
COMMAND_FAILED_FATAL ( code: 2500 ).....	20
COMMAND_TIMEOUT ( code: 2501 ).....	20
COMMAND_CONNECTIONSLIMIT ( code: 2502 ).....	20
COMMAND_MAINTENANCE_MODE ( code: 2503 ).....	21
COMMAND_PENDING_PREVIOUS_EXECUTION ( code: 3101 ).....	21
COMMAND_FTP_MAX_ACCOUNTS ( code: 3601 ).....	21
COMMAND_EMAIL_REDIRECTION_MAX_ACCOUNTS ( code: 3602 ).....	21
COMMAND_EMAIL_DISTRIBUTIONLIST_MAX ( code: 3603 ).....	21
COMMAND_SERVER_KVM_GET_ERROR ( code: 3604 ).....	21
COMMAND_DEDICATED_SERVER_OS_ERROR ( code: 3605 ).....	21
COMMAND_LICENSE_MAX_HOSTINGS ( code: 3606 ).....	21
COMMAND_BILLING_PRODUCT_ERROR ( code: 3607 ).....	22
COMMAND_BILLING_CREDIT ( code: 3608 ).....	22
COMMAND_EMAIL_MAX_HOSTINGS ( code: 3609 ).....	22
COMMAND_DOMAIN_FORBIDDEN_CONTACT_UPDATE ( code: 3610 ).....	22
COMMAND_DDBB_SQLSERVER_NOT_ENOUGH_SPACE ( code: 3612 ).....	22
COMMAND_EMAIL_DISTRIBUTION_LIST_EXISTS ( code: 3613 ).....	22
COMMAND_SMS_NOT_REMAINING_MESSAGES ( code: 3614 ).....	23
COMMAND_MYSQL_MAX_DATABASES ( code: 3615 ).....	23
COMMAND_DOMAIN_IP_NOT_RESOLVING_DNS ( code: 3616 ).....	23
COMMAND_EMAIL_NOT_ENABLED ( code: 3617 ).....	23
COMMAND_REGENERATE_PASSWORD_SEND_METHOD ( code: 3618 ).....	23

COMMAND_EMAIL_ACCOUNT_SET_REPEATED_PASSWORD ( code: 3619 ) .....	23
COMMAND_BILLING_DOMAIN_FORBIDDEN_PREDEFINED_CONTACTS ( code: 3620 ) .....	23
COMMAND_DOMAIN_TRANSFER_ALREADY_APPROVED ( code: 3630 ) .....	23
COMMAND_DOMAIN_CERT_ALREADY_ASSOCIATED ( code: 3631 ) .....	24
COMMAND_BILLING_PRODUCT_TEMPORARILY_DISABLED ( code: 3621 ) .....	24
COMMAND_NFS_VOLUME_DOES_NOT_EXIST ( code: 3640 ) .....	24
COMMAND_PARAM_IS_NOT_A_VALID_CERTIFICATE ( code: 3641 ) .....	24
COMMAND_DOMAIN_SERVER_ZONE_NOT_FOUND ( code: 3645 ) .....	24
<b>Commands</b> .....	<b>25</b>
AdditionalStorage commands .....	26
AdditionalStorage_Repository_GetAll .....	26
AdditionalStorage_Repository_SetPassword .....	26
AdditionalStorage_Repository_Create .....	26
AdditionalStorage_Repository_Delete .....	27
Billing commands .....	28
Billing_NoRenewable_GetAll .....	28
Billing_NoRenewable_SetRenewable .....	28
Billing_Renew_Hosting .....	28
Billing_Renew_Multihosting .....	28
Billing_Renew_Domain .....	29
Billing_Renew_Vps .....	29
Billing_Renew_Sms .....	29
Billing_Price_Hosting .....	30
Billing_Price_Multihosting .....	30
Billing_Price_Vps_Managed_Linux .....	31
Billing_Price_Vps_Managed_Windows .....	31
Billing_Price_Vps_Unmanaged_Linux .....	31
Billing_Price_Vps_Unmanaged_Windows .....	32
Billing_Price_Domain .....	32
Billing_Price_Sms .....	33
Billing_Period_Hosting .....	33
Billing_Period_Multihosting .....	34
Billing_Period_Domain .....	34
Billing_Period_Vps .....	34
Billing_Buy_Hosting .....	34
Billing_Buy_Multihosting .....	35
Billing_Buy_Vps_Managed_Linux .....	35
Billing_Buy_Vps_Managed_Windows .....	36
Billing_Buy_Vps_Unmanaged_Linux .....	36
Billing_Buy_Vps_Unmanaged_Windows .....	36
Billing_Buy_Domain .....	37
Billing_Buy_DomainUsingContactsIds .....	37
Billing_Buy_Sms .....	38
Billing_Transfer_GetRequirements .....	38
Billing_Transfer_Domain .....	39
Billing_Autorenew_GetAll .....	39
Billing_Autorenew_SetNoRenewable .....	40
Billing_Autorenew_SetOff .....	40
Billing_Autorenew_SetOn .....	40
Billing_GetConfigurableVpsOptions .....	41
Billing_Help_DomainExtraData .....	41
Dedicated commands .....	42
Dedicated_Services_GetAll .....	42
Dedicated_Services_ChangeStatus .....	42
Dedicated_GetTransferValue .....	42
Dedicated_Control_GetApacheProcesses .....	43
Dedicated_Control_GetSystemStatus .....	43
Dedicated_Control_GetMysqlProcesses .....	43
Dedicated_Control_GetWebserverStatus .....	44

Dedicated_Control_GetCpuLoadByHosting	44
Dedicated_Control_GetSystemLoad	44
Dedicated_Control_KillMysqlProcess	44
Dedicated_ForVps_Vps_Create_Unmanaged	45
Dedicated_ForVps_Vps_Create_Managed	45
Dedicated_ForVps_Vps_GetAll	46
Dedicated_ForVps_Vps_GetTypes	46
Dedicated_ForVps_Vps_Activate	46
Dedicated_ForVps_Vps_Delete	47
Dedicated_ForVps_Vps_Deactivate	47
Dedicated_ForVps_Vps_Update	47
Dedicated_ForVps_GetTicketStatus	48
Dedicated_ForVps_GetServerInfo	48
Dedicated_Kvm_Get	48
Dedicated_Kvm_Unmount	49
Dedicated_Kvm_Mount	49
Dedicated_GetLicenses	49
Dedicated_Monitorization_Get	50
Dedicated_Monitorization_Activate	50
Dedicated_Monitorization_Deactivate	50
Dedicated_GetServerInfo	50
Dedicated_HostingSetup_Get	51
Dedicated_HostingSetup_GetAll	51
Dedicated_HostingSetup_GetOptions	51
Dedicated_HostingSetup_GetDhSetups	52
Dedicated_HostingSetup_Create	52
Dedicated_HostingSetup_GetCustomSetups	52
Dedicated_HostingSetup_Delete	52
Dedicated_HostingSetup_Update	53
Dedicated_GetMonitorizedPorts	53
Dedicated_PowerStatus_Restart	53
Dedicated_GetDataTransferInfo	54
Dedicated_Nfs_Get	54
Dedicated_MailReport_Set	54
Dedicated_MailReport_Get	54
Dedicated_MailReport_SetOff	55
Dedicated_GetUsedSpace	55
Dedicated_Hosting_GetAll	55
Dedicated_Hosting_GetSpace	56
Dedicated_Hosting_Activate	56
Dedicated_Hosting_Create	56
Dedicated_Hosting_Delete	57
Dedicated_Hosting_Deactivate	57
Dedicated_Hosting_GetSetupName	57
Dedicated_Hosting_ChangeSetup	58
Dedicated_GetBandwidthConsumption	58
Dedicated_GetSpaceAndTransferReportPerHosting	58
Domain commands	60
Domain_Redirection_Set301	60
Domain_Redirection_Get	60
Domain_Redirection_GetAll	60
Domain_Redirection_SetFrame	60
Domain_Redirection_SetStd	61
Domain_Redirection_Delete	61
Domain_Redirection_SetFrameSimple	62
Domain_CheckForTransfer	62
Domain_GetRenovationDate	62
Domain_GetRegistrationDate	62
Domain_GetAuthcode	63

Domain_Contacts_Set	63
Domain_Contacts_Get	63
Domain_Contacts_GetRegistrantVerificationInfo	64
Domain_Contacts_GetAll	64
Domain_Contacts_SetAll	64
Domain_WhoisPrivacy_Set	64
Domain_WhoisPrivacy_Get	65
Domain_UrlShortener_GetAll	65
Domain_UrlShortener_Activate	65
Domain_UrlShortener_Create	66
Domain_UrlShortener_Delete	66
Domain_UrlShortener_Deactivate	66
Domain_PanelIndependentAccess_SetPassword	66
Domain_GetInfoDomain	67
Domain_Zone_AddTypeSRV	67
Domain_Zone_GetTypeURL	67
Domain_Zone_DeleteTypeAname	68
Domain_Zone_AddTypeFrame	68
Domain_Zone_DeleteTypeTXT	68
Domain_Zone_GetAll	68
Domain_Zone_UpdateTypeAname	69
Domain_Zone_UpdateTypeA	69
Domain_Zone_DeleteTypeSRV	69
Domain_Zone_AddTypeMXS	70
Domain_Zone_DeleteTypeSPF	70
Domain_Zone_AddTypeA	70
Domain_Zone_GetTypeMX	71
Domain_Zone_AddTypeURL301	71
Domain_Zone_GetTypeAname	71
Domain_Zone_DeleteTypeAAAA	72
Domain_Zone_GetTypeFrame	72
Domain_Zone_AddTypeAname	72
Domain_Zone_UpdateTypeAAAA	73
Domain_Zone_UrlShortener_GetAll	73
Domain_Zone_UrlShortener_Activate	73
Domain_Zone_UrlShortener_Create	73
Domain_Zone_UrlShortener_Delete	74
Domain_Zone_UrlShortener_Deactivate	74
Domain_Zone_AddTypeMX	74
Domain_Zone_DeleteTypeFrame	75
Domain_Zone_AddTypeTXT	75
Domain_Zone_DeleteAll	75
Domain_Zone_DeleteTypeCname	75
Domain_Zone_DeleteTypeMX	76
Domain_Zone_GetTypeSPF	76
Domain_Zone_GetTypeA	76
Domain_Zone_Delete	77
Domain_Zone_AddTypeMXDetailedEmail	77
Domain_Zone_DeleteTypeA	77
Domain_Zone_DeleteTypeURL301	77
Domain_Zone_GetTypeCname	78
Domain_Zone_GetTypeTXT	78
Domain_Zone_DeleteTypeURL	78
Domain_Zone_AddTypeCname	79
Domain_Zone_GetTypeURL301	79
Domain_Zone_AddTypeURL	79
Domain_Zone_AddTypeAAAA	80
Domain_Zone_GetTypeSRV	80
Domain_Zone_GetTypeAAAA	80

Domain_Dnss_Set .....	81
Domain_Dnss_Get .....	81
Domain_Status_Get .....	81
Domain_Status_IsLocked .....	81
Domain_Status_Unlock .....	82
Domain_Status_Lock .....	82
Domain_Parking_Set .....	82
Domain_Parking_Get .....	82
Domain_Transfer_GetStatus .....	83
Domain_Transfer_GetAuthcode .....	83
Domain_Transfer_SetAuthcode .....	83
Domain_Transfer_Restart .....	84
Domain_Transfer_Cancel .....	84
Domain_GetExpirationDate .....	84
Domain_Test .....	84
Domain_CheckForRegister .....	85
Domain_NameServer_Modify .....	85
Domain_NameServer_GetAll .....	85
Domain_NameServer_Register .....	86
Domain_NameServer_Delete .....	86
DomainVerification commands .....	87
DomainVerification_GetPending .....	87
Hosting commands .....	88
Hosting_Dsn_Access_GetAll .....	88
Hosting_Dsn_Access_Create .....	88
Hosting_Dsn_Access_Delete .....	88
Hosting_Dsn_Sqlserver_GetAll .....	88
Hosting_Dsn_Sqlserver_Create .....	89
Hosting_Dsn_Sqlserver_Delete .....	89
Hosting_Dsn_Mysql_GetAll .....	89
Hosting_Dsn_Mysql_Create .....	90
Hosting_Dsn_Mysql_Delete .....	90
Hosting_Php_DeactivatePhpExecutionOnHtml .....	90
Hosting_Php_ActivatePhpExecutionOnHtml .....	91
Hosting_Php_SetVersion .....	91
Hosting_Php_OverwritePhpIniWithDefaults .....	91
Hosting_Php_PhpExecutionOnHtmlIsActive .....	91
Hosting_Php_OverwritePhpIni .....	92
Hosting_IsActive .....	92
Hosting_GetMailInfo .....	92
Hosting_Ftp_Get .....	92
Hosting_Ftp_Account_GetAll .....	93
Hosting_Ftp_Account_SetPassword .....	93
Hosting_Ftp_Account_CmdStatus_GetCreate .....	93
Hosting_Ftp_Account_CmdStatus_ListAll .....	94
Hosting_Ftp_Account_CmdStatus_GetSetPassword .....	94
Hosting_Ftp_Account_CmdStatus_GetDelete .....	94
Hosting_Ftp_Account_Create .....	95
Hosting_Ftp_Account_Delete .....	95
Hosting_Ftp_Activate .....	95
Hosting_Ftp_Deactivate .....	96
Hosting_Ftp_GetAccountsLimit .....	96
Hosting_Ssh_Get .....	96
Hosting_Ssh_Activate .....	96
Hosting_Ssh_Deactivate .....	97
Hosting_GetInfo .....	97
Hosting_UrlShortener_GetAll .....	97
Hosting_UrlShortener_Activate .....	97
Hosting_UrlShortener_Create .....	98

Hosting_UrlShortener_Delete .....	98
Hosting_UrlShortener_Deactivate .....	98
Hosting_TlsMail_Get .....	98
Hosting_TlsMail_Activate .....	99
Hosting_TlsMail_Deactivate .....	99
Hosting_Https_Get .....	99
Hosting_Https_Activate .....	99
Hosting_Https_Deactivate .....	100
Hosting_PanellIndependentAccess_SetPassword .....	100
Hosting_Email_UsedSpace_Get .....	100
Hosting_Email_UsedSpace_GetPerAlias .....	100
Hosting_Email_Antivirus_UpdateAttachedFileFilter .....	101
Hosting_Email_Antivirus_GetConfig .....	101
Hosting_Email_Antivirus_GetAttachedFilesFilters .....	101
Hosting_Email_Antivirus_SetConfig .....	102
Hosting_Email_Redirection_ActivateCatchAll .....	102
Hosting_Email_Redirection_Get .....	102
Hosting_Email_Redirection_GetAll .....	103
Hosting_Email_Redirection_DeactivateCatchAll .....	103
Hosting_Email_Redirection_Create .....	103
Hosting_Email_Redirection_Delete .....	103
Hosting_Email_Redirection_Update .....	104
Hosting_Email_Antispam_GetWhitelist .....	104
Hosting_Email_Antispam_AddDomainToWhitelist .....	104
Hosting_Email_Antispam_AddDomainToBlacklist .....	105
Hosting_Email_Antispam_RemoveDomainFromWhitelist .....	105
Hosting_Email_Antispam_GetScore .....	105
Hosting_Email_Antispam_GetBlacklist .....	105
Hosting_Email_Antispam_GetConfig .....	106
Hosting_Email_Antispam_Activate .....	106
Hosting_Email_Antispam_SetScore .....	106
Hosting_Email_Antispam_RemoveDomainFromBlacklist .....	107
Hosting_Email_Antispam_Deactivate .....	107
Hosting_Email_Antispam_SetConfig .....	107
Hosting_Email_Dkim_Get .....	107
Hosting_Email_Dkim_IsActive .....	108
Hosting_Email_Dkim_Activate .....	108
Hosting_Email_Dkim_Deactivate .....	108
Hosting_Email_AutomaticDeletion_DeleteRule .....	108
Hosting_Email_AutomaticDeletion_CreateRule .....	109
Hosting_Email_Autoresponse_Get .....	109
Hosting_Email_Autoresponse_Activate .....	110
Hosting_Email_Autoresponse_Create .....	110
Hosting_Email_Autoresponse_Delete .....	110
Hosting_Email_Autoresponse_Deactivate .....	111
Hosting_Email_Account_Empty .....	111
Hosting_Email_Account_Get .....	111
Hosting_Email_Account_GetAll .....	112
Hosting_Email_Account_SetPassword .....	112
Hosting_Email_Account_RemoveCopyTo .....	112
Hosting_Email_Account_Activate .....	113
Hosting_Email_Account_Create .....	113
Hosting_Email_Account_SetCopyTo .....	113
Hosting_Email_Account_Delete .....	114
Hosting_Email_Account_GetCopyTo .....	114
Hosting_Email_Account_Deactivate .....	114
Hosting_Email_Filter_SetGeolocalization .....	114
Hosting_Email_Filter_GetAll .....	115
Hosting_Email_Filter_GetGeolocalization .....	115

Hosting_Email_Filter_CreateForSpam .....	115
Hosting_Email_Filter_Create .....	116
Hosting_Email_Filter_Delete .....	116
Hosting_Certificate_Uninstall .....	116
Hosting_Certificate_DeactivateLetsEncrypt .....	117
Hosting_Certificate_Install .....	117
Hosting_Certificate_ActivateLetsEncrypt .....	117
Hosting_Ddbb_Access_GetAllMdb .....	117
Hosting_Ddbb_GetAll .....	118
Hosting_Ddbb_MariaDb_GetAll .....	118
Hosting_Ddbb_MariaDb_SetPassword .....	118
Hosting_Ddbb_MariaDb_Create .....	119
Hosting_Ddbb_MariaDb_Delete .....	119
Hosting_Ddbb_Sqlserver_GetAll .....	119
Hosting_Ddbb_Sqlserver_SetPassword .....	120
Hosting_Ddbb_Sqlserver_SetTransactionsSize .....	120
Hosting_Ddbb_Sqlserver_SetDataSize .....	120
Hosting_Ddbb_Sqlserver_GetGlobalSpaceInfo .....	121
Hosting_Ddbb_Sqlserver_Create .....	121
Hosting_Ddbb_Sqlserver_Delete .....	121
Hosting_Ddbb_Sqlserver_GetSpaceInfo .....	122
Hosting_Ddbb_MongoDb_GetAll .....	122
Hosting_Ddbb_MongoDb_SetPassword .....	122
Hosting_Ddbb_MongoDb_Delete .....	123
Hosting_Ddbb_Postgre_GetAll .....	123
Hosting_Ddbb_Postgre_SetPassword .....	123
Hosting_Ddbb_Postgre_Create .....	124
Hosting_Ddbb_Postgre_Delete .....	124
Hosting_Ddbb_Mysql_CreateWithIpAcl .....	124
Hosting_Ddbb_Mysql_GetAll .....	125
Hosting_Ddbb_Mysql_SetPassword .....	125
Hosting_Ddbb_Mysql_GetAclConfiguration .....	125
Hosting_Ddbb_Mysql_GetInfo .....	125
Hosting_Ddbb_Mysql_SetHostAcl .....	126
Hosting_Ddbb_Mysql_BackupTask_Set .....	126
Hosting_Ddbb_Mysql_BackupTask_GetAll .....	126
Hosting_Ddbb_Mysql_BackupTask_Delete .....	127
Hosting_Ddbb_Mysql_SetIpAcl .....	127
Hosting_Ddbb_Mysql_Create .....	127
Hosting_Ddbb_Mysql_Delete .....	128
Hosting_Ddbb_Mysql_Backup_Get .....	128
Hosting_Ddbb_Mysql_Backup_Restore .....	128
Hosting_Import_List .....	129
Hosting_Import_ApplyForEmailAccountImap .....	129
Hosting_Import_ListAll .....	129
Hosting_Import_ApplyForDdbb .....	130
Hosting_Import_ApplyForEmailAccountPop .....	130
Hosting_Import_ApplyForFtp .....	131
Hosting_GetDaysWithBackups .....	132
Hosting_Test .....	132
Hosting_Subdomain_GetAll .....	132
Hosting_Subdomain_Create .....	133
Hosting_Subdomain_Delete .....	133
Hosting_Backup_GetStatus .....	133
Hosting_Backup_Database_GetTables .....	133
Hosting_Backup_Restore_Database_Tables .....	134
Hosting_Backup_Restore_Resource .....	134
Hosting_Backup_Restore_EmailAccounts .....	135
Hosting_Backup_Restore_Database .....	135

Hosting_Backup_GetDatabases .....	135
Hosting_Backup_GetEmailAccounts .....	136
Hosting_Backup_GetResources .....	136
Hosting_Backup_Send_Database_Tables .....	136
Hosting_Backup_Send_Resource .....	137
Hosting_Backup_Send_Database .....	137
Housing commands .....	138
Housing_GetTransferValue .....	138
Housing_Kvm_Get .....	138
Housing_Kvm_Unmount .....	138
Housing_Kvm_Mount .....	138
Housing_Monitorization_Get .....	139
Housing_Monitorization_Activate .....	139
Housing_Monitorization_Deactivate .....	139
Housing_GetServerInfo .....	140
Housing_PowerStatus_Restart .....	140
Housing_GetDataTransferInfo .....	140
Housing_GetBandwidthConsumption .....	140
Import commands .....	142
Import_GetStatus .....	142
Multihosting commands .....	143
Multihosting_GetLicenses .....	143
Multihosting_GetEmailAccountsNumber .....	143
Multihosting_GetInfo .....	143
Multihosting_HostingSetup_Get .....	143
Multihosting_HostingSetup_GetAll .....	144
Multihosting_HostingSetup_GetOptions .....	144
Multihosting_HostingSetup_Create .....	144
Multihosting_HostingSetup_Delete .....	145
Multihosting_HostingSetup_Update .....	145
Multihosting_MailReport_Set .....	145
Multihosting_MailReport_Get .....	146
Multihosting_MailReport_SetOff .....	146
Multihosting_GetUsedTransfer .....	146
Multihosting_GetUsedSpace .....	147
Multihosting_Hosting_GetAll .....	147
Multihosting_Hosting_GetSpace .....	147
Multihosting_Hosting_Activate .....	148
Multihosting_Hosting_Create .....	148
Multihosting_Hosting_Delete .....	148
Multihosting_Hosting_Deactivate .....	149
Multihosting_Hosting_ChangeSetup .....	149
Payment commands .....	150
Payment_Method_GetAll .....	150
Payment_Credit_GetTotal .....	150
Payment_Credit_List .....	150
Payment_Credit_GetDetailed .....	150
Services commands .....	152
Services_GetDomains .....	152
Services_GetManagedDedicatedServers .....	152
Services_GetCloudServers .....	152
Services_GetSms .....	152
Services_GetDedicatedServers .....	153
Services_GetManagedCloudServers .....	153
Services_GetUnmanagedDedicatedServers .....	153
Services_GetDomainTransfers .....	154
Services_GetHousingServers .....	154
Services_GetHostings .....	154
Services_GetUnmanagedCloudServers .....	155



Services_GetUnmanagedVps .....	155
Services_GetAllManagedServersNames .....	155
Services_GetManagedVps .....	155
Services_GetMultihostings .....	156
Services_GetVps .....	156
<b>Sms commands .....</b>	<b>157</b>
Sms_SetUp .....	157
Sms_Phonebook_GetAllContacts .....	157
Sms_Phonebook_CreateContact .....	157
Sms_Phonebook_UpdateContact .....	158
Sms_Phonebook_DeleteContact .....	158
Sms_GetSender .....	158
Sms_Programmed_Cancel .....	158
Sms_PanelIndependentAccess_SetPassword .....	159
Sms_History_GetProgrammed .....	159
Sms_History_GetBulkSent .....	159
Sms_History_GetSent .....	160
Sms_GetCredit .....	160
Sms_Send_Bulk_Limited_Unicode .....	160
Sms_Send_Bulk_Limited_Gsm7 .....	161
Sms_Send_Bulk_Long_Unicode .....	161
Sms_Send_Bulk_Long_Gsm7 .....	161
Sms_DeliveryList_Get .....	162
Sms_DeliveryList_GetAll .....	162
Sms_DeliveryList_GetPhones .....	162
Sms_DeliveryList_Create .....	163
Sms_DeliveryList_Delete .....	163
Sms_DeliveryList_Update .....	163
<b>System commands .....</b>	<b>165</b>
System_GetRequestTypes .....	165
System_TestSimulation .....	165
System_MethodHelp .....	165
System_GetApiVersion .....	165
System_MethodSignature .....	165
System_ListMethods .....	166
System_GetResponseTypes .....	166
<b>User commands .....</b>	<b>167</b>
User_GetServices .....	167
User_Secondary_GetAll .....	167
User_Secondary_Create .....	167
User_Secondary_Delete .....	167
User_Secondary_Update .....	168
User_GetInvoicedAmount .....	168
User_GetInfo .....	169
User_GetNotifications .....	169
User_GetLastLoginDate .....	169
User_Default_GetAdminContact .....	169
User_Default_GetDnss .....	170
User_Default_GetTechContact .....	170
User_Data_GetValidLegalForms .....	170
<b>Vps commands .....</b>	<b>171</b>
Vps_Services_GetAll .....	171
Vps_Services_ChangeStatus .....	171
Vps_Control_GetApacheProcesses .....	171
Vps_Control_GetSystemStatus .....	172
Vps_Control_GetMysqlProcesses .....	172
Vps_Control_GetCpuLoadByHosting .....	172
Vps_Control_GetSystemLoad .....	173
Vps_Control_KillMysqlProcess .....	173

Vps_GetAllHostingsFullReport .....	173
Vps_GetLicenses .....	173
Vps_Monitorization_Get .....	174
Vps_Monitorization_Activate .....	174
Vps_Monitorization_Deactivate .....	174
Vps_GetServerInfo .....	175
Vps_HostingSetup_Get .....	175
Vps_HostingSetup_GetAll .....	175
Vps_HostingSetup_GetOptions .....	176
Vps_HostingSetup_Create .....	176
Vps_HostingSetup_Delete .....	176
Vps_HostingSetup_Update .....	176
Vps_GetMonitorizedPorts .....	177
Vps_PowerStatus_Unpause .....	177
Vps_PowerStatus_TurnOff .....	177
Vps_PowerStatus_Get .....	178
Vps_PowerStatus_TurnOn .....	178
Vps_PowerStatus_Reactivate .....	178
Vps_PowerStatus_Suspend .....	178
Vps_PowerStatus_ElectricRestart .....	179
Vps_PowerStatus_Pause .....	179
Vps_PowerStatus_Restart .....	179
Vps_GetDataTransferInfo .....	179
Vps_Nfs_Get .....	180
Vps_MailReport_Set .....	180
Vps_MailReport_Get .....	180
Vps_MailReport_SetOff .....	181
Vps_GetUsedSpace .....	181
Vps_Hosting_GetAll .....	181
Vps_Hosting_Activate .....	182
Vps_Hosting_Create .....	182
Vps_Hosting_Delete .....	182
Vps_Hosting_Deactivate .....	182
Vps_Hosting_GetSetupName .....	183
Vps_Hosting_ChangeSetup .....	183
Vps_GetBandwidthConsumption .....	183
Vps_GetSpaceAndTransferReportPerHosting .....	184

# Request and Response

## Request types

### Xmlrpc

Standard Xml Remote Procedure Call request.

Special HTTP request that sends XML encoded arguments and expects a XML encoded response.

Characteristics and requirements:

- Request data sent through HTTP POST query .
- Content-type indicated as "application/xml" on HTTP headers.
- User login data must be sent at "Authorization: Basic" on HTTP headers .
- Parameters must arrive ordered, as described in method signature.
- Only used with Xmlrpc response type.
- Simulation mode only available on web server at <https://dinahosting.com/api/> .

### Jsonrpc

Json Remote Procedure Call request.

Special HTTP request that sends JSON encoded arguments and expects a JSON encoded response.

Characteristics and requirements:

- Request data sent through HTTP POST query .
- Content-type indicated as "application/json" on HTTP headers.
- User login data must be sent at "Authorization: Basic" on HTTP headers .
- Parameters can be sent in any order. Keys are case sensitive.
- Only used with Jsonrpc response type.
- Simulation mode only available on web server at <https://dinahosting.com/api/>.

### Simple

Simple HTTP REQUEST .

Standard HTTP request that sends key=value pairs in a URL encoded request.

Characteristics and requirements:

- Request data sent through a standard HTTP GET or POST query .
- User login authentication data might be sent at "Authorization: Basic" on HTTP headers or in request parameters AUTH\_USER and AUTH\_PWD.
- Parameters can be sent in any order. Keys are case sensitive.
- Can be used with any responseType.
- Can force client-side activation of simulation mode. Just add a boolean parameter SIMULATE=true to the request.

Example: [https://dinahosting.com/special/api.php?AUTH\\_USER=yourUser&AUTH\\_PWD=yourPass&command=System\\_GetRequestTypes&responseType=Xml&SIMULATE=false](https://dinahosting.com/special/api.php?AUTH_USER=yourUser&AUTH_PWD=yourPass&command=System_GetRequestTypes&responseType=Xml&SIMULATE=false)

## Response types

### Xmlrpc

Extensible markup language Remote procedure call response.

Standard XML response built with Zend XmlRpc server.  
Easy to parse with Zend XmlRpc client, but heavier communication.

Recommended only for XmlRpc requests.

### Json

JavaScript Object Notation response.

Lightweight, multi-plataform, easy parseable response type.

Used with standard (Simple) HTTP requests.

### Jsonrpc

JavaScript Object Notation Remote procedure call response.

Same as standard Json .

Only for JsonRpc requests.

### Xml

Extensible markup language response.

Multi-plataform language. Lighter than our XmlRpc.

Used with standard (Simple) HTTP requests.

### Simple

Simple key value plain text response.

Lightest response, but it isn't standard.  
Parsing may be troublesome if response data is complex.  
Useful for testing purposes.

Used with standard (Simple) HTTP requests.

## Debug

Debugging response.

Used with standard (Simple) HTTP requests.

## Test

Testing response. Returns valid PHP code.  
Harder to parse than Json or XML.

Used with standard (Simple) HTTP requests.

# Errors

## Error codes

### SUCCESS ( code: 1000 )

Command execution ended OK and returned data if any.

Error message: *Success.*

Can be thrown by commands: All existing commands might return this error.

### SUCCESS\_PENDING ( code: 1001 )

Command execution ended OK but additional background action is pending.

Error message: *Success. Action is pending.*

Can be thrown by commands: None

### UNKNOWN\_COMMAND ( code: 2000 )

The user requested a command that doesn't exist.

Error message: *Unknown command.*

Can be thrown by commands: Thrown if requested command doesn't exist.

### COMMAND\_SYNTAX\_ERROR ( code: 2001 )

There is at least one error at command arguments.

A parameter may be missing or has got a bad format.

See additional information at errors structure.

Error message: *Command syntax error.*

Can be thrown by commands: All existing commands might return this error.

### COMMAND\_USE\_ERROR ( code: 2002 )

The user doesn't indicate a command on request, requested an invalid response type, or tried to use a command for something that isn't meant to.

Error message: *Command use error.*

Can be thrown by commands: All existing commands might return this error.

### PARAM\_MISSING ( code: 2003 )

The command requires a parameter that is missing on the request.

Error message: *Required param "%s" is missing.*

Can be thrown by commands: All existing commands might return this error.

### PARAM\_VALUE\_RANGE ( code: 2004 )

A command parameter value length isn't valid. The value length might be too long or too short.

Error message: *Param "%s" length is not valid.*

Can be thrown by commands: All existing commands might return this error.

### PARAM\_VALUE\_SYNTAX ( code: 2005 )

A command parameter value format isn't valid.



Error message: *Param "%s" value syntax is not valid.*  
 Can be thrown by commands: All existing commands might return this error.

## DELETED\_COMMAND ( code: 2100 )

The user requested a command that existed but it isn't available anymore, as it was deprecated and deleted.

Error message: *Command was deprecated and itsn't available anymore.*  
 Can be thrown by commands: None

## UNIMPLEMENTED\_COMMAND ( code: 2101 )

The user requested a command that exists but it isn't implemented, or tried to do simulation or MethodHelp over a command that doesn't implements it.

Error message: *Action %s currently cant be performed on this command.*  
 Can be thrown by commands: All existing commands might return this error.

## BILLING\_FAILURE ( code: 2104 )

The billing process failed. Check you've got enough cash and retry operation.

Error message: *Billing error.*  
 Can be thrown by commands: Billing\_Renew\_Hosting, Billing\_Renew\_Multihosting, Billing\_Renew\_Domain, Billing\_Renew\_Sms, Billing\_Buy\_Hosting, Billing\_Buy\_Multihosting, Billing\_Buy\_Domain, Billing\_Buy\_DomainUsingContactsIds, Billing\_Buy\_Sms, Billing\_Transfer\_Domain, Billing\_Autorenew\_SetOn

## BILLING\_NOT\_ALLOWED ( code: 2105 )

User is not authorized to perform any billing operations.

Error message: *Billing operations are not allowed.*  
 Can be thrown by commands: None

## AUTH\_ERROR\_USER ( code: 2200 )

The user authentication failed. Check that username and password sent on request are valid. Keep in mind that some request types require sending auth info on HTTP headers.

Error message: *Authentication error.*  
 Can be thrown by commands: All existing commands might return this error.

## AUTH\_ERROR\_OBJECT ( code: 2201 )

The authenticated user is not authorized to execute this action on the required object (domain, hosting, ...). This could happen if you aren't the owner of the object, or the object internal status doesn't allow this action (it isn't active ...).

Error message: *Authorization error.*  
 Can be thrown by commands: All existing commands might return this error.

## OBJECT\_EXISTS ( code: 2302 )

The request is valid but it can be only completed if the requested object (zone, hosting, email account...) doesn't exists.

Error message: *Param "%s" value already exists.*

Can be thrown by commands: Dedicated\_HostingSetup\_Create, Dedicated\_Hosting\_Create, Vps\_HostingSetup\_Create, Vps\_Hosting\_Create, Domain\_Zone\_AddTypeSRV, Domain\_Zone\_AddTypeFrame, Domain\_Zone\_AddTypeMXS, Domain\_Zone\_AddTypeA, Domain\_Zone\_AddTypeURL301, Domain\_Zone\_AddTypeAname, Domain\_Zone\_UrlShortener\_Activate, Domain\_Zone\_AddTypeMX, Domain\_Zone\_AddTypeTXT, Domain\_Zone\_AddTypeMXDetailedEmail, Domain\_Zone\_AddTypeCname, Domain\_Zone\_AddTypeURL, Domain\_Zone\_AddTypeAAAA, Billing\_Buy\_Hosting, Billing\_Buy\_Multihosting, Billing\_Buy\_Domain, Billing\_Buy\_DomainUsingContactsIds, Billing\_Buy\_Sms, Billing\_Transfer\_Domain, Multihosting\_HostingSetup\_Create, Multihosting\_Hosting\_Create, Hosting\_Dsn\_Access\_Create, Hosting\_Dsn\_Sqlserver\_Create, Hosting\_Dsn\_Mysql\_Create, Hosting\_Email\_Redirection\_Create, Hosting\_Email\_Antispam\_AddDomainToWhitelist, Hosting\_Email\_Antispam\_AddDomainToBlacklist, Hosting\_Email\_Autoresponse\_Create, Hosting\_Email\_Account\_Create, Hosting\_Ddbb\_MariaDb\_Create, Hosting\_Ddbb\_Sqlserver\_Create, Hosting\_Ddbb\_Postgre\_Create, Hosting\_Ddbb\_Mysql\_CreateWithIpAcl, Hosting\_Ddbb\_Mysql\_Create, Hosting\_Import\_ApplyForEmailAccountImap, Hosting\_Import\_ApplyForDdbb, Hosting\_Import\_ApplyForEmailAccountPop, Hosting\_Import\_ApplyForFtp, Hosting\_Subdomain\_Create

## OBJECT\_NOT\_EXISTS ( code: 2303 )

The request is valid but it can be only completed if the requested object (zone, hosting, email account ...) already exists.

Error message: *Param "%s" value doesn't exist.*

Can be thrown by commands: Dedicated\_Services\_ChangeStatus, Dedicated\_Control\_KillMysqlProcess, Dedicated\_HostingSetup\_Get, Dedicated\_HostingSetup\_Update, Dedicated\_Hosting\_GetSpace, Dedicated\_Hosting\_Activate, Dedicated\_Hosting\_Create, Dedicated\_Hosting\_Delete, Dedicated\_Hosting\_Deactivate, Dedicated\_Hosting\_GetSetupName, Dedicated\_Hosting\_ChangeSetup, Sms\_Programmed\_Cancel, Vps\_Services\_ChangeStatus, Vps\_Control\_KillMysqlProcess, Vps\_HostingSetup\_Get, Vps\_HostingSetup\_Update, Vps\_Hosting\_Activate, Vps\_Hosting\_Create, Vps\_Hosting\_Delete, Vps\_Hosting\_Deactivate, Vps\_Hosting\_GetSetupName, Vps\_Hosting\_ChangeSetup, Domain\_Zone\_DeleteTypeAname, Domain\_Zone\_DeleteTypeTXT, Domain\_Zone\_UpdateTypeAname, Domain\_Zone\_UpdateTypeA, Domain\_Zone\_DeleteTypeSRV, Domain\_Zone\_DeleteTypeSPF, Domain\_Zone\_DeleteTypeAAAA, Domain\_Zone\_UpdateTypeAAAA, Domain\_Zone\_AddTypeMX, Domain\_Zone\_DeleteTypeCname, Domain\_Zone\_DeleteTypeMX, Domain\_Zone\_Delete, Domain\_Zone\_AddTypeMXDetailedEmail, Domain\_Zone\_DeleteTypeA, Domain\_Zone\_AddTypeCname, Billing\_NoRenewable\_SetRenewable, Billing\_Renew\_Hosting, Billing\_Renew\_Domain, Billing\_Transfer\_Domain, Billing\_Autorenew\_SetNoRenewable, Billing\_Autorenew\_SetOff, Billing\_Autorenew\_SetOn, Multihosting\_HostingSetup\_Get, Multihosting\_HostingSetup\_Update, Multihosting\_Hosting\_GetSpace, Multihosting\_Hosting\_Activate, Multihosting\_Hosting\_Delete, Multihosting\_Hosting\_Deactivate, Multihosting\_Hosting\_ChangeSetup, Hosting\_Ftp\_Account\_SetPassword, Hosting\_Ftp\_Account\_Delete, Hosting\_Email\_Redirection\_Delete, Hosting\_Email\_Redirection\_Update, Hosting\_Email\_Antispam\_RemoveDomainFromWhitelist, Hosting\_Email\_Antispam\_RemoveDomainFromBlacklist, Hosting\_Email\_Autoresponse\_Get, Hosting\_Email\_Autoresponse\_Activate, Hosting\_Email\_Autoresponse\_Create, Hosting\_Email\_Autoresponse\_Delete, Hosting\_Email\_Autoresponse\_Deactivate, Hosting\_Email\_Account\_Empty, Hosting\_Email\_Account\_SetPassword, Hosting\_Email\_Account\_RemoveCopyTo, Hosting\_Email\_Account\_Activate, Hosting\_Email\_Account\_SetCopyTo, Hosting\_Email\_Account\_Delete, Hosting\_Email\_Account\_GetCopyTo, Hosting\_Email\_Account\_Deactivate, Hosting\_Import\_ApplyForEmailAccountImap, Hosting\_Import\_ApplyForDdbb, Hosting\_Import\_ApplyForEmailAccountPop, Hosting\_Import\_ApplyForFtp, Hosting\_Subdomain\_Create, Hosting\_Subdomain\_Delete

## OBJECT\_STATUS\_PROHIBITS\_OP ( code: 2304 )

The request is valid but it can be only completed if the requested object status meets some condition.

Some condition must be achieved before executing the command again.

Please, read the command documentation for additional information about command limitations and requirements.

Error message: *Param "%s" current value status prohibits operation.*

Can be thrown by commands: Domain\_WhoisPrivacy\_Set, Domain\_UrlShortener\_GetAll, Domain\_UrlShortener\_Activate, Domain\_UrlShortener\_Create, Domain\_UrlShortener\_Delete, Domain\_UrlShortener\_Deactivate, Domain\_Zone\_UrlShortener\_GetAll, Domain\_Zone\_UrlShortener\_Activate, Domain\_Zone\_UrlShortener\_Create, Domain\_Zone\_UrlShortener\_Delete, Domain\_Zone\_UrlShortener\_Deactivate, Domain\_Zone\_AddTypeMX, Domain\_Zone\_AddTypeMXDetailedEmail, Domain\_Transfer\_GetStatus, Domain\_Transfer\_GetAuthcode, Domain\_Transfer\_SetAuthcode, Domain\_Transfer\_Restart, Domain\_Transfer\_Cancel, Billing\_Autorenew\_SetOn, Hosting\_Ftp\_Account\_Create, Hosting\_UrlShortener\_GetAll, Hosting\_UrlShortener\_Activate, Hosting\_UrlShortener\_Create, Hosting\_UrlShortener\_Delete, Hosting\_UrlShortener\_Deactivate, Hosting\_TlsMail\_Get, Hosting\_TlsMail\_Activate, Hosting\_TlsMail\_Deactivate, Hosting\_Subdomain\_GetAll, Hosting\_Subdomain\_Create, Hosting\_Subdomain\_Delete

## OBJECT\_ASSOCIATION\_PROHIBITS\_OP ( code: 2305 )

The request is valid but can't be run because the object is currently associated to another object. Association must be broken before running this command.

Error message: *Param "%s" current value is associated to another object. Association must be broken.*

Can be thrown by commands: Domain\_Zone\_DeleteTypeAAAA, Domain\_Zone\_DeleteTypeCname, Domain\_Zone\_Delete, Domain\_Zone\_DeleteTypeA

## PARAMETER\_VALUE\_POLICY\_ERROR ( code: 2306 )

A internal or external parameter value is not valid. It's registry implementation dependant. This is relatively common if domain contacts are missing.

Error message: *Some argument value is not currently valid*

Can be thrown by commands: Domain\_Dnss\_Set, Billing\_Autorenew\_SetOn, Hosting\_Import\_ApplyForFtp

## UNIMPLEMENTED\_OBJECT\_SERVICE ( code: 2307 )

The request is valid but requested object won't admit it, neither now nor later. Please, read the command documentation for additional information about command limitations and requirements.

Error message: *Command cannot be run on requested object.*

Can be thrown by commands: Dedicated\_ForVps\_Vps\_Create\_Unmanaged, Dedicated\_ForVps\_Vps\_Create\_Managed, Dedicated\_ForVps\_Vps\_GetAll, Dedicated\_ForVps\_Vps\_GetTypes, Dedicated\_ForVps\_Vps\_Activate, Dedicated\_ForVps\_Vps\_Delete, Dedicated\_ForVps\_Vps\_Deactivate, Dedicated\_ForVps\_Vps\_Update, Dedicated\_GetMonitorizedPorts, Vps\_GetMonitorizedPorts, Domain\_CheckForTransfer, Domain\_Transfer\_GetAuthcode, Domain\_Transfer\_SetAuthcode, Domain\_CheckForRegister, Domain\_NameServer\_GetAll, Billing\_Renew\_Domain, Billing\_Autorenew\_SetOn, Hosting\_Dsn\_Access\_GetAll, Hosting\_Dsn\_Access\_Create, Hosting\_Dsn\_Access\_Delete, Hosting\_Dsn\_Sqlserver\_GetAll, Hosting\_Dsn\_Sqlserver\_Create, Hosting\_Dsn\_Sqlserver\_Delete, Hosting\_Dsn\_Mysql\_GetAll, Hosting\_Dsn\_Mysql\_Create, Hosting\_Dsn\_Mysql\_Delete, Hosting\_Php\_DeactivatePhpExecutionOnHtml, Hosting\_Php\_ActivatePhpExecutionOnHtml, Hosting\_Php\_SetVersion, Hosting\_Php\_OverwritePhpIniWithDefaults, Hosting\_Php\_PhpExecutionOnHtmlIsActive, Hosting\_Php\_OverwritePhpIni, Hosting\_GetMailInfo, Hosting\_Ftp\_Get, Hosting\_Ftp\_Account\_GetAll, Hosting\_Ftp\_Account\_SetPassword, Hosting\_Ftp\_Account\_Create, Hosting\_Ftp\_Account\_Delete, Hosting\_Ftp\_Activate, Hosting\_Ftp\_Deactivate, Hosting\_Ftp\_GetAccountsLimit, Hosting\_Ssh\_Get, Hosting\_Ssh\_Activate, Hosting\_Ssh\_Deactivate, Hosting\_TlsMail\_Get, Hosting\_TlsMail\_Activate, Hosting\_TlsMail\_Deactivate, Hosting\_Email\_Antivirus\_UpdateAttachedFileFilter, Hosting\_Email\_Antivirus\_GetConfig, Hosting\_Email\_Antivirus\_GetAttachedFilesFilters, Hosting\_Email\_Antivirus\_SetConfig, Hosting\_Email\_Redirection\_ActivateCatchAll, Hosting\_Email\_Redirection\_Get, Hosting\_Email\_Redirection\_GetAll

Hosting\_Email\_Redirection\_DeactivateCatchAll, Hosting\_Email\_Redirection\_Create,  
 Hosting\_Email\_Redirection\_Delete, Hosting\_Email\_Redirection\_Update,  
 Hosting\_Email\_Antispam\_GetWhitelist, Hosting\_Email\_Antispam\_AddDomainToWhitelist,  
 Hosting\_Email\_Antispam\_AddDomainToBlacklist, Hosting\_Email\_Antispam\_GetScore,  
 Hosting\_Email\_Antispam\_RemoveDomainFromWhitelist, Hosting\_Email\_Antispam\_GetConfig,  
 Hosting\_Email\_Antispam\_GetBlacklist, Hosting\_Email\_Antispam\_SetScore,  
 Hosting\_Email\_Antispam\_Activate, Hosting\_Email\_Antispam\_Deactivate,  
 Hosting\_Email\_Antispam\_RemoveDomainFromBlacklist, Hosting\_Email\_Antispam\_SetConfig,  
 Hosting\_Email\_Antispam\_SetConfig, Hosting\_Email\_Antispam\_Deactivate,  
 Hosting\_Email\_Autoresponse\_Activate, Hosting\_Email\_Autoresponse\_Get,  
 Hosting\_Email\_Autoresponse\_Delete, Hosting\_Email\_Autoresponse\_Create,  
 Hosting\_Email\_Autoresponse\_Deactivate, Hosting\_Email\_Autoresponse\_Deactivate,  
 Hosting\_Email\_Account\_Empty, Hosting\_Email\_Account\_Get, Hosting\_Email\_Account\_GetAll,  
 Hosting\_Email\_Account\_SetPassword, Hosting\_Email\_Account\_RemoveCopyTo,  
 Hosting\_Email\_Account\_Activate, Hosting\_Email\_Account\_Create, Hosting\_Email\_Account\_SetCopyTo,  
 Hosting\_Email\_Account\_Delete, Hosting\_Email\_Account\_GetCopyTo,  
 Hosting\_Email\_Account\_Deactivate, Hosting\_Email\_Filter\_SetGeolocalization,  
 Hosting\_Email\_Filter\_GetAll, Hosting\_Email\_Filter\_GetGeolocalization,  
 Hosting\_Email\_Filter\_CreateForSpam, Hosting\_Email\_Filter\_Create, Hosting\_Email\_Filter\_Delete,  
 Hosting\_Ddbb\_MariaDb\_GetAll, Hosting\_Ddbb\_MariaDb\_SetPassword, Hosting\_Ddbb\_MariaDb\_Create,  
 Hosting\_Ddbb\_MariaDb\_Delete, Hosting\_Ddbb\_Sqlserver\_GetAll, Hosting\_Ddbb\_Sqlserver\_SetPassword,  
 Hosting\_Ddbb\_Sqlserver\_SetTransactionsSize, Hosting\_Ddbb\_Sqlserver\_SetDataSize,  
 Hosting\_Ddbb\_Sqlserver\_GetGlobalSpaceInfo, Hosting\_Ddbb\_Sqlserver\_Create,  
 Hosting\_Ddbb\_Sqlserver\_Delete, Hosting\_Ddbb\_Sqlserver\_GetSpaceInfo,  
 Hosting\_Ddbb\_MongoDb\_GetAll, Hosting\_Ddbb\_MongoDb\_SetPassword,  
 Hosting\_Ddbb\_MongoDb\_Delete, Hosting\_Ddbb\_Postgre\_GetAll, Hosting\_Ddbb\_Postgre\_SetPassword,  
 Hosting\_Ddbb\_Postgre\_Create, Hosting\_Ddbb\_Postgre\_Delete, Hosting\_Ddbb\_Mysql\_CreateWithIpAcl,  
 Hosting\_Ddbb\_Mysql\_GetAll, Hosting\_Ddbb\_Mysql\_SetPassword,  
 Hosting\_Ddbb\_Mysql\_GetAclConfiguration, Hosting\_Ddbb\_Mysql\_GetInfo,  
 Hosting\_Ddbb\_Mysql\_SetHostAcl, Hosting\_Ddbb\_Mysql\_BackupTask\_Set,  
 Hosting\_Ddbb\_Mysql\_BackupTask\_GetAll, Hosting\_Ddbb\_Mysql\_BackupTask\_Delete,  
 Hosting\_Ddbb\_Mysql\_SetIpAcl, Hosting\_Ddbb\_Mysql\_Create, Hosting\_Ddbb\_Mysql\_Delete,  
 Hosting\_Ddbb\_Mysql\_Backup\_Get, Hosting\_Ddbb\_Mysql\_Backup\_Restore

### COMMAND\_FAILED ( code: 2400 )

Command failed because of an internal error. Command request is valid but we couldn't finish it. You can try again later o contact with support for additional details.

Error message: *Command failed.*

Can be throwed by commands: All existing commands might return this error.

### COMMAND\_FAILED\_FATAL ( code: 2500 )

Command execution ended with an internal fatal error.

The execution might end OK but something wrent wrong. Or maybe command execution doesn't started at all.

We'll look at the problem and fix it ASAP.

Error message: *Command failed, internal error.*

Can be throwed by commands: All existing commands might return this error.

### COMMAND\_TIMEOUT ( code: 2501 )

We received a timeout during execution of command.

Command execution could end ok but we didn't get the response at time.

Error message: *Timeout on execution, command might be complete.*

Can be throwed by commands: All existing commands might return this error.

## COMMAND\_CONNECTIONSLIMIT ( code: 2502 )

User tried to complete too many command executions in a period of time and/or got banned.  
Please, just wait and try again later.

Error message: *Too many connections, command won't execute. Try again later.*  
Can be thrown by commands: None

## COMMAND\_MAINTENANCE\_MODE ( code: 2503 )

Server is in maintenance mode.

Error message: *Maintenance mode. Try again later.*  
Can be thrown by commands: None

## COMMAND\_PENDING\_PREVIOUS\_EXECUTION ( code: 3101 )

There is a existing execution for this delayed command.

Error message: *There is a previous pending execution.*  
Can be thrown by commands: Hosting\_Ftp\_Account\_SetPassword, Hosting\_Ftp\_Account\_Create, Hosting\_Ftp\_Account\_Delete

## COMMAND\_FTP\_MAX\_ACCOUNTS ( code: 3601 )

Hosting already reached maximum FTP accounts number.

Error message: *Reached max number of FTP accounts.*  
Can be thrown by commands: None

## COMMAND\_EMAIL\_REDIRECTION\_MAX\_ACCOUNTS ( code: 3602 )

Hosting already reached maximum email redirections number.

Error message: *Reached max number of email redirections.*  
Can be thrown by commands: Hosting\_Email\_Redirection\_Create

## COMMAND\_EMAIL\_DISTRIBUTIONLIST\_MAX ( code: 3603 )

Hosting already reached maximum distribution list number.

Error message: *Reached max number of email distribution lists.*  
Can be thrown by commands: None

## COMMAND\_SERVER\_KVM\_GET\_ERROR ( code: 3604 )

Server is missing KVM information.

Error message: *Missing KVM information.*  
Can be thrown by commands: Dedicated\_Kvm\_Get, Housing\_Kvm\_Get

## COMMAND\_DEDICATED\_SERVER\_OS\_ERROR ( code: 3605 )

Server Operating System can't run this command.

Error message: *Server O.S. is not compatible with this command.*  
Can be thrown by commands: Dedicated\_Control\_GetApacheProcesses, Dedicated\_Control\_KillMysqlProcess, Vps\_Control\_KillMysqlProcess

## COMMAND\_LICENSE\_MAX\_HOSTINGS ( code: 3606 )

Server has reached the maximum number of hostings.  
This limit is defined by the current server license.

Error message: *Reached maximum number of hostings - license limit.*

Can be throwed by commands: Dedicated\_Hosting\_Create, Vps\_Hosting\_Create, Multihosting\_Hosting\_Create

## COMMAND\_BILLING\_PRODUCT\_ERROR ( code: 3607 )

The requested product is not valid and billing process is aborted.  
Remember that not all domain TLDs share same registration periods.

Error message: *Invalid product.*

Can be throwed by commands: Billing\_Renew\_Hosting, Billing\_Renew\_Multihosting, Billing\_Renew\_Domain, Billing\_Renew\_Vps, Billing\_Renew\_Sms, Billing\_Price\_Hosting, Billing\_Price\_Multihosting, Billing\_Price\_Vps\_Managed\_Linux, Billing\_Price\_Vps\_Managed\_Windows, Billing\_Price\_Vps\_Unmanaged\_Linux, Billing\_Price\_Vps\_Unmanaged\_Windows, Billing\_Price\_Domain, Billing\_Price\_Sms, Billing\_Period\_Hosting, Billing\_Period\_Multihosting, Billing\_Period\_Domain, Billing\_Period\_Vps, Billing\_Buy\_Hosting, Billing\_Buy\_Multihosting, Billing\_Buy\_Vps\_Managed\_Linux, Billing\_Buy\_Vps\_Managed\_Windows, Billing\_Buy\_Vps\_Unmanaged\_Linux, Billing\_Buy\_Vps\_Unmanaged\_Windows, Billing\_Buy\_Domain, Billing\_Buy\_DomainUsingContactsIds, Billing\_Buy\_Sms, Billing\_Transfer\_Domain, Billing\_Help\_DomainExtraData

## COMMAND\_BILLING\_CREDIT ( code: 3608 )

The user has not enough credit for completing the billing process.

Error message: *Not enough credit.*

Can be throwed by commands: Billing\_Renew\_Hosting, Billing\_Renew\_Multihosting, Billing\_Renew\_Domain, Billing\_Renew\_Sms, Billing\_Buy\_Hosting, Billing\_Buy\_Multihosting, Billing\_Buy\_Domain, Billing\_Buy\_DomainUsingContactsIds, Billing\_Buy\_Sms, Billing\_Transfer\_Domain

## COMMAND\_EMAIL\_MAX\_HOSTINGS ( code: 3609 )

Server cannot create more hostings currently on this server.  
It has reached the email accounts limit. Please, delete some and try again.

Error message: *Reached maximum number of email accounts on server.*

Can be throwed by commands: Hosting\_Email\_Account\_Create

## COMMAND\_DOMAIN\_FORBIDDEN\_CONTACT\_UPDATE ( code: 3610 )

Domain contacts update can't be run because of registry policy blocks certaing contactType data change.  
Example: .es registrant and .eu registrant contacts update are forbidden. These must be removed from request.

Error message: *Requested contact change is forbidden by registry policy.*

Can be throwed by commands: Domain\_Contacts\_Set, Domain\_Contacts\_SetAll

## COMMAND\_DDBB\_SQLSERVER\_NOT\_ENOUGH\_SPACE ( code: 3612 )

There is not enough space for the requested SqlServer operation.

Error message: *Not enough space. Check params "%s".*

Can be throwed by commands: Hosting\_Ddbb\_Sqlserver\_Create

## COMMAND\_EMAIL\_DISTRIBUTION\_LIST\_EXISTS ( code: 3613 )

Can't create the account. It exists a distribution list with the same email.

Error message: *It exists a distribution list with the same email.*

Can be thrown by commands: None

### COMMAND\_SMS\_NOT\_REMAINING\_MESSAGES ( code: 3614 )

The user has not remaining messages for completing the sending process.

Error message: *Not remaining messages.*

Can be thrown by commands: None

### COMMAND\_MYSQL\_MAX\_DATABASES ( code: 3615 )

Server cannot create more MySQL databases currently on this server.

Error message: *Reached max number of MySQL databases.*

Can be thrown by commands: None

### COMMAND\_DOMAIN\_IP\_NOT\_RESOLVING\_DNS ( code: 3616 )

Some IP doesn't resolve a DNS.

Error message: *Some IP doesn't resolve a DNS.*

Can be thrown by commands: None

### COMMAND\_EMAIL\_NOT\_ENABLED ( code: 3617 )

Email accounts are not allowed on this hosting.

Error message: *Email accounts are not allowed on this hosting.*

Can be thrown by commands: None

### COMMAND\_REGENERATE\_PASSWORD\_SEND\_METHOD ( code: 3618 )

Send method is not valid for this contact.

Error message: *Send method is not valid for this contact.*

Can be thrown by commands: None

### COMMAND\_EMAIL\_ACCOUNT\_SET\_REPEATED\_PASSWORD ( code: 3619 )

Password can not match any of the last 5 used in this account..

Error message: *Password can not match any of the last 5 used in this account.*

Can be thrown by commands: Hosting\_Email\_Account\_SetPassword

### COMMAND\_BILLING\_DOMAIN\_FORBIDDEN\_PREDEFINED\_CONTACTS ( code: 3620 )

Some registries have special requirements that must be fit by user default domain registration contacts.

Error message: *Invalid predefined contacts for this TLD .%s*

Can be thrown by commands: Billing\_Buy\_Domain, Billing\_Buy\_DomainUsingContactsIds

### COMMAND\_DOMAIN\_TRANSFER\_ALREADY\_APPROVED ( code: 3630 )

Domain transfer was already approved.

Error message: *Domain transfer was already approved.*

Can be thrown by commands: Domain\_Transfer\_Restart, Domain\_Transfer\_Cancel

### **COMMAND\_DOMAIN\_CERT\_ALREADY\_ASSOCIATED ( code: 3631 )**

A cert has already been associated with this domain.

Error message: *A cert has already been associated with this domain.*

Can be thrown by commands: None

### **COMMAND\_BILLING\_PRODUCT\_TEMPORARILY\_DISABLED ( code: 3621 )**

Product reference temporarily disabled by promotion or other reasons

Error message: *Product reference temporarily disabled by promotion or other reasons.*

Can be thrown by commands: None

### **COMMAND\_NFS\_VOLUME\_DOES\_NOT\_EXIST ( code: 3640 )**

Server does not have a nfs volume

Error message: *Server does not have a nfs volume*

Can be thrown by commands: None

### **COMMAND\_PARAM\_IS\_NOT\_A\_VALID\_CERTIFICATE ( code: 3641 )**

(Missing documentation).

Error message: *(Missing response message, the error code might be on implementation).*

Can be thrown by commands: None

### **COMMAND\_DOMAIN\_SERVER\_ZONE\_NOT\_FOUND ( code: 3645 )**

Domain's zone for server not found

Error message: *Domain's zone for server not found*

Can be thrown by commands: None



# Commands

## AdditionalStorage commands

### AdditionalStorage\_Repository\_GetAll

**Description:**

Returns the repositories of an additional storage.

Version: 1.0

**Parameters:**

**additionalStorage** (string): Additional storage hostname. The parameter is required.

**repositoryType** (string): All repository types or one of them. The parameter is required. Accepted parameter values: "all", "ftp", "webdav", "svn", "git".

**Response:**

type: **struct**

Example:

'None'

### AdditionalStorage\_Repository\_SetPassword

**Description:**

It changes a repository password in an additional storage.

Version: 1.0

**Parameters:**

**additionalStorage** (string): Additional storage hostname. The parameter is required.

**type** (string): Type. The parameter is required. Accepted parameter values: "ftp", "webdav", "svn", "git".

**username** (string): Account username. The parameter is required. Parameter value must be an alphanumeric string. Parameter length must be between 6 and 24 characters.

**folder** (string): Folder associated to the account. The parameter is required. Parameter value must be a valid name. Parameter length must be between 3 and 24 characters.

**password** (string): New account password. The parameter is required. Parameter value must be a string between 4 and 16 alphanumeric characters with fewer than 3 correlative or reaped characters.

**Response:**

type: **struct**

Example:

'None'

### AdditionalStorage\_Repository\_Create

**Description:**

It creates a repository in an additional storage.

Version: 1.0

**Parameters:**

**additionalStorage** (string): Additional storage hostname. The parameter is required.

**type** (string): Type. The parameter is required. Accepted parameter values: "ftp", "webdav", "svn", "git".

**username** (string): Account username. The parameter is required. Parameter value must be an alphanumeric string. Parameter length must be between 6 and 24 characters.

**password** (string): Account password. The parameter is required. Parameter value must be a string

between 4 and 16 alphanumeric characters with fewer than 3 correlative or reapedted characters.

**folder** (**string**): Folder associated to the account. The parameter is required. Parameter value must be a valid name. Parameter length must be between 3 and 24 characters.

**Response:**

type: **struct**

Example:

'None'

## AdditionalStorage\_Repository\_Delete

**Description:**

It deletes a repository in an additional storage.

Version: 1.0

**Parameters:**

**additionalStorage** (**string**): Additional storage hostname. The parameter is required.

**type** (**string**): Type. The parameter is required. Accepted parameter values: "ftp", "webdav", "svn", "git".

**username** (**string**): Account username. The parameter is required. Parameter value must be an alphanumeric string. Parameter length must be between 6 and 24 characters.

**folder** (**string**): Folder associated to the account. The parameter is required. Parameter value must be a valid name. Parameter length must be between 3 and 24 characters.

**Response:**

type: **struct**

Example:

'None'

## Billing commands

### Billing\_NoRenewable\_GetAll

**Description:**

Returns no-renewable services list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

'None'

### Billing\_NoRenewable\_SetRenewable

**Description:**

Set service as renewable. Service expiration will be reported at expiration date, service will be eligible for renew.

Service can be autorenewed but you must set autorenew on ( see Billing\_Autorenew\_SetOn ).

Version: 1.0

**Parameters:**

**id** (**int**): Service ID ( as returned by Billing\_NoRenewable\_GetAll ). The parameter is required. Parameter value must be a valid integer.

**Response:**

type: **void**

### Billing\_Renew\_Hosting

**Description:**

Manually renew an existing hosting. Payment is done at runtime.

Hosting must exist in a Dinahosting server.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

Version: 1.2

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**period** (**int**): Contratation period ( months ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.

**Response:**

type: **void**

### Billing\_Renew\_Multihosting

**Description:**

Manually renew an existing multihosting. Payment is done at runtime.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

Version: 1.2

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**period** (**int**): Renew period ( years ). The parameter is required. Parameter value must be a valid integer.

Parameter value must be between values 1 and 5 inclusive.

**Response:**

type: **void**

## Billing\_Renew\_Domain

**Description:**

Manually renew an existing domain name. Payment is done at runtime.

Domain must be currently registered at Dinahosting.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

1.2 -> 1.3: OBJECT\_NOT\_EXISTS error.

1.3 -> 1.4: New TLDs are supported.

Version: 1.4

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**period** (**int**): Renew period ( years ). The parameter is required. Parameter value must be a valid integer.

Parameter value must be between values 1 and 10 inclusive.

**whoisProtection** (**bool**): Default is false. Set true to also renovate an already active whois protection for this domain. The parameter is optional. Parameter value must be a boolean ( true or false ).

**paymentMethodType** (**string**): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo", "bono".

**Response:**

type: **void**

## Billing\_Renew\_Vps

**Description:**

Renews a VPS.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**period** (**int**): Contratation period (months). The parameter is required. Accepted parameter values: "1", "3", "12".

**Response:**

type: **boolean**

Example:

'None'

## Billing\_Renew\_Sms

**Description:**

Manually renew an existing SMS account. Payment is done at runtime.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

Version: 1.2

**Parameters:**

**account** (**string**): Sms account number. The parameter is required.

**credit** (**int**): Additional credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50", "100", "1000", "5000", "20000".

**Response:**

type: **void**

## Billing\_Price\_Hosting

**Description:**

Get price for a hosting ( new or renew ).

Version: 1.2

**Parameters:**

**period** (**int**): Contratation period ( months ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.

**hostingType** (**int**): Hosting type. 1 = Basic, 2 = Advanced, 3 = Advanced Plus, 10 = Email, 419 = Wordpress, 420 = Wordpress Optimized, 421 = Wordpress Pro, 422 = Wordpress Pro Optimized, 425 = Prestashop, 426 = Prestashop Pro, 428 = WooCommerce, 429 = WooCommerce Pro, 431 = MediaWiki, 470 = Joomla, 480 = Moodle, 490 = Drupal. The parameter is required. Accepted parameter values: "1", "2", "3", "10", "419", "420", "421", "422", "425", "426", "428", "429", "431", "470", "480", "490".

**serverOs** (**string**): Host server operating system. The parameter is required. Accepted parameter values: "LINUX", "WINDOWS".

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create", "renew".

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD", "EUR".

**Response:**

type: **float**

Example:

'None'

## Billing\_Price\_Multihosting

**Description:**

Get price for a multihosting ( new or renew ).

Version: 1.2

**Parameters:**

**period** (**int**): Contratation period ( years ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 5 inclusive.

**serverOs** (**string**): Host server operating system. The parameter is required. Accepted parameter values: "LINUX", "WINDOWS".

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create", "renew".

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values:

"USD","EUR".

**Response:**

type: **float**

Example:

'None'

## Billing\_Price\_Vps\_Managed\_Linux

**Description:**

Gets the price of a managed linux VPS.

Version: 1.0

**Parameters:**

**type** (**string**): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".

**period** (**int**): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".

**vcpusCount** (**int**): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".

**diskType** (**string**): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

**Response:**

type: **float**

Example:

'None'

## Billing\_Price\_Vps\_Managed\_Windows

**Description:**

Gets the price of a managed windows VPS.

Version: 1.0

**Parameters:**

**type** (**string**): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".

**period** (**int**): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".

**vcpusCount** (**int**): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".

**diskType** (**string**): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

**Response:**

type: **float**

Example:

'None'

## Billing\_Price\_Vps\_Unmanaged\_Linux

### Description:

Gets the price of an unmanaged linux VPS.

Version: 1.0

### Parameters:

**type** (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro", "V-Server Pro Plus", "V-Server Elite".

**period** (int): Contratation period (months). The parameter is required. Accepted parameter values: "1", "3", "12".

**vcpusCount** (int): Number of vcpus. The parameter is required. Accepted parameter values: "1", "2", "4".

**diskType** (string): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**action** (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create", "renew".

**currency** (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD", "EUR".

### Response:

type: float

Example:

'None'

## Billing\_Price\_Vps\_Unmanaged\_Windows

### Description:

Gets the price of an unmanaged windows VPS.

Version: 1.0

### Parameters:

**type** (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro", "V-Server Pro Plus", "V-Server Elite".

**period** (int): Contratation period (months). The parameter is required. Accepted parameter values: "1", "3", "12".

**vcpusCount** (int): Number of vcpus. The parameter is required. Accepted parameter values: "1", "2", "4".

**diskType** (string): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**diskImage** (string): Disk image. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

**action** (string): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create", "renew".

**currency** (string): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD", "EUR".

### Response:

type: float

Example:

'None'

## Billing\_Price\_Domain

### Description:

Get price for a domain name ( register, transfer or renew ).

1.0 -> 1.1: New TLDs are supported.



Version: 1.1

**Parameters:**

**tld** (**string**): Domain name TLD. Example: com.es (as is). The parameter is required.

**period** (**int**): Registration period ( years ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew","transfer".

**whoisProtection** (**bool**): Default is false. Set true to include domain whois protection price. The parameter is optional. Parameter value must be a boolean ( true or false ).

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

**Response:**

 type: **float**

Example:

*'None'*

## Billing\_Price\_Sms

**Description:**

Get price for a SMS account ( new or renew ).

Version: 1.0

**Parameters:**

**credit** (**int**): Initial credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50","100","1000","5000","20000".

**action** (**string**): Action done over domain. Different actions may have different prices. The parameter is required. Accepted parameter values: "create","renew".

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD","EUR".

**Response:**

 type: **float**

Example:

*'None'*

## Billing\_Period\_Hosting

**Description:**

Get valid contratation periods (years) for each hosting type.

Version: 1.0

**Parameters:**

**action** (**string**): Action done over hosting. Different actions may have different periods. The parameter is required. Accepted parameter values: "create","renew".

**Response:**

 type: **struct**

Example:

```
array('1' => array(1,2,3,5,10),
```

```
'2' => array(1,2,3,5,10),
```

```
'3' => array(1,2,3,5,10),
```

```
'419' => array(1,2,3,5,10),
```

```
'425' => array(1,2,3,5,10),
```

```
'431' => array(1,2,3,5,10),
'436' => array(1,2,3,5,10),
'470' => array(1,2,3,5,10),
'498' => array(1,2,3,5,10),
)
```

## Billing\_Period\_Multihosting

### Description:

Get valid contratation periods ( years ) for a multihosting.

Version: 1.0

### Parameters:

**action** (string): Action done over multihosting. Different actions may have different periods. The parameter is required. Accepted parameter values: "create", "renew".

### Response:

type: **struct**

Example:

```
array('mh' => array(1,2,3,4,5),
)
```

## Billing\_Period\_Domain

### Description:

Get valid contratation periods (years) for each domain TLD and action.

1.1 -> 1.2: New TLDs are supported.

Version: 1.2

### Parameters:

**action** (string): Action done over domain. Different actions may have different periods. The parameter is required. Accepted parameter values: "create", "renew", "transfer".

**tld** (string): Domain name TLD. Example: com.es (as is). The parameter is optional.

### Response:

type: **struct**

Example:

(Example too long, see web.)

## Billing\_Period\_Vps

### Description:

Gets the selectable billing periods for VPS.

Version: 1.0

### Parameters:

**action** (string): Action done over VPS. Different actions may have different periods. The parameter is required. Accepted parameter values: "create", "renew".

### Response:

type: **struct**

Example:

'None'

## Billing\_Buy\_Hosting

### Description:

Buy a new hosting.

1.2 -> 1.3: Improved simulation mode. Deprecated hostingType 810.

1.3 -> 1.4: COMMAND\_BILLING\_CREDIT error.

Version: 1.4

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**period** (int): Contratation period ( months ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.

**hostingType** (int): Hosting type. 1 = Basic, 2 = Advanced, 3 = Advanced Plus, 10 = Email, 419 = Wordpress, 420 = Wordpress Optimized, 421 = Wordpress Pro, 422 = Wordpress Pro Optimized, 425 = Prestashop, 426 = Prestashop Pro, 428 = WooCommerce, 429 = WooCommerce Pro, 431 = MediaWiki, 470 = Joomla, 480 = Moodle, 490 = Drupal. The parameter is required. Accepted parameter values: "1","2","3","10","419","420","421","422","425","426","428","429","431","470","480","490".

**serverOs** (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".

### Response:

type: void

## Billing\_Buy\_Multihosting

### Description:

Buy a new multihosting.

1.2 -> 1.3: Improved simulation mode.

1.3 -> 1.4: COMMAND\_BILLING\_CREDIT error.

Version: 1.4

### Parameters:

**period** (int): Contratation period ( years ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 5 inclusive.

**serverOs** (string): Host server operating system. The parameter is required. Accepted parameter values: "LINUX","WINDOWS".

### Response:

type: void

## Billing\_Buy\_Vps\_Managed\_Linux

### Description:

Buys a managed linux VPS.

Version: 1.0

### Parameters:

**type** (string): Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".

**period** (int): Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".

**vcpusCount** (int): Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".

**diskType** (string): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**diskImage** (**string**): Disk image. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

**Response:**

type: **boolean**

Example:

'None'

## Billing\_Buy\_Vps\_Managed\_Windows

**Description:**

Buys a managed windows VPS.

Version: 1.0

**Parameters:**

**type** (**string**): Type. The parameter is required. Accepted parameter values: "V-Server Pro", "V-Server Pro Plus", "V-Server Elite".

**period** (**int**): Contratation period (months). The parameter is required. Accepted parameter values: "1", "3", "12".

**vcpusCount** (**int**): Number of vcpus. The parameter is required. Accepted parameter values: "1", "2", "4".

**diskType** (**string**): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**Response:**

type: **boolean**

Example:

'None'

## Billing\_Buy\_Vps\_Unmanaged\_Linux

**Description:**

Buys an unmanaged linux VPS.

Version: 1.0

**Parameters:**

**type** (**string**): Type. The parameter is required. Accepted parameter values: "V-Server Pro", "V-Server Pro Plus", "V-Server Elite".

**period** (**int**): Contratation period (months). The parameter is required. Accepted parameter values: "1", "3", "12".

**vcpusCount** (**int**): Number of vcpus. The parameter is required. Accepted parameter values: "1", "2", "4".

**diskType** (**string**): Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**diskImage** (**string**): Disk image. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

**Response:**

type: **boolean**

Example:

'None'

## Billing\_Buy\_Vps\_Unmanaged\_Windows

**Description:**

Buys an unmanaged windows VPS.

Version: 1.0

**Parameters:**

**type (string):** Type. The parameter is required. Accepted parameter values: "V-Server Pro","V-Server Pro Plus","V-Server Elite".

**period (int):** Contratation period (months). The parameter is required. Accepted parameter values: "1","3","12".

**vcpusCount (int):** Number of vcpus. The parameter is required. Accepted parameter values: "1","2","4".

**diskType (string):** Disk type. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk types. The parameter is required.

**diskImage (string):** Disk image. Use the command 'Billing\_GetConfigurableVpsOptions' to get the list of available disk images. The parameter is required.

**Response:**

type: **boolean**

Example:

'None'

## Billing\_Buy\_Domain

**Description:**

Buy and register a unused domain name.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

1.2 -> 1.3: New TLDs are supported.

1.3 -> 1.4: New contact keys (RegistrantIdentification, AdminIdentification, TechIdentification, BillingIdentification) - required for esNic domain validation.

1.4 -> 1.5: New optional contact keys (RegistrantFax, AdminFax, TechFax, BillingFax)

1.5 -> 1.6: New parameter extraData.

1.6 -> 1.7: New validation for esNic registration and new error code: COMMAND\_BILLING\_DOMAIN\_FORBIDDEN\_PREDEFINED\_CONTACTS

Version: 1.7

**Parameters:**

**domain (string):** Domain name. The parameter is required.

**period (int):** Registration period ( years ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.

**contacts (struct):** Array containing contacts ( see Domain\_Contacts\_GetAll ). Omit OrganizationName and LegalForm parameters in cases where contact is not an organization. The parameter is required. Parameter value must be a valid contacts.

**dnss (array):** Array containing at least 2 DNS strings. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.

**whoisProtection (bool):** Default is false. Set true to buy domain whois protection. Is activated immediatly after register this domain. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showRegistrantInfoOnWhois (bool):** Default is false. Set true to show registrant info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showAdminContactInfoOnWhois (bool):** Default is false. Set true to show admin contact info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showTechContactInfoOnWhois (bool):** Default is false. Set true to show tech contact info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**paymentMethodType (string):** Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".

**extraData (struct):** Hash array containing additional data. The parameter is optional.

**Response:**

type: **void**

## Billing\_Buy\_DomainUsingContactsIds

### Description:

Buy and register a unused domain name.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

1.2 -> 1.3: New TLDs are supported.

1.3 -> 1.4: EsNic domain validation for technical contact: must be a natural person

1.4 -> 1.5: New parameter extraData.

1.5 -> 1.6: New validation for esNic registration and new error code: COMMAND\_BILLING\_DOMAIN\_FORBIDDEN\_PREDEFINED\_CONTACTS

Version: 1.6

### Parameters:

**domain** (string): Domain name. The parameter is required.

**period** (int): Registration period ( years ). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.

**contacts** (struct): Array containing contacts ids. The parameter is required. Parameter value must be valid contacts ids.

**dnss** (array): Array containing at least 2 DNS strings. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.

**whoisProtection** (bool): Default is false. Set true to buy domain whois protection. Is activated immediatly after register this domain. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showRegistrantInfoOnWhois** (bool): Default is false. Set true to show registrant info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showAdminContactInfoOnWhois** (bool): Default is false. Set true to show admin contact info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**showTechContactInfoOnWhois** (bool): Default is false. Set true to show tech contact info on whois. The parameter is optional. Parameter value must be a boolean ( true or false ).

**paymentMethodType** (string): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo","bono".

**extraData** (struct): Hash array containing additional data. The parameter is optional.

### Response:

type: void

## Billing\_Buy\_Sms

### Description:

Buy a new SMS account.

1.0 -> 1.1: Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

Version: 1.2

### Parameters:

**account** (string): Sms account number. The parameter is required.

**credit** (int): Initial credit. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "50","100","1000","5000","20000".

### Response:

type: void

## Billing\_Transfer\_GetRequirements

### Description:

Gets transfer requirements.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **struct**

Example:

'None'

## Billing\_Transfer\_Domain

**Description:**

Buy and transfer to Dinahosting an existing domain name.

Some domains require an auth code to complete the transfer. There is a command for this:

Api\_Command\_Domain\_Transfer\_SetAuthcode.

1.0 -> 1.1 Improved simulation mode.

1.1 -> 1.2: COMMAND\_BILLING\_CREDIT error.

1.2 -> 1.3: OBJECT\_EXISTS and OBJECT\_NOT\_EXISTS errors.

1.3 -> 1.4: New TLDs are supported.

1.4 -> 1.5: New contact keys (RegistrantIdentification, AdminIdentification, TechIdentification, BillingIdentification) - required for esNic domain validation.

EsNic domain validation for technical contact: must be a natural person

Version: 1.5

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**period** (**int**): Registration period ( years ). Default: 1 year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 10 inclusive.

**contacts** (**struct**): Array containing contacts ( see Domain\_Contacts\_GetAll ). Optional, but many of the tlds require registrant information (see Billing\_Transfer\_GetRequirements ). The parameter is optional. Parameter value must be a valid contacts.

**dnss** (**array**): Array containing DNS strings ( see Domain\_Dnss\_Get ). Optional. The parameter is optional. Parameter value must be a valid array of 2-8 unique DNS.

**whoisProtection** (**bool**): Default is false. Set true to buy domain whois protection. Is activated inmediately after transfer this domain. The parameter is optional. Parameter value must be a boolean ( true or false ).

**paymentMethodType** (**string**): Default is saldo. Set it to "bono" to use that payment Method. The parameter is optional. Accepted parameter values: "saldo", "bono".

**Response:**

type: **void**

## Billing\_Autorenew\_GetAll

**Description:**

Returns renewable services with autorenovation info.

1.0 -> 1.1 Return only renewable services.

1.1 -> 1.2 Added domain parameter to filter results

Version: 1.2

**Parameters:**

**currency** (**string**): Currency. Default: EUR. The parameter is optional. Accepted parameter values: "USD", "EUR".

**showRenovationPeriods** (**bool**): True to add possible renovation periods (months) for each service. The parameter is optional. Parameter value must be a boolean ( true or false ).

**domain** ([string](#)): Domain to filter results The parameter is optional.

**Response:**

type: [array](#)

Example:

(Example too long, see web.)

## Billing\_Autorenew\_SetNoRenewable

**Description:**

Set service as no-renewable.

Service won't autorenew automatically at expiration date, service expiration won't be reported at expiration date, service won't be eligible for renew.

Version: 1.00

**Parameters:**

**id** ([int](#)): Service ID ( as returned by Billing\_Autorenew\_GetAll ). The parameter is required. Parameter value must be a valid integer.

**Response:**

type: [void](#)

## Billing\_Autorenew\_SetOff

**Description:**

Set autorenovation OFF for a service. Service won't autorenew automatically at expiration date.

Version: 1.02

**Parameters:**

**id** ([int](#)): Service ID ( as returned by Billing\_Autorenew\_GetAll ). The parameter is required. Parameter value must be a valid integer.

**Response:**

type: [void](#)

## Billing\_Autorenew\_SetOn

**Description:**

Set autorenovation ON for a service. Service will renew automatically at expiration date.

No payment is done until expiration date.

1.0 -> 1.1: Added some validation error codes for invalid paymentMethodId .

Version: 1.1

**Parameters:**

**id** ([int](#)): Service ID ( as returned by Billing\_Autorenew\_GetAll ). The parameter is required. Parameter value must be a valid integer.

**paymentMethodId** ([int](#)): Payment method ID (as returned by Payment\_Method\_GetAll). The parameter is required. Parameter value must be a valid integer.

**period** ([int](#)): Renovation period ( years or months, see periodUnit parameter ). Optional. The parameter is optional. Parameter value must be a valid integer. Parameter value must be between values 1 and 120 inclusive.

**periodUnit** ([string](#)): Period unit. The parameter is required. Accepted parameter values: "YEAR","MONTH".



**Response:**

 type: **void**

## Billing\_GetConfigurableVpsOptions

**Description:**

Gets configurable options for buying a VPS.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

 type: **struct**

Example:

'None'

## Billing\_Help\_DomainExtraData

**Description:**

Get extraData structure information.

Version: 1.0

**Parameters:**

**action** (**string**): Action that might need extraData. The parameter is required. Accepted parameter values: "create", "transfer".

**tld** (**string**): Domain name TLD. Example: com.es (as is). The parameter is optional.

**Response:**

 type: **struct**

Example:

'None'

## Dedicated commands

### Dedicated\_Services\_GetAll

**Description:**

Returns services status.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('iis' => 'started',
'pop3' => 'started',
'smtp' => 'started',
'ftp' => 'started',
'mysql' => 'started',
'web' => 'started',
)
```

### Dedicated\_Services\_ChangeStatus

**Description:**

Change server service status.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**serviceName** (**string**): Service name. The parameter is required. Accepted parameter values: "web", "mysql", "pop3", "smtp", "ftp", "dinascanner".

**serviceAction** (**string**): Action over service. The parameter is required. Accepted parameter values: "start", "stop", "activate", "deactivate", "restart".

**Response:**

type: **void**

### Dedicated\_GetTransferValue

**Description:**

Returns max data transfer value ( MBs ).

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **int**

Example:

1500

## Dedicated\_Control\_GetApacheProcesses

### Description:

Returns apache active processes information.

Only for linux managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: array

Example:

(Example too long, see web.)

## Dedicated\_Control\_GetSystemStatus

### Description:

Returns ssystem information.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: struct

Example:

(Example too long, see web.)

## Dedicated\_Control\_GetMysqlProcesses

### Description:

Returns mysql active processes information.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: array

Example:

```
array(array('info' => 'show',
'id' => 6,
'host' => 'gdc.dinasever.com:37883',
'ddbb' => 'NULL',
'user' => 'root',
'command' => 'Query',
'time' => 'NULL',
```

```
'status' => 'NULL',
))
```

## Dedicated\_Control\_GetWebserverStatus

### Description:

Returns webserver information.  
 Only for managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: **struct**  
 Example:  
 (Example too long, see web.)

## Dedicated\_Control\_GetCpuLoadByHosting

### Description:

Returns cpu load by hosting information.  
 Only for managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: **array**  
 Example:  

```
array(array('hosting' => 'others',
'used_cpu' => 100,
'percent_used_cpu' => 100,
))
```

## Dedicated\_Control\_GetSystemLoad

### Description:

Returns cpu load by hosting information.  
 Only for managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: **struct**  
 Example:  
 (Example too long, see web.)

## Dedicated\_Control\_KillMysqlProcess

**Description:**

Kill mysql active processes .  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**pid** (**int**): Process id. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: **void**

## Dedicated\_ForVps\_Vps\_Create\_Unmanaged

**Description:**

Generates and returns a ticket for creating an unmanaged VPS in a dedicated server. The ticket status can be consulted with the command '\Dedicated\_ForVps\_GetTicketStatus\'. Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**alias** (**string**): VPS alias. The parameter is optional.  
**description** (**string**): VPS description. The parameter is optional.  
**diskImage** (**string**): Image disk. Use the command 'Dedicated\_ForVps\_Vps\_GetTypes' to get a list of available images. The parameter is required.  
**vcpusCount** (**int**): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.  
**ram** (**int**): RAM (MB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.  
**diskSpace** (**int**): Disk space (GB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 10.  
**transferLimit** (**int**): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available transfer limit. The parameter is optional.

**Response:**

type: **int**

Example:

'None'

## Dedicated\_ForVps\_Vps\_Create\_Managed

**Description:**

Generates and returns a ticket for creating a managed VPS in a dedicated server. The ticket status can be consulted with the command '\Dedicated\_ForVps\_GetTicketStatus\'. Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**alias** (**string**): VPS alias. The parameter is optional.  
**description** (**string**): VPS description. The parameter is optional.

**diskImage** (**string**): Image disk. Use the command 'Dedicated\_ForVps\_Vps\_GetTypes' to get a list of available images. The parameter is required.

**vcpusCount** (**int**): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.

**ram** (**int**): RAM (MB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.

**diskSpace** (**int**): Disk space (GB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 10.

**transferLimit** (**int**): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available transfer limit. The parameter is optional.

**Response:**

type: **int**

Example:

'None'

## Dedicated\_ForVps\_Vps\_GetAll

**Description:**

It returns the VPS list of the dedicated server.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

'None'

## Dedicated\_ForVps\_Vps\_GetTypes

**Description:**

Returns the types of VPS, based on the type of administration and disk image.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

'None'

## Dedicated\_ForVps\_Vps\_Activate

**Description:**

Activates a VPS in a dedicated server.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.  
**vpsHostname** (*string*): VPS hostname. The parameter is required.

**Response:**  
 type: *void*

## Dedicated\_ForVps\_Vps\_Delete

**Description:**  
 Generates and returns a ticket for deleting a VPS in a dedicated server.  
 The ticket status can be consulted with the command '\Dedicated\_ForVps\_GetTicketStatus'.  
 Version: 1.0

**Parameters:**  
**serverName** (*string*): Server name. The parameter is required.  
**vpsHostname** (*string*): VPS hostname. The parameter is required.

**Response:**  
 type: *void*

## Dedicated\_ForVps\_Vps\_Deactivate

**Description:**  
 Desactivates a VPS in a dedicated server.  
 Version: 1.0

**Parameters:**  
**serverName** (*string*): Server name. The parameter is required.  
**vpsHostname** (*string*): VPS hostname. The parameter is required.

**Response:**  
 type: *void*

## Dedicated\_ForVps\_Vps\_Update

**Description:**  
 Generates and returns a ticket for updating a VPS in a dedicated server. The ticket status can be consulted with the command '\Dedicated\_ForVps\_GetTicketStatus'.  
 Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field.  
 Version: 1.0

**Parameters:**  
**serverName** (*string*): Server name. The parameter is required.  
**vpsHostname** (*string*): VPS hostname. The parameter is required.  
**alias** (*string*): VPS alias. The parameter is optional.  
**description** (*string*): VPS description. The parameter is optional.  
**vcpusCount** (*int*): Number of vcpus. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 4 inclusive.  
**ram** (*int*): RAM (MB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available RAM. The parameter is required. Parameter value must be a valid integer. Parameter value must be greater or equal than 512.  
**diskSpace** (*int*): Disk space (GB). Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available space. The parameter is required. Parameter value must be a valid integer. Parameter value must

be greater or equal than 10.

**transferLimit** (`int`): Transfer limit (GB). Set empty for unlimited. Use the command 'Dedicated\_ForVps\_GetServerInfo' to check the available transfer limit. The parameter is optional.

**Response:**

type: `int`

Example:

'None'

## Dedicated\_ForVps\_GetTicketStatus

**Description:**

Returns the status information about creating, updating and modifying VPS in a dedicated server.

Version: 1.0

**Parameters:**

**serverName** (`string`): Server name. The parameter is required.

**ticketId** (`int`): Ticket ID. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: `array`

Example:

'None'

## Dedicated\_ForVps\_GetServerInfo

**Description:**

Returns the information of a dedicated server.

Version: 1.0

**Parameters:**

**serverName** (`string`): Server name. The parameter is required.

**Response:**

type: `struct`

Example:

'None'

## Dedicated\_Kvm\_Get

**Description:**

Returns KVM configuration.

Only for non-managed dedicated servers.

1.0 -> 1.1: new keys: user, password

Version: 1.1

**Parameters:**

**serverName** (`string`): Server name The parameter is required.

**Response:**

type: `struct`

Example:

`array('user' => '6pri',`



```
'password' => 'yeqryr8',
'ip' => '10.10.10.10',
)
```

## Dedicated\_Kvm\_Unmount

### Description:

Unmount KVM.

Only for non-managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (**string**): Server name The parameter is required.

**allowed\_ip** (**string**): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

### Response:

type: **void**

## Dedicated\_Kvm\_Mount

### Description:

Mount KVM and get public IP.

Only for non-managed dedicated servers.

1.0 -> 1.1: returns mounted IP

1.1 -> 1.2: new parameter: allowed\_ip

Version: 1.2

### Parameters:

**serverName** (**string**): Server name The parameter is required.

**allowed\_ip** (**string**): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

### Response:

type: **string**

Example:

'None'

## Dedicated\_GetLicenses

### Description:

Returns licenses information.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**

Example:

```
array('totalLicenses' => 25,
'availableLicenses' => 19,
```

```
'usedLicenses' => 6,
)
```

## Dedicated\_Monitorization\_Get

### Description:

Returns monitorization configuration.  
 Only for non-managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**  
 Example:  

```
array('emails' => array(""),
'activated' => false,
)
```

## Dedicated\_Monitorization\_Activate

### Description:

Activate monitorization.  
 Only for non-managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.  
**emails** (**array**): Array of emails. The parameter is required.

### Response:

type: **void**

## Dedicated\_Monitorization\_Deactivate

### Description:

Deactivate monitorization.  
 Only for non-managed dedicated servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **void**

## Dedicated\_GetServerInfo

### Description:

Returns dedicated server info.  
 Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_Get

**Description:**

Returns an user custom hosting setup.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**setupName** (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_GetAll

**Description:**

Returns all available hosting setups.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_GetOptions

**Description:**

Returns all available hosting setup options.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_GetDhSetups

### Description:

Returns all DH default hosting setups.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_Create

### Description:

Creates a new user custom hosting setup.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

**setupName** (**string**): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**options** (**struct**): Setup options ( as returned by Dedicated\_HostingSetup\_Get ). The parameter is required.

### Response:

type: **void**

## Dedicated\_HostingSetup\_GetCustomSetups

### Description:

Returns all user custom hosting setups.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**

Example:

(Example too long, see web.)

## Dedicated\_HostingSetup\_Delete

**Description:**

Deletes a existing user custom hosting setup.  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**setupName** (**string**): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: **void**

## Dedicated\_HostingSetup\_Update

**Description:**

Updates an existing user custom hosting setup.  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**setupName** (**string**): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.  
**options** (**struct**): New setup options ( as returned by Dedicated\_HostingSetup\_Get ). The parameter is required.

**Response:**

type: **void**

## Dedicated\_GetMonitorizedPorts

**Description:**

Returns monitorized ports with their status.  
 Only for unmanaged servers.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Dedicated\_PowerStatus\_Restart

**Description:**

Restart server.  
 Only for non-managed dedicated servers.  
 Version: 1.0

**Parameters:**

`serverName` (`string`): Server name. The parameter is required.

**Response:**

type: `void`

## Dedicated\_GetDataTransferInfo

**Description:**

Returns data transfer summary.

Version: 1.0

**Parameters:**

`serverName` (`string`): Server name. The parameter is required.

**Response:**

type: `struct`

Example:

(Example too long, see web.)

## Dedicated\_Nfs\_Get

**Description:**

Returns NFS server name

Version: 1.0

**Parameters:**

`serverName` (`string`): Server name The parameter is required.

**Response:**

type: `struct`

Example:

'None'

## Dedicated\_MailReport\_Set

**Description:**

Set consumption email report.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

`serverName` (`string`): Server name. The parameter is required.

`emailTo` (`string`): Destination emails separated with commas. The parameter is required. Parameter value must be a valid email address.

`periodicity` (`string`): Report periodicity. The parameter is required. Accepted parameter values: "daily", "weekly", "monthly".

**Response:**

type: `void`

## Dedicated\_MailReport\_Get

**Description:**

Returns consumption email report status.  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

`serverName` (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('periodicity' => 'never',
'emailTo' => "",
)
```

## Dedicated\_MailReport\_SetOff

**Description:**

Deactivate consumption email report.  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

`serverName` (**string**): Server name. The parameter is required.

**Response:**

type: **void**

## Dedicated\_GetUsedSpace

**Description:**

Returns used space.  
 Only for managed dedicated servers.  
 Version: 1.0

**Parameters:**

`serverName` (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('total_space' => 12,
'space_limit' => 0,
'homes_space' => 0,
'web_space' => 12.287109375,
'webmail_space' => 0,
'bds_space' => 0.095703125,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)
```

## Dedicated\_Hosting\_GetAll

**Description:**

Returns existing hostings on dedicated.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**onlyActiveHostings** (bool): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean ( true or false ).

**Response:**

type: array

Example:

```
array('probando2.shiai.es','probandohosting.com','aaaaaaaa.es','bbbbbb.com','154fdsa1f.com','pruebahostin
gdw27.com')
```

## Dedicated\_Hosting\_GetSpace

**Description:**

Returns used hosting space.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**domainName** (string): Hosting domain name. The parameter is required.

**Response:**

type: struct

Example:

```
array('total_space' => 2,
'space_limit' => 2000,
'homes_space' => 0,
'web_space' => 1.537109375,
'webmail_space' => 0,
'bds_space' => 0.0478515625,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)
```

## Dedicated\_Hosting\_Activate

**Description:**

Activates an existing but inactive hosting.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**domainName** (string): Hosting domain name. The parameter is required.

**Response:**

type: void



## Dedicated\_Hosting\_Create

### Description:

Creates a new hosting.

Only for managed dedicated servers.

1.0 -> 1.1: Added an existing setup name validation.

Version: 1.1

### Parameters:

**serverName** (*string*): Server name. The parameter is required.

**domainName** (*string*): Hosting domain name. The parameter is required.

**setupName** (*string*): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**setZones** (*bool*): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean ( true or false ).

### Response:

type: *void*

## Dedicated\_Hosting\_Delete

### Description:

Deletes an existing hosting.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (*string*): Server name. The parameter is required.

**domainName** (*string*): Hosting domain name. The parameter is required.

### Response:

type: *void*

## Dedicated\_Hosting\_Deactivate

### Description:

Deactivates an existing and active hosting.

Only for managed dedicated servers.

Version: 1.0

### Parameters:

**serverName** (*string*): Server name. The parameter is required.

**domainName** (*string*): Hosting domain name. The parameter is required.

### Response:

type: *void*

## Dedicated\_Hosting\_GetSetupName

### Description:

Returns hosting setup name.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**domainName** (*string*): Hosting domain name. The parameter is required.

**Response:**

type: *string*

Example:

'None'

## Dedicated\_Hosting\_ChangeSetup

**Description:**

Changes setup of a existing hosting.

Only for managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**domainName** (*string*): Hosting domain name. The parameter is required.

**setupName** (*string*): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: *void*

## Dedicated\_GetBandwidthConsumption

**Description:**

Returns bandwidth consumption summary.

Only for non-managed dedicated servers.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**month** (*int*): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.

**year** (*int*): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2012 inclusive.

**Response:**

type: *struct*

Example:

(Example too long, see web.)

## Dedicated\_GetSpaceAndTransferReportPerHosting

**Description:**

Returns a summary of the space and transfer data per hosting.

Version: 1.0

**Parameters:**

`serverName` (string): Server name. The parameter is required.

**Response:**

type: `struct`

Example:

'None'

## Domain commands

### Domain\_Redirection\_Set301

**Description:**

Sets 301 URL Redirecion on whole domain or domain zone.

1.0 -> 1.1: Optionally now can add a subdomain redirection.

Version: 1.1

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**url** (string): URL redirection destination. The parameter is required. Parameter value must be a valid URL.

**host** (string): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

**Response:**

type: void

### Domain\_Redirection\_Get

**Description:**

Returns current URL redirection on whole domain, if any.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: struct

Example:

```
array('Type' => 'FRAME',
'Url' => 'http://www.dinahosting.com',
'Title' => "",
'Description' => "",
'Keywords' => "",
)
```

### Domain\_Redirection\_GetAll

**Description:**

Returns all URL redirections on whole domain and domain zone redirections, if any.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: array

Example:

(Example too long, see web.)

## Domain\_Redirection\_SetFrame

### Description:

Sets Frame URL Redirecion on whole domain.

Version: 1.1

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**url** (*string*): URL redirection framed destination. The parameter is required. Parameter value must be a valid URL.

**title** (*string*): Frame title. The parameter is optional. Parameter length must be between 0 and 255 characters.

**description** (*string*): Frame description meta. The parameter is optional. Parameter length must be between 0 and 255 characters.

**keywords** (*string*): Frame keywords meta. The parameter is optional. Parameter length must be between 0 and 255 characters.

**favicon** (*string*): Frame favicon URL. The parameter is optional. Parameter length must be between 0 and 255 characters.

**viewport** (*string*): Frame viewport meta. The parameter is optional. Parameter length must be between 0 and 255 characters.

### Response:

type: *void*

## Domain\_Redirection\_SetStd

### Description:

Sets Standard URL Redirecion on whole domain or domain zone.

1.0 -> 1.1: Optionally now can add a subdomain redirection.

Version: 1.1

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**url** (*string*): URL redirection destination. The parameter is required. Parameter value must be a valid URL.

**host** (*string*): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

### Response:

type: *void*

## Domain\_Redirection\_Delete

### Description:

Delete URL redirection over whole domain.

1.0 -> 1.1: Optionally now can remove any subdomain redirections.

1.1-> 1.2: Optionally now can remove specific subdomain redirections.

Version: 1.2

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**removeAll** (*bool*): Set true for deleting any subdomain redirection zones. The parameter is optional. Parameter value must be a boolean ( true or false ).

**subdomain** (*?string*): Subdomain (restricted: www, mail, ftp, mysql, sql, autoconfig, autodiscover, \*, @). The parameter is optional.

**Response:**

 type: `void`

## Domain\_Redirection\_SetFrameSimple

**Description:**

Sets Frame Redirection on whole domain or domain zone.

Version: 1.1

**Parameters:**
**domain** (`string`): Domain name. The parameter is required.

**url** (`string`): URL redirection destination. The parameter is required. Parameter value must be a valid URL.

**host** (`string`): Subdomain hostname. The parameter is optional. Parameter length must be between 0 and 64 characters.

**Response:**

 type: `void`

## Domain\_CheckForTransfer

**Description:**

Check if a domain name is available for transfer to Dinahosting.

Returns true if domain is registered but not in Dinahosting.

This doesn't guarantee a successful transfer.

Version: 1.1

**Parameters:**
**domain** (`string`): Domain name. The parameter is required.

**Response:**

 type: `bool`

Example:

*false*

## Domain\_GetRenovationDate

**Description:**

Returns expected domain deactivation date.

Version: 1.0

**Parameters:**
**domain** (`string`): Domain name. The parameter is required.

**Response:**

 type: `string`

Example:

*'2013-07-06'*

## Domain\_GetRegistrationDate

**Description:**

Returns registration date of domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: string

Example:

'2003-02-10'

## Domain\_GetAuthcode

**Description:**

Returns authcode of domain for outgoing transfer.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: string

Example:

'BqDCrmqJ788}\*~'

## Domain\_Contacts\_Set

**Description:**

Sets new contacts of domain. Only updates one type of contacts ( example: billing ).

Version: 1.1

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**contactType** (string): Contact type (registrant, admin, billing or technical ). The parameter is required.

Accepted parameter values: "registrant","admin","billing","tech".

**contacts** (struct): Array containing contacts ( see Domain\_Contacts\_Get for format ). The parameter is required. Parameter value must be a valid contacts.

**Response:**

type: void

## Domain\_Contacts\_Get

**Description:**

Returns array with contacts of domain. Only returns contacts of one type ( example: registrant ).

1.0 -> 1.1: Added Fax key.

Version: 1.1

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**contactType** (string): Contact type (registrant, admin, billing or technical ). The parameter is required.

Accepted parameter values: "registrant","admin","billing","tech".

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Domain\_Contacts\_GetRegistrantVerificationInfo

**Description:**

Gets registrant verification info.

Version: 1.0

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**Response:**

type: **struct**

Example:

'None'

## Domain\_Contacts\_GetAll

**Description:**

Returns array with contacts of domain.

1.0 -> 1.1: Added Fax key.

Version: 1.1

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Domain\_Contacts\_SetAll

**Description:**

Sets new contacts of domain.

Version: 1.1

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**contacts** (**struct**): Array containing contacts ( see Domain\_Contacts\_GetAll ). The parameter is required.

Parameter value must be a valid contacts.

**Response:**

type: **void**

## Domain\_WhoisPrivacy\_Set

**Description:**



Configures domain whois privacy.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**redirectionEmail** (*string*): Email direction where received emails are sent to. The parameter is optional. Parameter value must be a valid email address.

**enable** (*boolean*): Enables or disables whois protection. The parameter is optional. Parameter value must be a boolean ( true or false ).

**Response:**

type: *void*

## Domain\_WhoisPrivacy\_Get

**Description:**

Returns domain whois privacy configuration.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *struct*

Example:

```
array('activated' => true,
'redirection' => 'informedk_org562@privacidadwhois.com',
)
```

## Domain\_UrlShortener\_GetAll

**Description:**

Returns shortened URLs on a domain.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain. The parameter is required.

**Response:**

type: *struct*

Example:

*'None'*

## Domain\_UrlShortener\_Activate

**Description:**

Activates shortener on a domain.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain. The parameter is required.

**Response:**

 type: `void`

## Domain\_UrlShortener\_Create

**Description:**

Creates a new shortened URL on a domain.

Use with caution: shortened URLs on a domain make traffic being redirected to the shortener.

Version: 1.0

**Parameters:**

`domain` (`string`): Domain. The parameter is required.

`url` (`string`): URL to be shorten. The parameter is required. Parameter value must be a valid URL.

`title` (`string`): URL title. The parameter is optional.

**Response:**

 type: `void`

## Domain\_UrlShortener\_Delete

**Description:**

Deletes a shortened URL on a domain

Version: 1.0

**Parameters:**

`domain` (`string`): Domain. The parameter is required.

`keyword` (`string`): Keyword of the shortened URL to be deleted. The parameter is required.

**Response:**

 type: `void`

## Domain\_UrlShortener\_Deactivate

**Description:**

Deactivates shortener on a domain.

Use with caution: it deletes all the shortened URLs created under the domain.

Version: 1.0

**Parameters:**

`domain` (`string`): Domain. The parameter is required.

**Response:**

 type: `void`

## Domain\_PanelIndependentAccess\_SetPassword

**Description:**

Set a new password for domain panel independent access.

Version: 1.1

**Parameters:**

**domain** ([string](#)): Domain name. The parameter is required.  
**newPassword** ([string](#)): New password. The parameter is required.

**Response:**  
 type: [void](#)

## Domain\_GetInfoDomain

**Description:**  
 Returns registry info of domain.  
 Version: 1.0

**Parameters:**  
**domain** ([string](#)): Domain name. The parameter is required.

**Response:**  
 type: [struct](#)  
 Example:  
 (Example too long, see web.)

## Domain\_Zone\_AddTypeSRV

**Description:**  
 Adds a new type SRV zone ( service zone ).  
 This command doesn't work with .tel domains.  
 1.0 -> 1.1: Improved parameter validation and documentation.  
 Version: 1.1

**Parameters:**  
**domain** ([string](#)): Domain name. The parameter is required.  
**service** ([string](#)): Service. The parameter is required. Accepted parameter values: "http", "ftp", "scp", "https", "tftp", "smtp", "imap", "pop3", "imaps", "pop3s", "named", "ldap", "ssh", "dhcp", "radius", "sip", "sippederationtls", "xmpp-server", "xmpp-client", "jabber", "autodiscover", "sipinternaltls".  
**protocol** ([string](#)): Protocol. The parameter is required. Accepted parameter values: "tcp", "udp", "tls".  
**priority** ([int](#)): Priority (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.  
**weight** ([int](#)): Weight (integer value) used when same priority. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.  
**port** ([int](#)): Destination port number (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 65535 inclusive.  
**destination** ([string](#)): Destination domain . The parameter is required.

**Response:**  
 type: [void](#)

## Domain\_Zone\_GetTypeURL

**Description:**  
 Returns domain zones.  
 Version: 1.0

**Parameters:**  
**domain** ([string](#)): Domain name. The parameter is required.

**Response:**

 type: **array**

Example:

```
array(array('destinationUrl' => 'http://www.example.com/test',
'hostname' => 'test',
))
```

## Domain\_Zone\_DeleteTypeAname

**Description:**

Deletes a domain ANAME type zone.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**Response:**

 type: **void**

## Domain\_Zone\_AddTypeFrame

**Description:**

Adds a new type FRAME zone ( frame redirection ).

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**destinationUrl** (**string**): Destination URL. The parameter is required. Parameter value must be a valid URL.

**Response:**

 type: **void**

## Domain\_Zone\_DeleteTypeTXT

**Description:**

Deletes a domain TXT type zone.

Version: 1.1

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**value** (**string**): Value (optional). For specifying what to delete if there are several zones under the same hostname. The parameter is optional.

**Response:**

 type: **void**

## Domain\_Zone\_GetAll

### Description:

Returns domain zones.

Version: 1.0

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**orderBy** (*string*): Ordenation types: no ordenation, by type, by hostname. The parameter is optional.

Accepted parameter values: "none", "type", "hostname".

### Response:

type: *array*

Example:

(Example too long, see web.)

## Domain\_Zone\_UpdateTypeAname

### Description:

Update an existing type ANAME zone.

Version: 1.0

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**destinationHostname** (*string*): Destination hostname. The parameter is required.

**oldDestinationHostname** (*string*): Old destination hostname. The parameter is required.

### Response:

type: *void*

## Domain\_Zone\_UpdateTypeA

### Description:

Update an existing type A zone.

This command doesn't work with .tel domains.

Version: 1.0

### Parameters:

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (*string*): New IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters. Parameter value must be a valid IP v4 address.

**oldIp** (*string*): Old IP v4 Address. The parameter is optional.

### Response:

type: *void*

## Domain\_Zone\_DeleteTypeSRV

### Description:

Deletes a domain SRV type zone.

1.0 -> 1.1: Major bugfix.

Version: 1.1

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**service** (*string*): Service. The parameter is required. Accepted parameter values: "http", "ftp", "scp", "https", "ftpd", "smtp", "imap", "pop3", "imaps", "pop3s", "named", "ldap", "ssh", "dhcp", "radius", "sip", "sippederationtls", "xmpp-server", "xmpp-client", "jabber", "autodiscover", "sipinternaltls".

**protocol** (*string*): Protocol. The parameter is required. Accepted parameter values: "tcp", "udp", "tls".

**priority** (*int*): Priority (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.

**weight** (*int*): Weight (integer value) used when same priority. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 100 inclusive.

**port** (*int*): Destination port number (integer value). The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 0 and 65535 inclusive.

**destination** (*string*): Destination domain . The parameter is required.

**Response:**

 type: *void*

## Domain\_Zone\_AddTypeMXS

**Description:**

Adds a new type MX simple zone.

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (*string*): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters. Parameter value must be a valid IP v4 address.

**Response:**

 type: *void*

## Domain\_Zone\_DeleteTypeSPF

**Description:**

Deletes a domain SPF type zone.

Version: 1.1

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**value** (*string*): Value (optional). For specifying what to delete if there are several zones under the same hostname. The parameter is optional.

**Response:**

 type: *void*

## Domain\_Zone\_AddTypeA

**Description:**

Adds a new type A zone.  
 This command doesn't work with .tel domains.  
 Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.  
**hostname** (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.  
**ip** (string): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters. Parameter value must be a valid IP v4 address.

**Response:**

type: void

## Domain\_Zone\_GetTypeMX

**Description:**

Returns domain zones.  
 Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: array

Example:

```
array(array('host' => 'SOA',
'type' => 'MX',
'address' => 'mx4.mepegotolafiesta.com.',
))
```

## Domain\_Zone\_AddTypeURL301

**Description:**

Adds a new type URL zone ( standard redirection ).  
 This command doesn't work with .tel domains.  
 Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.  
**hostname** (string): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.  
**destinationUrl** (string): Destination URL. The parameter is required. Parameter value must be a valid URL.

**Response:**

type: void

## Domain\_Zone\_GetTypeAname

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *array*

Example:

```
array(array('destinationHostname' => 'google.com',
'hostname' => '@',
))
```

## Domain\_Zone\_DeleteTypeAAAA

**Description:**

Deletes a domain AAAA type zone.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (*string*): IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters. Parameter value must be a valid IP v6 address.

**Response:**

type: *void*

## Domain\_Zone\_GetTypeFrame

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *array*

Example:

```
array(array('destinationUrl' => 'http://www.dinahosting.com',
'hostname' => '@',
),array('destinationUrl' => 'http://www.dinahosting.com',
'hostname' => 'www',
))
```

## Domain\_Zone\_AddTypeAname

**Description:**

Adds a new type ANAME zone to the root domain.

Version: 1.0



**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**destinationHostname** (*string*): Destination hostname. The parameter is required.

**Response:**

type: *void*

## Domain\_Zone\_UpdateTypeAAAA

**Description:**

Update an existing type AAAA zone.

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (*string*): New IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters. Parameter value must be a valid IP v6 address.

**oldIp** (*string*): Old IP v6 Address. The parameter is optional.

**Response:**

type: *void*

## Domain\_Zone\_UrlShortener\_GetAll

**Description:**

Returns shortened URLs on a subdomain.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain. The parameter is required.

**Response:**

type: *struct*

Example:

'None'

## Domain\_Zone\_UrlShortener\_Activate

**Description:**

Activates shortener on a subdomain.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain. The parameter is required.

**hostname** (*string*): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**Response:**

type: *void*

## Domain\_Zone\_UrlShortener\_Create

### Description:

Creates a new shortened URL on a subdomain.  
Version: 1.0

### Parameters:

**domain** (string): Domain. The parameter is required.  
**url** (string): URL to be shorten. The parameter is required. Parameter value must be a valid URL.  
**title** (string): URL title. The parameter is optional.

### Response:

type: void

## Domain\_Zone\_UrlShortener\_Delete

### Description:

Deletes a shortened URL on a subdomain  
Version: 1.0

### Parameters:

**domain** (string): Domain. The parameter is required.  
**keyword** (string): Keyword of the shortened URL to be deleted. The parameter is required.

### Response:

type: void

## Domain\_Zone\_UrlShortener\_Deactivate

### Description:

Deactivates shortener on a subdomain.  
Use with caution: it deletes all the shortened URLs created under the subdomain.  
Version: 1.0

### Parameters:

**domain** (string): Domain. The parameter is required.

### Response:

type: void

## Domain\_Zone\_AddTypeMX

### Description:

Adds a new type MX zone.  
This command doesn't work with .tel domains.  
The MX zone requires that a A zone already exists.  
Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.  
**host** (string): Host (empty for default host: @). Required for ip values. The parameter is optional. Parameter length must be between 0 and 64 characters.

**priority** (**int**): Priority. The parameter is required. Parameter value must be a valid integer. Accepted parameter values: "1", "5", "10", "15", "20", "25", "30", "40", "50".

**value** (**string**): Value (servername or ip). The parameter is required.

**Response:**

type: **void**

## Domain\_Zone\_DeleteTypeFrame

**Description:**

Deletes a domain FRAME type zone.

Version: 1.0

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional. Parameter length must be between 0 and 64 characters.

**Response:**

type: **void**

## Domain\_Zone\_AddTypeTXT

**Description:**

Adds a new type TXT zone .

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**text** (**string**): Text. The parameter is required.

**Response:**

type: **void**

## Domain\_Zone\_DeleteAll

**Description:**

Deletes domain zones of a given type.

Version: 1.0

**Parameters:**

**domain** (**string**): Domain name. The parameter is required.

**type** (**string**): Zone type. The parameter is required. Accepted parameter values: "A", "AAAA", "CNAME", "frame", "URL", "URL\_301", "TXT", "SRV", "SPF", "MX".

**Response:**

type: **void**

## Domain\_Zone\_DeleteTypeCname

**Description:**

Deletes a domain CNAME type zone.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**hostname** (string): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**Response:**

type: void

## Domain\_Zone\_DeleteTypeMX

**Description:**

Deletes a domain MX type zone.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**host** (string): Host. The parameter is required.

**address** (string): Address. The parameter is required. Parameter length must be between 0 and 64 characters.

**Response:**

type: void

## Domain\_Zone\_GetTypeSPF

**Description:**

Returns domain zones.

*This command has pending some modifications: Make response data format more friendly.*

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: array

Example:

```
array(array('address' => 'no|||',
))
```

## Domain\_Zone\_GetTypeA

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

 type: **array**

Example:

(Example too long, see web.)

## Domain\_Zone\_Delete

**Description:**

Deletes a domain zone ( any type except A, AAAA and SRV ).

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**Response:**

 type: **void**

## Domain\_Zone\_AddTypeMXDetailedEmail

**Description:**

Adds a new type MX zone.

This command doesn't work with .tel domains.

The MX zone requires that a A zone already exists.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**mxNumber** (**int**): Mx number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 7 inclusive.

**hostname** (**string**): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**Response:**

 type: **void**

## Domain\_Zone\_DeleteTypeA

**Description:**

Deletes a domain A type zone.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (**string**): IP v4 Address. The parameter is required. Parameter length must be between 0 and 15 characters. Parameter value must be a valid IP v4 address.

**Response:**

 type: **void**

## Domain\_Zone\_DeleteTypeURL301

### Description:

Deletes a domain URL 301 type zone.

Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.

**hostname** (string): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional.

Parameter length must be between 0 and 64 characters.

### Response:

type: void

## Domain\_Zone\_GetTypeCname

### Description:

Returns domain zones.

Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.

### Response:

type: array

Example:

```
array(array('destinationHostname' => 'google.com',
'hostname' => 'preproduccion',
))
```

## Domain\_Zone\_GetTypeTXT

### Description:

Returns domain zones.

Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.

### Response:

type: array

Example:

```
array(array('host' => 'key_9344._domainkey',
'text' => 'Address',
))
```

## Domain\_Zone\_DeleteTypeURL

### Description:

Deletes a domain URL type zone.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Hostname. Set empty for deleting an entire domain redirect. The parameter is optional. Parameter length must be between 0 and 64 characters.

**Response:**

type: *void*

## Domain\_Zone\_AddTypeCname

**Description:**

Adds a new type CNAME zone.

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**destinationHostname** (*string*): Destination hostname. The parameter is required.

**Response:**

type: *void*

## Domain\_Zone\_GetTypeURL301

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *array*

Example:

```
array(array('destinationUrl' => 'http://aaa.es',
'hostname' => 'ns',
))
```

## Domain\_Zone\_AddTypeURL

**Description:**

Adds a new type URL zone ( standard redirection ).

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**hostname** (*string*): Origin hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**destinationUrl** (*string*): Destination URL. The parameter is required. Parameter value must be a valid URL.

**Response:**

 type: **void**

## Domain\_Zone\_AddTypeAAAA

**Description:**

Adds a new type AAAA zone.

This command doesn't work with .tel domains.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**hostname** (**string**): Hostname. The parameter is required. Parameter length must be between 0 and 64 characters.

**ip** (**string**): IP v6 Address. The parameter is required. Parameter length must be between 0 and 39 characters. Parameter value must be a valid IP v6 address.

**Response:**

 type: **void**

## Domain\_Zone\_GetTypeSRV

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**
**domain** (**string**): Domain name. The parameter is required.

**Response:**

 type: **array**

Example:

```
array(array('service' => 'http',
'protocol' => 'tcp',
'priority' => 10,
'weight' => 70,
'port' => 8080,
'destination' => 'comooor.es',
),array('service' => 'http',
'protocol' => 'tcp',
'priority' => 10,
'weight' => 30,
'port' => 80,
'destination' => 'comor.es',
))
```

## Domain\_Zone\_GetTypeAAAA

**Description:**

Returns domain zones.

Version: 1.0

**Parameters:**



**domain** (string): Domain name. The parameter is required.

**Response:**

type: array

Example:

```
array(array('ip' => '2001:470:0:76:0:0:0:2',
'hostname' => 'trescuatro',
))
```

## Domain\_Dnss\_Set

**Description:**

Sets new DNS of domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**dnss** (array): Array containing DNS strings. The parameter is required. Parameter value must be a valid array of 2-8 unique DNS.

**Response:**

type: void

## Domain\_Dnss\_Get

**Description:**

Returns array with DNS of domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: struct

Example:

```
array('Dns1' => 'ns2.gestiondecuenta.com',
'Dns2' => 'ns.gestiondecuenta.com',
)
```

## Domain\_Status\_Get

**Description:**

Returns status of domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: string

Example:

```
'clientDeleteProhibited, clientTransferProhibited'
```

## Domain\_Status\_IsLocked

**Description:**

This commands checks domain status and returns boolean true if domain is currently locked.  
Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **bool**

Example:

*true*

## Domain\_Status\_Unlock

**Description:**

Unlocks domain status.  
Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **void**

## Domain\_Status\_Lock

**Description:**

Locks domain status.  
Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **void**

## Domain\_Parking\_Set

**Description:**

Sets domain on parking.  
Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**title** (string): Title for parking webpage. The parameter is optional. Parameter length must be between 0 and 254 characters.

**description** (string): Description for parking webpage. The parameter is optional.

**Response:**

type: **void**

## Domain\_Parking\_Get

**Description:**

Returns parking info.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('Title' => 'Mi página en parking',  
'Description' => 'Estas es la web de dinahosting',  
)
```

## Domain\_Transfer\_GetStatus

**Description:**

Returns status of incoming domain transfer.

On transfer codes: 'WAIT\_INIT','WAIT\_APROB','WAIT\_AUTH','WAIT\_TRANSFER'

Error codes: 'KO\_AUTH','FAILED','BLOCKED','REJECTED','CANCELED','DELETED','UNKNOWN'

OK codes: 'ACTIVE'

1.0 -> 1.1 Transfer codes and simulation mode.

Version: 1.1

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **string**

Example:

'None'

## Domain\_Transfer\_GetAuthcode

**Description:**

Returns authcode used for incoming domain transfer.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **string**

Example:

'None'

## Domain\_Transfer\_SetAuthcode

**Description:**

Set new authcode for incoming domain transfer.

Changes won't take effect until a new transfer petition takes place.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**newAuthcode** (*string*): New authcode. The parameter is required.

**Response:**

type: *void*

## Domain\_Transfer\_Restart

**Description:**

Restarts a domain transfer.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *void*

## Domain\_Transfer\_Cancel

**Description:**

Cancels a domain transfer.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *void*

## Domain\_GetExpirationDate

**Description:**

Returns registry expiration date of domain.

Version: 1.0

**Parameters:**

**domain** (*string*): Domain name. The parameter is required.

**Response:**

type: *string*

Example:

'2014-02-10'

## Domain\_Test

**Description:**

Domain command Testing

Version: 1.0

**Parameters:**

**domain** (string): Domain name, for testing purposes. The parameter is required.

**optional** (string): An optional parameter. The parameter is optional.

**Response:**

type: **struct**

Example:

*'None'*

## Domain\_CheckForRegister

**Description:**

Check if a domain name is available for register on Dinahosting.

Returns true if domain is not registered and domain is not currently in activation process on Dinahosting.

Version: 1.1

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **bool**

Example:

*false*

## Domain\_NameServer\_Modify

**Description:**

Modify existing DNS server over domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**hostname** (string): Hostname. The parameter is required.

**ip** (string): IP Address. The parameter is required. Parameter value must be a valid IP address ( v4 or v6 ).

Keep in mind that v6 is currently experimental, might not work.

**Response:**

type: **void**

## Domain\_NameServer\_GetAll

**Description:**

Returns array with registered DNS server names and IPs over domain.

Version: 1.0

**Parameters:**

**domain** (string): Domain name. The parameter is required.

**Response:**

type: **array**

Example:

```
array(array('name' => 'ns2.afiliacionweb.com',
'ip' => '85.112.29.233',
),array('name' => 'ns1.afiliacionweb.com',
'ip' => '85.112.29.232',
))
```

## Domain\_NameServer\_Register

### Description:

Register a DNS server over domain.

Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.

**hostname** (string): Hostname. The parameter is required.

**ip** (string): IP Address. The parameter is required. Parameter value must be a valid IP address ( v4 or v6 ).

Keep in mind that v6 is currently experimental, might not work.

### Response:

type: void

## Domain\_NameServer\_Delete

### Description:

Delete a DNS server over domain.

Version: 1.0

### Parameters:

**domain** (string): Domain name. The parameter is required.

**hostname** (string): Hostname. The parameter is required.

### Response:

type: void

# DomainVerification commands

## DomainVerification\_GetPending

**Description:**

Returns all pending domain verifications.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Hosting commands

### Hosting\_Dsn\_Access\_GetAll

**Description:**

Returns all MS Access database source names.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: array

Example:

(Example too long, see web.)

### Hosting\_Dsn\_Access\_Create

**Description:**

Creates a new Access Data Source Name.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**ddbName** (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**dsnName** (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

**dsnDescription** (string): Database description. The parameter is required.

**Response:**

type: void

### Hosting\_Dsn\_Access\_Delete

**Description:**

Delete selected Access Database Source Name.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**dsnName** (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

**Response:**

type: void



## Hosting\_Dsn\_Sqlserver\_GetAll

### Description:

Returns all SqlServer database source names.  
Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: array

Example:

(Example too long, see web.)

## Hosting\_Dsn\_Sqlserver\_Create

### Description:

Creates a new Sqlserver Data Source Name.  
Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**ddbName** (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**dsnName** (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

**dsnDescription** (string): Database description. The parameter is required.

### Response:

type: void

## Hosting\_Dsn\_Sqlserver\_Delete

### Description:

Delete selected Sqlserver Database Source Name.  
Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**dsnName** (string): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

### Response:

type: void

## Hosting\_Dsn\_Mysql\_GetAll

### Description:

Returns all Mysql database source names.  
Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Hosting\_Dsn\_Mysql\_Create

**Description:**

Creates a new Mysql Data Source Name.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**dsnName** (**string**): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

**dsnDescription** (**string**): Database description. The parameter is required.

**Response:**

type: **void**

## Hosting\_Dsn\_Mysql\_Delete

**Description:**

Delete selected Mysql Database Source Name.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**dsnName** (**string**): DSN name. The parameter is required. Parameter value must be a valid database server name (dsn). Parameter length must be between 1 and 30 characters, only alphanumeric, underscore, dot or dash.

**Response:**

type: **void**

## Hosting\_Php\_DeactivatePhpExecutionOnHtml

**Description:**

Forbids PHP execution on HTML.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Php\_ActivatePhpExecutionOnHtml

**Description:**

Allows PHP execution on HTML.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Php\_SetVersion

**Description:**

Changes hosting PHP version.

Version: 1.2 Version parameter allows any content. Availability validation of PHP version added.

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**version** (**string**): Check available versions with Hosting\_GetInfo command. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Php\_OverwritePhpIniWithDefaults

**Description:**

Overwrite php.ini values with defaults.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Php\_PhpExecutionOnHtmlIsActive

**Description:**

Checks if hosting can execute PHP code on HTML.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **bool**

Example:  
'None'

## Hosting\_Php\_OverwritePhpIni

**Description:**  
Overwrite php.ini values with custom text.  
Version: 1.1

**Parameters:**  
**hosting** (string): Hosting name. The parameter is required.  
**text** (string): New text contents for php.ini file. The parameter is required.

**Response:**  
type: void

## Hosting\_IsActive

**Description:**  
Returns if a hosting is active.  
Version: 1.0

**Parameters:**  
**hosting** (string): Hosting name. The parameter is required.

**Response:**  
type: bool  
Example:  
'None'

## Hosting\_GetMailInfo

**Description:**  
Returns hosting mailserver information.  
Version: 1.0

**Parameters:**  
**hosting** (string): Hosting name. The parameter is required.

**Response:**  
type: struct  
Example:  

```
array('server' => 'v1193.dinserver.com',  
'host' => 'mail.test-edu-personalizacion-avanzada.com',  
'TLS_activable' => true,  
'TLS_is_active' => false,  
)
```

## Hosting\_Ftp\_Get

**Description:**  
Get FTP access status.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **bool**

Example:

*true*

## Hosting\_Ftp\_Account\_GetAll

**Description:**

Returns all existing FTP accounts.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array(array('account' => 'test-linux-personal',
'folder' => "",
'date' => '2010-03-17 12:51:02',
'main' => true,
))
```

## Hosting\_Ftp\_Account\_SetPassword

**Description:**

Changes FTP account password.

1.0 -> 1.1: Minor improvements.

1.1 -> 1.2: Improved password validation.

1.2 -> 1.3: Turned into a delayed execution command.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.3

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**password** (string): New account password. The parameter is required. Parameter length must be between 6 and 71 characters. Parameter length must be at least 6 characters. It can contain only alphanumeric characters. String must contain at least one number and one character. Unsafe patterns as 111111 or 123456 aren't allowed.

**Response:**

type: **int**

Example:

*'None'*

## Hosting\_Ftp\_Account\_CmdStatus\_GetCreate

**Description:**

Get delayed Hosting\_Ftp\_Account\_Create execution information.

Version: 1.0

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**orderId** (int): Command order id. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: array

Example:

'None'

## Hosting\_Ftp\_Account\_CmdStatus\_ListAll

**Description:**

List all imports

Version: 1.0

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**Response:**

type: array

Example:

'None'

## Hosting\_Ftp\_Account\_CmdStatus\_GetSetPassword

**Description:**

Get delayed Hosting\_Ftp\_Account\_SetPassword execution information.

Version: 1.0

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**orderId** (int): Command order id. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: array

Example:

'None'

## Hosting\_Ftp\_Account\_CmdStatus\_GetDelete

**Description:**

Get delayed Hosting\_Ftp\_Account\_Delete execution information.

Version: 1.0

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**orderId** (int): Command order id. The parameter is required. Parameter value must be a valid integer.

**Response:**

 type: `array`

Example:

'None'

## Hosting\_Ftp\_Account\_Create

**Description:**

Creates a new FTP account.

1.0 -&gt; 1.1: Minor improvements.

1.1 -&gt; 1.2: Folder parameter is now optional (may be left empty or missing).

1.2 -&gt; 1.3: Turned into a delayed execution command.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.3

**Parameters:**
**hosting** (`string`): Hosting name. The parameter is required.

**account** (`string`): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**password** (`string`): Account password. The parameter is required. Parameter length must be between 6 and 71 characters.

**folder** (`string`): Remote folder. If missing or empty, the FTP account will use a default folder (like /www/). The parameter is optional. Parameter length must be between 0 and 150 characters. Parameter length must be between 1 and 255 characters. It should be a valid unix path.

**Response:**

 type: `int`

Example:

'None'

## Hosting\_Ftp\_Account\_Delete

**Description:**

Deletes a FTP account.

1.0 -&gt; 1.1: Minor improvements.

1.1 -&gt; 1.2: Turned into a delayed execution command.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.2

**Parameters:**
**hosting** (`string`): Hosting name. The parameter is required.

**account** (`string`): FTP account name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

 type: `int`

Example:

'None'

## Hosting\_Ftp\_Activate

**Description:**

Activate FTP access.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_Ftp\_Deactivate

**Description:**

Deactivate FTP access.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_Ftp\_GetAccountsLimit

**Description:**

Get FTP account number limit for this hosting.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: struct

Example:

```
array('moreAccountsAllowed' => false,  
'accountLimitNumber' => 0,  
)
```

## Hosting\_Ssh\_Get

**Description:**

Get SSH access status.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: bool

Example:

```
false
```



## Hosting\_Ssh\_Activate

**Description:**

Activate SSH access.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_Ssh\_Deactivate

**Description:**

Deactivate SSH access.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_GetInfo

**Description:**

Returns hosting webserver information with traffic and space metrics.  
Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: struct

Example:

(Example too long, see web.)

## Hosting\_UrlShortener\_GetAll

**Description:**

Returns shortened URLs on a hosting.  
Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: struct

Example:

'None'

## Hosting\_UrlShortener\_Activate

**Description:**

Activates shortener on a hosting.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**folder** (string): Hosting folder to store shorten URLs. The parameter is required.

**Response:**

type: void

## Hosting\_UrlShortener\_Create

**Description:**

Creates a new shortened URL on a hosting.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**url** (string): URL to be shorten. The parameter is required. Parameter value must be a valid URL.

**title** (string): URL title. The parameter is optional.

**Response:**

type: void

## Hosting\_UrlShortener\_Delete

**Description:**

Deletes a shortened URL on a hosting

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**keyword** (string): Keyword of the shortened URL to be deleted. The parameter is required.

**Response:**

type: void

## Hosting\_UrlShortener\_Deactivate

**Description:**

Deactivates shortener on a hosting.

Use with caution: it deletes all the shortened URLs created under the hosting.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_TlsMail\_Get

### Description:

Get TLS mail activation status.  
 True means active, false means inactive.  
 Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: **bool**  
 Example:  
*'None'*

## Hosting\_TlsMail\_Activate

### Description:

Activate TLS over mail.  
 Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: **void**

## Hosting\_TlsMail\_Deactivate

### Description:

Deactivate TLS over mail.  
 Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: **void**

## Hosting\_Https\_Get

### Description:

Get HTTPS access status.  
 Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: **bool**  
 Example:  
*false*

## Hosting\_Https\_Activate

**Description:**

Activate HTTPs access.  
Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_Https\_Deactivate

**Description:**

Deactivate HTTPs access.  
Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_PanelIndependentAccess\_SetPassword

**Description:**

Set a new password for hosting panel independent access.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.  
**newPassword** (string): New password. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_UsedSpace\_Get

**Description:**

Get the used space by email accounts.  
Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_UsedSpace\_GetPerAlias

**Description:**

Get the used space by an email account.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars.

String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Antivirus\_UpdateAttachedFileFilter

**Description:**

Update antivirus file filters configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**tag** (string): Tag. The parameter is required.

**action** (string): Action on filtered emails. The parameter is required. Accepted parameter values: "allow", "deny", "scan".

**Response:**

type: void

## Hosting\_Email\_Antivirus\_GetConfig

**Description:**

Returns antivirus configuration data.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: struct

Example:

```
array('delete_mesage' => true,
'label_subject' => false,
'active' => true,
)
```

## Hosting\_Email\_Antivirus\_GetAttachedFilesFilters

**Description:**

Returns antivirus file filters configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Hosting\_Email\_Antivirus\_SetConfig

**Description:**

Updates antivirus configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**action** (string): Action over infected emails. The parameter is required. Accepted parameter values: "mark", "delete", "none".

**Response:**

type: **void**

## Hosting\_Email\_Redirection\_ActivateCatchAll

**Description:**

Activate catch-all email redirection.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**destination** (array): Array of destination email accounts. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_Redirection\_Get

**Description:**

Returns an email account configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Redirection alias ( email account without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: **struct**

Example:

```
array('alias' => 'dfsdf',
'catch-all' => false,
'date' => '2009-07-22 11:30:40',
'destination' => array('pla@olaa.com'),
```

)

## Hosting\_Email\_Redirection\_GetAll

### Description:

Returns all email account redirections.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: array

Example:

```
array(array('alias' => 'dfsdf',
'catch-all' => false,
'date' => '2009-07-22 11:30:40',
'active' => true,
'destination' => array('pla@olaa.com'),
))
```

## Hosting\_Email\_Redirection\_DeactivateCatchAll

### Description:

Deactivate catch-all email redirection.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: void

## Hosting\_Email\_Redirection\_Create

### Description:

Creates a new email redirection on a new alias.

If already exists a redirection alias or an email account over this alias name, this command will fail.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Redirection alias ( email account without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**destination** (array): Array of destination email accounts. The parameter is required.

### Response:

type: void

## Hosting\_Email\_Redirection\_Delete

**Description:**

Delete an existing email redirection.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing redirection alias ( email account without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Redirection\_Update

**Description:**

Update an existing email redirection.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing redirection alias ( email account without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**destination** (array): Array of destination email accounts. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_Antispam\_GetWhitelist

**Description:**

Returns spam assasin white list configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: array

Example:

```
array(array('id' => 4503,
'action' => 'allow',
'domain' => 'ola@ola.com',
))
```

## Hosting\_Email\_Antispam\_AddDomainToWhitelist

**Description:**

Add a domain to whitelist.

Version: 1.1



**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**domain** (string): Domain added to whitelist. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_Antispam\_AddDomainToBlacklist

**Description:**

Add a domain to blacklist.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**domain** (string): Domain added to blacklist. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_Antispam\_RemoveDomainFromWhitelist

**Description:**

Remove a domain from antispam whitelist.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**domain** (string): Domain removed from whitelist. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_Antispam\_GetScore

**Description:**

Returns spam assasin aggresivity score.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: int

Example:

6

## Hosting\_Email\_Antispam\_GetBlacklist

**Description:**

Returns spam assasin black list configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **array**

Example:

```
array(array('id' => 12743,
'action' => 'reject',
'domain' => 'dadasdadsad.com',
),array('id' => 12745,
'action' => 'reject',
'domain' => 'alaaaaa@funciona.com',
))
```

## Hosting\_Email\_Antispam\_GetConfig

**Description:**

Returns spam assasin configuration data.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('delete_mesage' => false,
'label_subject' => true,
'active' => true,
)
```

## Hosting\_Email\_Antispam\_Activate

**Description:**

Activate spam assasin.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_Antispam\_SetScore

**Description:**

Updates spam assasin aggresivity score.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**score** (**int**): New aggressivity score. The parameter is required. Parameter value must be between values 1 and 10 inclusive.

**Response:**

type: **void**

## Hosting\_Email\_Antispam\_RemoveDomainFromBlacklist

**Description:**

Remove a domain from antispam blacklist.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**domain** (**string**): Domain removed from blacklist. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_Antispam\_Deactivate

**Description:**

Deactivate spam assasin.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_Antispam\_SetConfig

**Description:**

Updates spam assasin configuration.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**action** (**string**): Action over detected spam emails. The parameter is required. Accepted parameter values: "mark", "delete", "none".

**list** (**array**): Whitelist. The parameter is optional.

**Response:**

type: **void**

## Hosting\_Email\_Dkim\_Get

**Description:**

Get DKIM TXT zone (hostname and text).

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('hostname' => 'default._domainkey',  
'text' => 'v=DKIM1; g=*; k=rsa; p=',  
)
```

## Hosting\_Email\_Dkim\_IsActive

**Description:**

Get DKIM activation status.

True means active, false means inactive.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **bool**

Example:

*false*

## Hosting\_Email\_Dkim\_Activate

**Description:**

Activate TLS over mail.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_Dkim\_Deactivate

**Description:**

Deactivate TLS over mail.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **void**

## Hosting\_Email\_AutomaticDeletion\_DeleteRule

**Description:**

Deletes automatic deletion rule for one email account.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: **void**

## Hosting\_Email\_AutomaticDeletion\_CreateRule

**Description:**

Creates a new automatic deletion rule for one email account.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Account alias (email account without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**scope** (string): All messages or those that have been read. The parameter is required. Accepted parameter values: "all", "read".

**antique** (string): Antique messages. The parameter is required. Accepted parameter values: "week", "fortnight", "month", "quarter-year", "half-year", "year", "two-year", "three-year", "four-year", "five-year".

**ifExceedsSize** (string): If used space exceeds indicated size. The parameter is required. Accepted parameter values: "any", "50MB", "100MB", "200MB", "500MB", "1GB".

**Response:**

type: **void**

## Hosting\_Email\_Autoresponse\_Get

**Description:**

Returns an existing email autoresponse configuration.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing email account alias ( without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: **struct**

Example:

```
array('alias' => 'acura',
'date' => '2007-03-14 00:24:51',
'text' => 'Prueba con fechas',
'from' => '2009-04-20',
'to' => '2009-04-27',
'activated' => true,
```

```
'excluded-emails' => ",
)
```

## Hosting\_Email\_Autoresponse\_Activate

### Description:

Activates an existing email autoresponse.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing email account alias ( without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

### Response:

type: void

## Hosting\_Email\_Autoresponse\_Create

### Description:

Creates a new email autoresponse.

1.1 -> 1.2: Parameter "excluded" is optional.

Version: 1.2

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing email account alias ( without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**text** (string): Autoresponse text. The parameter is required.

**excluded** (array): Exluded emails. The parameter is optional.

**start** (string): Start datetime. Format: yyyy-MM-dd hh:mm:ss. If empty, autoresponse starts now. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**end** (string): End datetime. Format: yyyy-MM-dd hh:mm:ss. If empty, autoresponse doesn't expire. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**copy** (array): Copy emails. The parameter is optional.

### Response:

type: void

## Hosting\_Email\_Autoresponse\_Delete

### Description:

Deletes an existing email autoresponse.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**alias** (string): Existing email account alias ( without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

 type: **void**

## Hosting\_Email\_Autoresponse\_Deactivate

**Description:**

Deactivates an existing email autoresponse.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**alias** (**string**): Existing email account alias ( without hostname ). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

 type: **void**

## Hosting\_Email\_Account\_Empty

**Description:**

Empties an existing email account.

**Warning: This process can't be reversed.**

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**account** (**string**): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

 type: **void**

## Hosting\_Email\_Account\_Get

**Description:**

Returns an existing email account.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**account** (**string**): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

 type: **struct**

Example:

```
array('alias' => 'test-linux-personal',
'account' => 'test-linux-personal@test-linux-personal.dinahosting.com',
'password' => '????????',
'catch-all' => false,
```

```
'auto_response' => false,
'date' => '2010-03-17 12:51:02',
'id' => 0,
'disabled' => false,
)
```

## Hosting\_Email\_Account\_GetAll

### Description:

Returns all email accounts.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: **struct**

Example:

```
array(array('alias' => 'test-linux-personal',
'account' => 'test-linux-personal@test-linux-personal.dinahosting.com',
'catch-all' => false,
'auto_response' => false,
'date' => '2010-03-17 12:51:02',
'id' => 0,
'disabled' => false,
))
```

## Hosting\_Email\_Account\_SetPassword

### Description:

Changes email account password.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**password** (string): New account password. The parameter is required. Password length must be between 8 and 35 characters. It can contain only alphanumeric, underscore or dash chars. String must start with a alphanumeric char. String must not contain email alias. String must not contain more than 2 consecutive repeating characters. String must not contain more than 2 consecutive numbers in any order.

### Response:

type: **void**

## Hosting\_Email\_Account\_RemoveCopyTo

### Description:

Remove copy-to for an email account.

Version: 1.0

### Parameters:



**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Account\_Activate

**Description:**

Activates an existing but inactive email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Account\_Create

**Description:**

Creates a new email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**password** (string): Account password. The parameter is required. Password length must be between 8 and 35 characters. It can contain only alphanumeric, underscore or dash chars. String must start with a alphanumeric char. String must not contain email alias. String must not contain more than 2 consecutive repeating characters. String must not contain more than 2 consecutive numbers in any order.

**Response:**

type: void

## Hosting\_Email\_Account\_SetCopyTo

**Description:**

Sets copy-to emails for an email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**destination** (array): Array of destination email accounts. The parameter is required.

**Response:**

type: void

## Hosting\_Email\_Account\_Delete

**Description:**

Deletes an existing email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Account\_GetCopyTo

**Description:**

Returns copy-to emails for an email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: struct

Example:

```
array('czxccxz@dinahosting.com','asffsasf@dinahosting.com')
```

## Hosting\_Email\_Account\_Deactivate

**Description:**

Deactivates an existing email account.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**account** (string): Email account alias (without hostname). The parameter is required. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**Response:**

type: void

## Hosting\_Email\_Filter\_SetGeolocalization

### Description:

Changes geolocalization status. If active, country filters (OriginCountryCode) will take effect on received emails.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**active** (bool): True active . The parameter is required. Parameter value must be a boolean ( true or false ).

### Response:

type: void

## Hosting\_Email\_Filter\_GetAll

### Description:

Return email filter rule list.

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**country** (string): Country. The parameter is optional.

### Response:

type: array

Example:

```
array(array('action' => 'mark',
'date' => '2009-07-22 09:25:06',
'ruleId' => 258105,
'searchSubject' => 'EmailSubject',
'searchObject' => 'nvbnbv',
'markLabel' => 'vbn',
))
```

## Hosting\_Email\_Filter\_GetGeolocalization

### Description:

Returns geolocalization status ( active true, inactive false ).

Version: 1.1

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

### Response:

type: bool

Example:

false

## Hosting\_Email\_Filter\_CreateForSpam

### Description:

Create a new email filter rule for spam.

Version: 1.1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**action** (string): Action on filtered emails. The parameter is required. Accepted parameter values: "reject", "moveToSpamFolder", "moveToFolder", "redirectToEmailAccount".

**actionDestiny** (string): Value associated with action. Only required by actions "moveToFolder" and "redirectToEmailAccount". The parameter is optional.

**Response:**

type: void

## Hosting\_Email\_Filter\_Create

**Description:**

Create a new email filter rule.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**searchObject** (string): String that must match for filter email (do action). The parameter is required.

**searchSubject** (string): Search subject. The parameter is required. Accepted parameter values: "EmailSubject", "EmailFrom", "EmailTo", "EmailHeaders", "OriginCountryCode".

**action** (string): Action on filtered emails. The parameter is required. Accepted parameter values: "mark", "delete".

**markLabel** (string): Label for filtered emails if action equals 'mark'. The parameter is optional.

**Response:**

type: void

## Hosting\_Email\_Filter\_Delete

**Description:**

Delete a existing email filter rule.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**ruleId** (int): Rule ID. The parameter is required.

**Response:**

type: void

## Hosting\_Certificate\_Uninstall

**Description:**

Uninstalls the certificate of the hosting

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Certificate\_DeactivateLetsEncrypt

**Description:**

Deactivates Let's encrypt certificate on the hosting

Version: 1.0

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Certificate\_Install

**Description:**

Installs certificate on the hosting

Version: 1.0

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**cert** (**string**): Certificate. The parameter is required.

**key** (**string**): Key. The parameter is required.

**certIntermediate** (**string**): Certificate intermediate. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Certificate\_ActivateLetsEncrypt

**Description:**

Activates Let's encrypt certificate on the hosting

Version: 1.0

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**subdomains** (**string**): Subdomains separated by commas. The parameter is optional.

**Response:**

 type: **void**

## Hosting\_Ddbb\_Access\_GetAllMdb

**Description:**

Returns all Access .mdb databases .

**Warning: This won't work with accdb databases (access 2007+).**

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: array

Example:

```
array('1ruebaRomina.mdb','1ruebaRomina2.mdb','1ruebaRuben.mdb','1ruebaRuben2.mdb')
```

## Hosting\_Ddbb\_GetAll

**Description:**

Returns all databases.

Version: 1.0

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: array

Example:

```
array(array('name' => 'kbsoporte',
'account' => 'kbsoporte',
'date' => '2012-11-05 20:59:35',
'autoinstall' => false,
'type' => 'mysql',
'access_host' => 'any',
))
```

## Hosting\_Ddbb\_MariaDb\_GetAll

**Description:**

Returns all MariaDb databases.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: array

Example:

```
'None'
```

## Hosting\_Ddbb\_MariaDb\_SetPassword

**Description:**

Set new password for selected MariaDb Database.

Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**ddbName** (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (string): User name. The parameter is required. Parameter length must be between 1 and 16

characters.

**userPassword** (*string*): New password. The parameter is required.

**Response:**

type: *void*

## Hosting\_Ddbb\_MariaDb\_Create

**Description:**

Creates a new MariaDb DataBase.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (*string*): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

**userPassword** (*string*): User password. The parameter is required. Parameter length must be between 6 and 71 characters.

**Response:**

type: *void*

## Hosting\_Ddbb\_MariaDb\_Delete

**Description:**

Delete selected MariaDb Database.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (*string*): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

**Response:**

type: *void*

## Hosting\_Ddbb\_Sqlserver\_GetAll

**Description:**

Returns all SqlServer databases.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**Response:**

type: *array*

Example:

```
array(array('name' => 'unabdd',
'account' => 'unusuario',
'password' => '????????',
'date' => '2011-07-13 17:41:03',
'data_space' => 3,
'transaction_space' => 1,
))
```

## Hosting\_Ddbb\_Sqlserver\_SetPassword

### Description:

Set new password for selected Sqlserver Database.

Version: 1.1

### Parameters:

**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**password** (**string**): New password. The parameter is required.

### Response:

type: **void**

## Hosting\_Ddbb\_Sqlserver\_SetTransactionsSize

### Description:

Set transaction size for selected Sqlserver Database.

Version: 1.1

### Parameters:

**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**transactionsSize** (**int**): Maximum data volume transmitted ( MB ). The parameter is required. Parameter value must be greater than 0.

### Response:

type: **void**

## Hosting\_Ddbb\_Sqlserver\_SetDataSize

### Description:

Set data size for selected Sqlserver Database.

Version: 1.1

### Parameters:

**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.



**dataSize** (**int**): Maximum data volume stored ( MB ). The parameter is required. Parameter value must be greater than 2.

**Response:**

type: **void**

## Hosting\_Ddbb\_Sqlserver\_GetGlobalSpaceInfo

**Description:**

It gets the space info of SQL Server formatted in MB: space limit and available space.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('limit' => 200,
'used' => 0,
'reserved' => 0,
'available' => 200,
)
```

## Hosting\_Ddbb\_Sqlserver\_Create

**Description:**

Creates a new Sqlserver DataBase.

Version: 1.1

**Parameters:**

**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (**string**): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

**userPassword** (**string**): User password. The parameter is required. Parameter length must be between 6 and 71 characters. Parameter value must be an alphanumeric string.

**dataSize** (**int**): Maximum data volume stored ( MB ). The parameter is required. Parameter value must be greater than 4.

**transactionsSize** (**int**): Maximum data volume transmitted ( MB ). The parameter is required. Parameter value must be greater than 0.

**accessFrom** (**string**): Access from: Any location (default), Localhost and specific IP (XX.XX.XX.XX), Only localhost (localhost). The parameter is optional.

**Response:**

type: **void**

## Hosting\_Ddbb\_Sqlserver\_Delete

**Description:**

Delete selected Sqlserver Database.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

type: *void*

## Hosting\_Ddbb\_Sqlserver\_GetSpaceInfo

**Description:**

Returns a sqlserver databases space report.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**Response:**

type: *struct*

Example:

(Example too long, see web.)

## Hosting\_Ddbb\_MongoDb\_GetAll

**Description:**

Returns all MongoDB databases.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**Response:**

type: *array*

Example:

'None'

## Hosting\_Ddbb\_MongoDb\_SetPassword

**Description:**

Set new password for selected MongoDB Database.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (*string*): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

**userPassword** (*string*): New password. The parameter is required.

**Response:**

 type: **void**

## Hosting\_Ddbb\_MongoDb\_Delete

**Description:**

Delete selected MongoDB Database.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

 type: **void**

## Hosting\_Ddbb\_Postgre\_GetAll

**Description:**

Returns all Postgre databases.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**Response:**

 type: **array**

Example:

```
array(array('server' => 'pgsql01.dinserver.com',
'account' => 'userpostgresiphone',
'name' => 'postgreiphone',
'autoinstall' => false,
'date' => '2011-01-31 14:25:08',
))
```

## Hosting\_Ddbb\_Postgre\_SetPassword

**Description:**

Set new password for selected Postgre Database.

Version: 1.1

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**ddbName** (**string**): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (**string**): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

**userPassword** (**string**): New password. The parameter is required.

**Response:**

 type: `void`

## Hosting\_Ddbb\_Postgre\_Create

**Description:**

Creates a new Postgre DataBase.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

`ddbName` (`string`): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

`userName` (`string`): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

`userPassword` (`string`): User password. The parameter is required. Parameter length must be between 6 and 71 characters.

**Response:**

 type: `void`

## Hosting\_Ddbb\_Postgre\_Delete

**Description:**

Delete selected Postgre Database.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

`ddbName` (`string`): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

 type: `void`

## Hosting\_Ddbb\_Mysql\_CreateWithIpAcl

**Description:**

Creates a new Mysql DataBase with IP ACL.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

`ddbName` (`string`): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

`userName` (`string`): User name. The parameter is required. Parameter length must be between 1 and 16 characters.

`userPassword` (`string`): User password. The parameter is required. Parameter length must be between 6 and 16 characters.

`accessHostIp` (`string`): Accepted IP v4 . The parameter is required. Parameter value must be a valid IP v4

address.

**Response:**

type: `void`

## Hosting\_Ddbb\_Mysql\_GetAll

**Description:**

Returns all Mysql databases.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

**Response:**

type: `array`

Example:

(Example too long, see web.)

## Hosting\_Ddbb\_Mysql\_SetPassword

**Description:**

Set new password for selected Mysql Database.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

`ddbName` (`string`): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

`password` (`string`): New password. The parameter is required.

**Response:**

type: `void`

## Hosting\_Ddbb\_Mysql\_GetAclConfiguration

**Description:**

Returns selected Mysql Database ACL configuration.

Version: 1.1

**Parameters:**

`hosting` (`string`): Hosting name. The parameter is required.

`ddbName` (`string`): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

type: `struct`

Example:

```
array('host' => 'ANY',
)
```

## Hosting\_Ddbb\_Mysql\_GetInfo

### Description:

Returns selected Mysql Database information.

Version: 1.1

### Parameters:

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

### Response:

type: *struct*

Example:

```
array('server' => 'hl77.dinserver.com',
'name' => 'test2BBDD',
'account' => 'test2',
'date' => '2009-08-26 22:14:15',
'autoinstall' => false,
)
```

## Hosting\_Ddbb\_Mysql\_SetHostAcl

### Description:

Sets selected Mysql Database hostname ACL configuration.

Version: 1.1

### Parameters:

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**hostname** (*string*): Accepted hostname or "any" string (without quotes). The parameter is required.

### Response:

type: *void*

## Hosting\_Ddbb\_Mysql\_BackupTask\_Set

### Description:

Sets the Backup Task for a selected Mysql dataBase.

Version: 1.1

### Parameters:

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**taskPeriod** (*string*): Task Period ( ##d for days, ##h for hours or ##m for minutes ). The parameter is required. Parameter value must be a valid task period ( ##d ##m ##h ).

### Response:

type: *void*

## Hosting\_Ddbb\_Mysql\_BackupTask\_GetAll

**Description:**

Returns all Mysql backup tasks on this hosting.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('oscommerce_47aa503883f33' => '1m',  
)
```

## Hosting\_Ddbb\_Mysql\_BackupTask\_Delete

**Description:**

Deletes an existing Backup Task for a selected Mysql dataBase.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**ddbName** (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

type: **void**

## Hosting\_Ddbb\_Mysql\_SetIpAcl

**Description:**

Sets selected Mysql Database hostname ACL configuration.  
Version: 1.1

**Parameters:**

**hosting** (string): Hosting name. The parameter is required.

**ddbName** (string): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**ip** (string): Accepted IP v4 . The parameter is required. Parameter value must be a valid IP v4 address.

**Response:**

type: **void**

## Hosting\_Ddbb\_Mysql\_Create

**Description:**

Creates a new Mysql DataBase.  
Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**userName** (*string*): User name. The parameter is required. Parameter length must be between 6 and 16 characters.

**userPassword** (*string*): User password. The parameter is required. Parameter length must be between 1 and 41 characters.

**accessHost** (*string*): Access host. Default: any. The parameter is required. Accepted parameter values: "any", "localhost".

**Response:**

type: *void*

## Hosting\_Ddbb\_Mysql\_Delete

**Description:**

Delete selected Mysql Database.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

type: *void*

## Hosting\_Ddbb\_Mysql\_Backup\_Get

**Description:**

Return an existing Backup Task for a selected Mysql dataBase.

Version: 1.1

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**Response:**

type: *struct*

Example:

(Example too long, see web.)

## Hosting\_Ddbb\_Mysql\_Backup\_Restore

**Description:**

Restore an existing Backup Task for a selected Mysql dataBase.

Version: 1.1



**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**ddbName** (*string*): Database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**file** (*string*): Backup file The parameter is required.

**Response:**

type: *void*

## Hosting\_Import\_List

**Description:**

List unfinished (PENDING or STARTED) imports.

Version: 1.0

**Parameters:**

**hosting** (*string*): Destination hosting name. The parameter is required.

**Response:**

type: *array*

Example:

'None'

## Hosting\_Import\_ApplyForEmailAccountmap

**Description:**

Apply for an email import through IMAP to your hosting.

1.0 -> 1.1: Removed unused parameter.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.1

**Parameters:**

**hosting** (*string*): Destination hosting name. The parameter is required.

**originEmailAddress** (*string*): Origin full email address. The parameter is required. Parameter value must be a valid email address.

**originPassword** (*string*): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.

**originServer** (*string*): Origin server hostname or IP with (or without) port. Default port: 143. The parameter is required.

**create** (*bool*): Create destination email account. The parameter is optional. Parameter value must be a boolean ( true or false ).

**destinationUser** (*string*): Destination new email account alias (without hostname). If it's empty, we'll use originEmailAddress account alias as the destination name. The parameter is optional. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**destinationPassword** (*string*): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.

**notificationMail** (*string*): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

**Response:**

type: *int*

Example:

'None'

## Hosting\_Import\_ListAll

### Description:

List all imports  
Version: 1.0

### Parameters:

**hosting** (string): Destination hosting name. The parameter is required.

### Response:

type: array

Example:

'None'

## Hosting\_Import\_ApplyForDdbb

### Description:

Apply for a database import to your hosting.

1.0 -> 1.1: Removed unused parameter.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.1

### Parameters:

**hosting** (string): Destination hosting name. The parameter is required.

**originDdbbName** (string): Origin database name. The parameter is required. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**originUser** (string): Origin user name. The parameter is required. Parameter length must be between 1 and 16 characters.

**originPassword** (string): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.

**originServer** (string): Origin server hostname or IP with (or without) port. Default port: 3306. The parameter is required.

**create** (bool): Create destination DDBB. The parameter is optional. Parameter value must be a boolean (true or false).

**destinationDdbbName** (string): Destination new database name. If it's empty, we'll use originDdbbName as the destination database name. The parameter is optional. Parameter value must be a valid database name. Parameter length must be between 1 and 64 characters. It can contain only alphanumeric or underscore chars. String must start with a alphabetical char and contain at least one numeric char.

**destinationUser** (string): Destination new user name. If it's empty, we'll use originUser as the destination user name on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.

**destinationPassword** (string): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.

**notificationMail** (string): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

### Response:

type: int

Example:

'None'

## Hosting\_Import\_ApplyForEmailAccountPop

**Description:**

Apply for an email import through POP to your hosting.

1.0 -> 1.1: Removed unused parameter.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.1

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**originEmailAddress** (string): Origin full email address. The parameter is required. Parameter value must be a valid email address.

**originPassword** (string): Origin user password. The parameter is required. Parameter length must be between 1 and 255 characters.

**originServer** (string): Origin server hostname or IP with (or without) port. Default port: 110. The parameter is required.

**create** (bool): Create destination account. The parameter is optional. Parameter value must be a boolean ( true or false ).

**destinationUser** (string): Destination new email account alias (without hostname). If it's empty, we'll use originEmailAddress account alias as the destination name. The parameter is optional. Parameter length must be between 1 and 40 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**destinationPassword** (string): Only used if create is true. Destination user password. If it's empty, we'll use originPassword as the destination password on creation. The parameter is optional. Parameter length must be between 0 and 16 characters.

**destinationFolder** (string): Destination folder outside inbox. The parameter is optional. Parameter length must be between 0 and 255 characters.

**notificationMail** (string): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

**Response:**

type: int

Example:

'None'

## Hosting\_Import\_ApplyForFtp

**Description:**

Apply for a database import to your hosting.

1.0 -> 1.1: Removed unused parameter.

1.1 -> 1.2: Parameters "create" and "destinationPassword" are now deprecated.

Please note this command has a delayed execution and it'll return SUCCESS\_PENDING code on success and a ticket id (integer) on the data field. Version: 1.2

**Parameters:**

**hosting** (string): Destination hosting name. The parameter is required.

**originUser** (string): Origin user name. The parameter is required. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

**originPassword** (string): Origin user password. The parameter is required. Parameter length must be between 3 and 35 characters.

**originServer** (string): Origin server hostname. Port is 21. The parameter is required.

**originDirectory** (string): Destination directory. The parameter is optional. Parameter length must be between 1 and 255 characters. It should be a valid unix path.

**create** (bool): Deprecated parameter. The parameter is optional. Parameter value must be a boolean ( true or false ).

**destinationUser** (string): Destination new user name. If it's empty, we'll use originUser as the destination user name on creation. The parameter is optional. Parameter length must be between 1 and 35 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric

char.

**destinationPassword** (*string*): Deprecated parameter. The parameter is optional.

**notificationMail** (*string*): Email address for a notification. The parameter is optional. Parameter value must be a valid email address.

**Response:**

type: *int*

Example:

'None'

## Hosting\_GetDaysWithBackups

**Description:**

It returns the days with hosting backups.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**from** (*string*): It gets the days starting from specified date. If empty, the command returns the dates of the current year. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: *struct*

Example:

'None'

## Hosting\_Test

**Description:**

Hosting command Testing

Version: 1.0

**Parameters:**

**hosting** (*string*): Domain name, for testing purposes. The parameter is required.

**optional** (*string*): Just an optional parameter The parameter is optional.

**Response:**

type: *struct*

Example:

'None'

## Hosting\_Subdomain\_GetAll

**Description:**

Returns all existing subdomains on hosting.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting domain name. The parameter is required.

**Response:**

type: *array*

Example:  
 (Example too long, see web.)

## Hosting\_Subdomain\_Create

### Description:

Create a subdomain on hosting.  
 1.0 -> 1.1: Fix on directory path validation.  
 Version: 1.1

### Parameters:

**hosting** (**string**): Hosting domain name. The parameter is required.  
**host** (**string**): Host for subdomain name. The parameter is required. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.  
**directory** (**string**): Subdirectory name for account files. Default value: same as host . The parameter is optional. Parameter length must be between 1 and 255 characters. It should be a valid unix path.  
**ftpAccount** (**string**): FTP account name. Default value: same as host . The parameter is optional. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

### Response:

type: **void**

## Hosting\_Subdomain\_Delete

### Description:

Delete a subdomain on hosting.  
 Version: 1.0

### Parameters:

**hosting** (**string**): Hosting domain name. The parameter is required.  
**host** (**string**): Host for subdomain name. The parameter is required. Parameter length must be between 1 and 50 characters. It can contain only alphanumeric, underscore, dot or dash chars. String must start with a alphanumeric char.

### Response:

type: **void**

## Hosting\_Backup\_GetStatus

### Description:

It returns the status of the restorings or sendings programmed.  
 Version: 1.0

### Parameters:

**hosting** (**string**): Hosting name. The parameter is required.

### Response:

type: **struct**  
 Example:  
 'None'

## Hosting\_Backup\_Database\_GetTables

### Description:

It returns the database tables in a hosting backup.

Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**date** (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**databaseName** (string): Database name. The parameter is required.

**databaseType** (string): Database type. The parameter is required. Accepted parameter values: "sqlServer", "otherType".

### Response:

type: **struct**

Example:

'None'

## Hosting\_Backup\_Restore\_Database\_Tables

### Description:

It restores one or several database tables from a backup.

Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**date** (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**databaseName** (string): Database name. The parameter is required.

**databaseType** (string): Database type. The parameter is required. Accepted parameter values: "sqlServer", "otherType".

**tables** (string): Database tablenamees separated by comma. The parameter is required.

**notificationEmail** (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

### Response:

type: **struct**

Example:

'None'

## Hosting\_Backup\_Restore\_Resource

### Description:

It restores a resource from a hosting backup.

Version: 1.0

### Parameters:

**hosting** (string): Hosting name. The parameter is required.

**date** (string): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**resourcePath** (string): Resource path The parameter is required.

**notificationEmail** (string): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

**Response:**

 type: **struct**

Example:

'None'

## Hosting\_Backup\_Restore\_EmailAccounts

**Description:**

It restores one or several email accounts from backup.

Version: 1.0

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**date** (**string**): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**emailAccounts** (**string**): Email accounts separated by comma. The parameter is required.

**notificationEmail** (**string**): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

**Response:**

 type: **struct**

Example:

'None'

## Hosting\_Backup\_Restore\_Database

**Description:**

It restores a database backup.

Version: 1.0

**Parameters:**
**hosting** (**string**): Hosting name. The parameter is required.

**date** (**string**): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**databaseName** (**string**): Database name. The parameter is required.

**databaseType** (**string**): Database type. The parameter is required. Accepted parameter values: "sqlServer","otherType".

**notificationEmail** (**string**): Email to send the notification of restoring. The parameter is required. Parameter value must be a valid email address.

**Response:**

 type: **struct**

Example:

'None'

## Hosting\_Backup\_GetDatabases

**Description:**

It returns the databases in a hosting backup.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: **struct**

Example:

'None'

## Hosting\_Backup\_GetEmailAccounts

**Description:**

It returns the email accounts in a hosting backup.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: **struct**

Example:

'None'

## Hosting\_Backup\_GetResources

**Description:**

It returns the resources in a hosting backup.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**folderPath** (*string*): If empty, it returns the root folder content. The parameter is optional.

**Response:**

type: **struct**

Example:

'None'

## Hosting\_Backup\_Send\_Database\_Tables

**Description:**

It sends one or several database tables from a backup to a specified email.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**databaseName** (*string*): Database name. The parameter is required.



**databaseType** (*string*): Database type. The parameter is required. Accepted parameter values: "sqlServer", "otherType".

**tables** (*string*): Database tablenamees separated by comma. The parameter is required.

**email** (*string*): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

**Response:**

type: *struct*

Example:

'None'

## Hosting\_Backup\_Send\_Resource

**Description:**

It sends a resource from a hosting backup to a specified email.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**resourcePath** (*string*): Resource path The parameter is required.

**email** (*string*): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

**Response:**

type: *struct*

Example:

'None'

## Hosting\_Backup\_Send\_Database

**Description:**

It sends a database backup to a specified email.

Version: 1.0

**Parameters:**

**hosting** (*string*): Hosting name. The parameter is required.

**date** (*string*): Date of the backup. The parameter is required. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**databaseName** (*string*): Database name. The parameter is required.

**databaseType** (*string*): Database type. The parameter is required. Accepted parameter values: "sqlServer", "otherType".

**email** (*string*): Email to send the backup. The parameter is required. Parameter value must be a valid email address.

**Response:**

type: *struct*

Example:

'None'

## Housing commands

### Housing\_GetTransferValue

**Description:**

Returns max data transfer value ( MBs ).

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **int**

Example:

0

### Housing\_Kvm\_Get

**Description:**

Returns KVM configuration.

1.0 -> 1.1: new keys: user, password

Version: 1.1

**Parameters:**

**serverName** (string): Server name The parameter is required.

**Response:**

type: **struct**

Example:

```
array('user' => 'lhcu',  
'password' => '6olto0e3',  
'ip' => '10.10.10.10',  
)
```

### Housing\_Kvm\_Unmount

**Description:**

Unmount KVM.

Version: 1.0

**Parameters:**

**serverName** (string): Server name The parameter is required.

**allowed\_ip** (string): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

**Response:**

type: **void**

### Housing\_Kvm\_Mount

**Description:**

Mount KVM and get public IP.  
 1.0 -> 1.1: returns mounted IP  
 1.1 -> 1.2: new parameter: allowed\_ip  
 Version: 1.2

**Parameters:**

**serverName** (**string**): Server name The parameter is required.  
**allowed\_ip** (**string**): IP that will be granted access. If it's missing, we'll use the requesting client IP. The parameter is optional. Parameter value must be a valid IP v4 address.

**Response:**

type: **string**  
 Example:  
 'None'

## Housing\_Monitorization\_Get

**Description:**

Returns monitorization configuration.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**  
 Example:  

```
array('emails' => array(""),
'activated' => false,
)
```

## Housing\_Monitorization\_Activate

**Description:**

Activate monitorization.  
 Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**emails** (**array**): Array of emails. The parameter is required.

**Response:**

type: **void**

## Housing\_Monitorization\_Deactivate

**Description:**

Deactivate monitorization.  
 Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**Response:**

type: [void](#)

## Housing\_GetServerInfo

**Description:**

Returns data transfer summary.

Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**Response:**

type: [struct](#)

Example:

```
array('main_ip' => '82.98.148.16',
'ips' => array(),
'url' => "",
)
```

## Housing\_PowerStatus\_Restart

**Description:**

Restart server.

Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**Response:**

type: [void](#)

## Housing\_GetDataTransferInfo

**Description:**

Returns data transfer summary.

Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**Response:**

type: [struct](#)

Example:

(Example too long, see web.)

## Housing\_GetBandwidthConsumption

**Description:**

Returns bandwidth consumption summary.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**month** (**int**): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.

**year** (**int**): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2012 inclusive.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Import commands

### Import\_GetStatus

**Description:**

Get import status for a known ticketId

Possible status strings: 'PENDING', 'STARTED', 'OK', 'ERROR', 'CANCEL'

Version: 1.0

**Parameters:**

**importId** (**string**): Import ID The parameter is required. Parameter value must be a valid integer.

**Response:**

type: **string**

Example:

'OK'

# Multihosting commands

## Multihosting\_GetLicenses

**Description:**

Returns licenses information.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('totalLicenses' => 5,  
'availableLicenses' => 4,  
'usedLicenses' => 1,  
)
```

## Multihosting\_GetEmailAccountsNumber

**Description:**

Returns email accounts number and limit.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('used_email' => 1,  
'assign_email' => -1,  
'email_limit' => 200,  
)
```

## Multihosting\_GetInfo

**Description:**

Returns multihosting info.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Multihosting\_HostingSetup\_Get

### Description:

Returns an user custom hosting setup.

Only for multihosting.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

**setupName** (string): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

### Response:

type: **struct**

Example:

(Example too long, see web.)

## Multihosting\_HostingSetup\_GetAll

### Description:

Returns all available hosting setups.

Only for multihosting.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: **struct**

Example:

```
array(array('common' => 'N',
'SetupName' => 'Unlimited',
),array('common' => 'Y',
'SetupName' => 'hosting correo10',
),array('common' => 'Y',
'SetupName' => 'hosting profesional linux',
),array('common' => 'Y',
'SetupName' => 'hosting personal linux',
))
```

## Multihosting\_HostingSetup\_GetOptions

### Description:

Returns all available hosting setup options.

Only for multihosting.

Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: **struct**

Example:

(Example too long, see web.)



## Multihosting\_HostingSetup\_Create

### Description:

Creates a new user custom hosting setup.  
 Only for multihosting.  
 Version: 1.0

### Parameters:

**serverName** ([string](#)): Server name. The parameter is required.  
**setupName** ([string](#)): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.  
**options** ([struct](#)): Setup options ( as returned by `Multihosting_HostingSetup_Get` ). The parameter is required.

### Response:

type: [void](#)

## Multihosting\_HostingSetup\_Delete

### Description:

Deletes a existing user custom hosting setup.  
 Only for multihosting.  
 Version: 1.0

### Parameters:

**serverName** ([string](#)): Server name. The parameter is required.  
**setupName** ([string](#)): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

### Response:

type: [void](#)

## Multihosting\_HostingSetup\_Update

### Description:

Updates an existing user custom hosting setup.  
 Only for multihosting.  
 Version: 1.0

### Parameters:

**serverName** ([string](#)): Server name. The parameter is required.  
**setupName** ([string](#)): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.  
**options** ([struct](#)): New setup options ( as returned by `Multihosting_HostingSetup_Get` ). The parameter is required.

### Response:

type: [void](#)

## Multihosting\_MailReport\_Set

### Description:

Set consumption email report.

Only for multihosting.  
Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**emailTo** (**string**): Email destination. The parameter is required. Parameter value must be a valid email address.

**periodicity** (**string**): Report periodicity. The parameter is required. Accepted parameter values: "daily", "monthly", "yearly".

**Response:**

type: **void**

## Multihosting\_MailReport\_Get

**Description:**

Returns consumption email report status.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('periodicity' => 'never',
'emailTo' => ",
)
```

## Multihosting\_MailReport\_SetOff

**Description:**

Deactivate consumption email report.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **void**

## Multihosting\_GetUsedTransfer

**Description:**

Returns used transfer ammount.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **struct**

Example:

```
array('total_traffic' => 4712,
'traffic_limit' => 90000,
)
```

## Multihosting\_GetUsedSpace

**Description:**

Returns used space.

Only for multihosting.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **struct**

Example:

```
array('total_space' => 0.5,
'space_limit' => 4000,
'homes_space' => 0.0126953125,
'web_space' => 0.1474609375,
'webmail_space' => 0,
'bds_space' => 0.345703125,
'mail_space' => 0,
'ftp_space' => 0,
'others_space' => 0,
)
```

## Multihosting\_Hosting\_GetAll

**Description:**

Returns existing hostings on multihosting.

Only for multihosting.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**onlyActiveHostings** (**bool**): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean ( true or false ).

**Response:**

 type: **array**

Example:

```
array('buscador.iumx.eu')
```

## Multihosting\_Hosting\_GetSpace

**Description:**

Returns used hosting space.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

type: **struct**

Example:

'None'

## Multihosting\_Hosting\_Activate

**Description:**

Activates an existing but inactive hosting.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

type: **void**

## Multihosting\_Hosting\_Create

**Description:**

Creates a new hosting.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**setupName** (**string**): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**setZones** (**bool**): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean ( true or false ).

**Response:**

type: **void**

## Multihosting\_Hosting\_Delete

**Description:**

Deletes an existing hosting.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**domainName** ([string](#)): Hosting domain name. The parameter is required.

**Response:**

type: [void](#)

## Multihosting\_Hosting\_Deactivate

**Description:**

Deactivates an existing and active hosting.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**domainName** ([string](#)): Hosting domain name. The parameter is required.

**Response:**

type: [void](#)

## Multihosting\_Hosting\_ChangeSetup

**Description:**

Changes setup of a existing hosting.

Only for multihosting.

Version: 1.0

**Parameters:**

**serverName** ([string](#)): Server name. The parameter is required.

**domainName** ([string](#)): Hosting domain name. The parameter is required.

**setupName** ([string](#)): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: [void](#)

## Payment commands

### Payment\_Method\_GetAll

**Description:**

Returns all payment methods.

1.0 -> 1.1: Only returns payment methods for autorenewals. Other payment methods are discarded.

Version: 1.1

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

### Payment\_Credit\_GetTotal

**Description:**

Returns total credit.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **float**

Example:

0.1

### Payment\_Credit\_List

**Description:**

Returns credit modification list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

### Payment\_Credit\_GetDetailed

**Description:**

Returns detailed credit information.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **struct**

Example:

```
array('credit' => 0,  
'points' => 0.1,  
'refundable' => 0,  
'notRefundable' => 0,  
'others' => 0,  
'afiliated' => 0,  
'entered' => 0,  
'bonos' => 0,  
'devolucion_credit' => 0,  
)
```

## Services commands

### Services\_GetDomains

**Description:**

Returns domain list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

### Services\_GetManagedDedicatedServers

**Description:**

Returns managed dedicated servers list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'd546.dinserver.com',  
'startDate' => '2013-06-05',  
'endDate' => '2014-06-05',  
))
```

### Services\_GetCloudServers

**Description:**

Returns cloud server list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('cloudAlias' => 'cl364',  
'startDate' => '2011-06-10',  
'endDate' => '2011-06-10',  
))
```

### Services\_GetSms

**Description:**



Returns Sms list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('account' => 650074563,
'startDate' => '2009-07-17',
'endDate' => '2013-09-17',
))
```

## Services\_GetDedicatedServers

**Description:**

Returns dedicated servers list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'dw27.dinasever.com',
'startDate' => '2011-02-16',
'endDate' => '2012-02-16',
),array('serverName' => 'd331.dinasever.com',
'startDate' => '2011-07-12',
'endDate' => '2011-10-12',
))
```

## Services\_GetManagedCloudServers

**Description:**

Returns managed cloud server list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('cloudAlias' => 'c568',
'startDate' => '2014-02-06',
'endDate' => '2014-02-06',
))
```

## Services\_GetUnmanagedDedicatedServers

**Description:**

Returns unmanaged dedicated servers list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'dl446.dinserver.com',
'startDate' => '2013-04-05',
'endDate' => '2013-07-05',
))
```

## Services\_GetDomainTransfers

**Description:**

Returns the user's domain transfers list. Status information is not immediate; it may take up to 1 hour to change.

On transfer codes: 'WAIT\_INIT','WAIT\_APROB','WAIT\_AUTH','WAIT\_TRANSFER'

Error codes: 'KO\_AUTH','FAILED','BLOCKED','REJECTED','CANCELED','DELETED','UNKNOWN'

OK codes: 'ACTIVE'

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

'None'

## Services\_GetHousingServers

**Description:**

Returns housing servers list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'colo01.dinserver.com',
'startDate' => '2011-07-01',
'endDate' => '2011-08-01',
),array('serverName' => 'colo02.dinserver.com',
'startDate' => '2011-07-01',
'endDate' => '2011-08-01',
))
```

## Services\_GetHostings

**Description:**

Returns hosting list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Services\_GetUnmanagedCloudServers

**Description:**

Returns unmanaged cloud server list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

'None'

## Services\_GetUnmanagedVps

**Description:**

Returns unmanaged VPS list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Services\_GetAllManagedServersNames

**Description:**

Returns managed server names list ( cloud, dedicated, vps, multihosting ).

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

'None'

## Services\_GetManagedVps

**Description:**

Returns managed VPS list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'vl193.dinserver.com',
'startDate' => '2012-12-14',
'endDate' => '2013-08-09',
),array('serverName' => 'vw151.dinserver.com',
'startDate' => '2013-03-19',
'endDate' => '2014-03-19',
))
```

## Services\_GetMultihostings

**Description:**

Returns Multihostings list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Services\_GetVps

**Description:**

Returns VPS list.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

```
array(array('serverName' => 'vl328.dinserver.com',
'startDate' => '2011-04-19',
'endDate' => '2011-05-19',
))
```

## Sms commands

### Sms\_SetUp

**Description:**

Sets a sms account up.

Version: 1.0

**Parameters:**

**account** (string): Account number. The parameter is required.

**alias** (string): Account alias. The parameter is optional. Parameter length must be between 0 and 100 characters.

**sender** (string): Default sender. The parameter is optional. Parameter length must be between 0 and 11 characters.

**Response:**

type: string

Example:

'None'

### Sms\_Phonebook\_GetAllContacts

**Description:**

Returns phonebook contact list.

1.0 -> 1.1: Added missing id key on response.

Version: 1.1

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**Response:**

type: struct

Example:

```
array('229' => array('name' => 'Soporte',
'number' => 900854000,
'id' => 229,
),
)
```

### Sms\_Phonebook\_CreateContact

**Description:**

Create a new phonebook contact.

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**name** (string): Contact name. The parameter is required. Parameter length must be between 1 and 50 characters.

**phoneNumber** (string): Contact phone number. The parameter is required. Parameter value must be a valid phone number.

**Response:**

 type: `void`

## Sms\_Phonebook\_UpdateContact

**Description:**

Update an existing phonebook contact.

Version: 1.0

**Parameters:**
**account** (`string`): Sms account number. The parameter is required.

**id** (`int`): Contact id. The parameter is required. Parameter value must be a valid integer.

**name** (`string`): Contact name. The parameter is required. Parameter length must be between 1 and 50 characters.

**phoneNumber** (`string`): Contact phone number. The parameter is required. Parameter value must be a valid phone number.

**Response:**

 type: `void`

## Sms\_Phonebook\_DeleteContact

**Description:**

Delete an existing phonebook contact.

Version: 1.0

**Parameters:**
**account** (`string`): Sms account number. The parameter is required.

**id** (`int`): Contact id. The parameter is required. Parameter value must be a valid integer.

**Response:**

 type: `void`

## Sms\_GetSender

**Description:**

Returns sender name ( used for "from" field ).

Version: 1.0

**Parameters:**
**account** (`string`): Sms account number. The parameter is required.

**Response:**

 type: `string`

Example:

*'Pruebas'*

## Sms\_Programmed\_Cancel

**Description:**

Cancel programmed SMS.

Please note that this only works with SMS that has not yet been sent.

Version: 1.0

**Parameters:**

**account** (*string*): Sms account number. The parameter is required.

**id** (*int*): History ID. The parameter is required.

**Response:**

type: *void*

## Sms\_PanellIndependentAccess\_SetPassword

**Description:**

Set a new password for SMS account panel independent access.

Version: 1.1

**Parameters:**

**account** (*string*): Sms account number. The parameter is required.

**newPassword** (*string*): New password. The parameter is required.

**Response:**

type: *void*

## Sms\_History\_GetProgrammed

**Description:**

Returns a delivery list.

Version: 1.0

**Parameters:**

**account** (*string*): Sms account number. The parameter is required.

**start** (*int*): Start. The parameter is required. Parameter value must be a valid integer.

**end** (*int*): End. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: *array*

Example:

'None'

## Sms\_History\_GetBulkSent

**Description:**

Returns a delivery list.

Version: 1.0

**Parameters:**

**account** (*string*): Sms account number. The parameter is required.

**start** (*int*): Start. The parameter is required. Parameter value must be a valid integer.

**end** (*int*): End. The parameter is required. Parameter value must be a valid integer.

**Response:**

type: *array*

Example:  
'None'

## Sms\_History\_GetSent

**Description:**  
Returns a delivery list.  
Version: 1.0

**Parameters:**  
**account** (string): Sms account number. The parameter is required.  
**start** (int): Start. First SMS to obtain, starting at 0. The parameter is required. Parameter value must be a valid integer.  
**end** (int): End. Last SMS to obtain. The parameter is required. Parameter value must be a valid integer.

**Response:**  
type: array  
Example:  
'None'

## Sms\_GetCredit

**Description:**  
Returns available SMS credit.  
Version: 1.0

**Parameters:**  
**account** (string): Sms account number. The parameter is required.

**Response:**  
type: string  
Example:  
0.000000

## Sms\_Send\_Bulk\_Limited\_Unicode

**Description:**  
Sends one SMS to one or many destination numbers.  
This command accepts any UTF-8 encoded character.  
Limited to one SMS for destination number.  
Version: 1.0

**Parameters:**  
**account** (string): Sms account number. The parameter is required.  
**contents** (string): Text message. It may contain any UTF-8 character. The parameter is required. Parameter length must be between 1 and 70 characters.  
**to** (array): Destination numbers. The parameter is required.  
**from** (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.  
**when** (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .



**Response:**

type: **void**

## Sms\_Send\_Bulk\_Limited\_Gsm7

**Description:**

Sends one SMS to one or many destination numbers.

This command accepts GSM7 valid UTF-8 encoded characters only.

Limited to one SMS for destination number.

Valid charlist: @ £ \$ ¥ è é ù ì ò Ç Ø ø Å å ? \_ ? ? ? ? ? ? ? ? ? ? ^ { } \ [ ~ ] | Æ æ ß É ! " # ¨ % ' ( ) + , - . / : ;  
 ? ; A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ¿ a b c d e f g h i j k l m n o p q r s t  
 u v w x y z ä ö ñ ü à 0 1 2 3 4 5 6 7 8 9

Valid special chars: whitespace, carriage return and line feed.

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**contents** (string): Text message. Limited to a 7bit charlist. The parameter is required. Parameter length must be between 1 and 160 characters. Parameter value must contain GSM7 valid chars only.

**to** (array): Destination numbers. The parameter is required.

**from** (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.

**when** (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: **void**

## Sms\_Send\_Bulk\_Long\_Unicode

**Description:**

Sends SMS to one or many destination numbers.

This command accepts any UTF-8 encoded character.

Please note the SMS may be divided on many smaller SMS (more credit is consumed).

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**contents** (string): Text message. It may contain any UTF-8 character. The parameter is required. Parameter length must be between 1 and 335 characters.

**to** (array): Destination numbers. The parameter is required.

**from** (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.

**when** (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: **void**

## Sms\_Send\_Bulk\_Long\_Gsm7

**Description:**

Sends SMS to one or many destination numbers.

This command accepts GSM7 valid UTF-8 encoded characters only.

Please note the SMS may be divided on many smaller SMS (more credit is consumed).

Valid charlist: @ £ \$ ¥ è é ù ò Ç Ø ø Å å ? \_ ? ? ? ? ? ? ? ? ^ { } \ [ ~ ] | Æ æ ß É ! " # ¨ % ' ( ) + , - . / : ;  
 ? j A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ñ Ü § ç a b c d e f g h i j k l m n o p q r s t  
 u v w x y z ä ö ñ ü à

Valid special chars: whitespace, carriage return and line feed.

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**contents** (string): Text message. Limited to a 7bit charlist. The parameter is required. Parameter length must be between 1 and 765 characters. Parameter value must contain GSM7 valid chars only.

**to** (array): Destination numbers. The parameter is required.

**from** (string): Sender name. The parameter is required. Parameter length must be between 1 and 11 characters.

**when** (string): Date and time, for programmed sending. Set empty for send now. Year-month-day Hour:minute:second. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**Response:**

type: void

## Sms\_DeliveryList\_Get

**Description:**

Returns a delivery list.

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**name** (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

**Response:**

type: struct

Example:

```
array('members' => array(),
)
```

## Sms\_DeliveryList\_GetAll

**Description:**

Return all delivery lists.

Version: 1.0

**Parameters:**

**account** (string): Sms account number. The parameter is required.

**Response:**

type: array

Example:

```
array('name' => 'sfgjshgef',
'members' => array(),
)
```

## Sms\_DeliveryList\_GetPhones

### Description:

Returns a delivery list.

Version: 1.0

### Parameters:

**account** (string): Sms account number. The parameter is required.

**name** (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

### Response:

type: array

Example:

`array(743214323)`

## Sms\_DeliveryList\_Create

### Description:

Create a new delivery list.

Version: 1.0

### Parameters:

**account** (string): Sms account number. The parameter is required.

**name** (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

**ids** (array): Phonebook ids ( array ). The parameter is required.

### Response:

type: void

## Sms\_DeliveryList\_Delete

### Description:

Delete an existing delivery list.

Version: 1.0

### Parameters:

**account** (string): Sms account number. The parameter is required.

**name** (string): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

### Response:

type: void

## Sms\_DeliveryList\_Update

### Description:

Update an existing delivery list ( overwrites previous ids ).

Version: 1.0

### Parameters:

**account** (**string**): Sms account number. The parameter is required.

**name** (**string**): Delivery list name. The parameter is required. Parameter length must be between 1 and 20 characters.

**ids** (**array**): New phonebook ids ( array ). The parameter is required.

**Response:**

type: **void**

## System commands

### System\_GetRequestTypes

**Description:**

Returns valid Request Types.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **struct**

### System\_TestSimulation

**Description:**

Test if request is on simulation mode.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **bool**

### System\_MethodHelp

**Description:**

Returns command documentation.

Version: 1.0

**Parameters:**

**commandName** (**string**): Command name. The parameter is required.

**Response:**

type: **string**

### System\_GetApiVersion

**Description:**

Returns Api version information.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## System\_MethodSignature

**Description:**

Returns the method signature.

**Warning: This command is intended for automatic XmlRpc client use with XmlRpc requests only .**

Version: 1.0

**Parameters:**

**commandName** (string): Command name. The parameter is required.

**Response:**

type: array

## System\_ListMethods

**Description:**

Lists command names.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: array

Example:

(Example too long, see web.)

## System\_GetResponseTypes

**Description:**

Returns valid Response Types.

Default response type can be changed with parameter "responseType=xxx".

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: struct

## User commands

### User\_GetServices

**Description:**

Returns a list of services.

Version: 1.1

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

### User\_Secondary\_GetAll

**Description:**

Returns all secondary users.

Version: 1.0

**Parameters:**

This command doesn't accept any parameters

**Response:**

type: **array**

Example:

(Example too long, see web.)

### User\_Secondary\_Create

**Description:**

Create a secondary user.

1.0 -> 1.1 loginData is now flagged as a needed parameter.

Version: 1.1

**Parameters:**

**personalData** (**struct**): Secondary user personal identification data. The parameter is required. Parameter value must be a struct containing user information: firstname, lastname, NIF

**companyData** (**struct**): Secondary user company identification data. The parameter is optional. Parameter value must be a struct containing user information: company, legal\_form, CIF, company\_phone, company\_fax

**contactData** (**struct**): Secondary user contact information. The parameter is required. Parameter value must be a struct containing user information: country\_code, state, city, postal\_code, address, phone, fax, email\_address

**loginData** (**struct**): Secondary user login data. The parameter is required. Parameter value must be a struct containing user information: login, password

**Response:**

type: **string**

Example:

'None'

## User\_Secondary\_Delete

### Description:

Delete a secondary user.  
Version: 1.0

### Parameters:

**userId** (**int**): Secondary user ID. The parameter is required. Parameter value must be a valid integer.

### Response:

type: **string**

Example:  
'None'

## User\_Secondary\_Update

### Description:

Update an existing secondary user.  
1.0 -> 1.1 loginData is now flagged as a needed parameter.  
Version: 1.1

### Parameters:

**idSecondaryUser** (**int**): Existing secondary user ID. The parameter is required. Parameter value must be a valid integer.

**personalData** (**struct**): Secondary user personal identification data. The parameter is required. Parameter value must be a struct containing user information: firstname, lastname, NIF

**companyData** (**struct**): Secondary user company identification data. The parameter is optional. Parameter value must be a struct containing user information: company, legal\_form, CIF, company\_phone, company\_fax

**contactData** (**struct**): Secondary user contact information. The parameter is required. Parameter value must be a struct containing user information: country\_code, state, city, postal\_code, address, phone, fax, email\_address

**loginData** (**struct**): Secondary user login data. The parameter is required. Parameter value must be a struct containing user information: login, password

### Response:

type: **string**

Example:  
'None'

## User\_GetInvoicedAmount

### Description:

Returns invoiced amount.  
Version: 1.0

### Parameters:

**startDate** (**string**): Starting date and time. Example: 2011-01-31 00:00:00. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

**endDate** (**string**): Ending date and time. Example: 2011-12-31 23:59:59. The parameter is optional. Parameter value must be a valid full date: year-month-day hour:minutes:seconds .

### Response:

type: **float**



Example:  
9355.02

## User\_GetInfo

**Description:**  
Returns all secondary users.  
Version: 1.0

**Parameters:**  
This command doesn't accept any parameters

**Response:**  
type: **struct**  
Example:  
(Example too long, see web.)

## User\_GetNotifications

**Description:**  
Returns user notifications.  
Version: 1.0

**Parameters:**  
This command doesn't accept any parameters

**Response:**  
type: **struct**  
Example:  
'None'

## User\_GetLastLoginDate

**Description:**  
Returns user last login date time.  
Version: 1.0

**Parameters:**  
This command doesn't accept any parameters

**Response:**  
type: **string**  
Example:  
'2011-05-03 13:22:05'

## User\_Default\_GetAdminContact

**Description:**  
Returns user default admin contact data.  
Version: 1.0

**Parameters:**  
This command doesn't accept any parameters

**Response:**

type: **struct**

Example:

'None'

## User\_Default\_GetDnss

### Description:

Returns user default DNS.

Version: 1.0

### Parameters:

This command doesn't accept any parameters

### Response:

type: **array**

Example:

`array('ns.dinahosting.com','ns2.dinahosting.com','ns3.dinahosting.com','ns4.dinahosting.com')`

## User\_Default\_GetTechContact

### Description:

Returns user default tech contact data.

Version: 1.0

### Parameters:

This command doesn't accept any parameters

### Response:

type: **struct**

Example:

'None'

## User\_Data\_GetValidLegalForms

### Description:

Returns all valid legal form IDs and descriptions.

Version: 1.0

### Parameters:

This command doesn't accept any parameters

### Response:

type: **array**

Example:

(Example too long, see web.)

## Vps commands

### Vps\_Services\_GetAll

**Description:**

Returns services status.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('smtp' => 'started',  
'streaming' => 'deactivated',  
'web' => 'started',  
'mysql' => 'started',  
'ftp' => 'started',  
'pop3' => 'started',  
'dinascanner' => 'started',  
)
```

### Vps\_Services\_ChangeStatus

**Description:**

Change server service status.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**serviceName** (string): Service name. The parameter is required. Accepted parameter values: "web","mysql","pop3","smtp","ftp","dinascanner".

**serviceAction** (string): Action over service. The parameter is required. Accepted parameter values: "start","stop","activate","deactivate","restart".

**Response:**

type: **void**

### Vps\_Control\_GetApacheProcesses

**Description:**

Returns apache active processes information.

Only for linux managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (string): Server name. The parameter is required.

**Response:**

 type: **array**

Example:

(Example too long, see web.)

## Vps\_Control\_GetSystemStatus

**Description:**

Returns sysyem information.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **struct**

Example:

(Example too long, see web.)

## Vps\_Control\_GetMysqlProcesses

**Description:**

Returns mysql active processes information.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **array**

Example:

```
array(array('info' => 'show',
'id' => 4614,
'host' => 'gdc.dinaserver.com:59830',
'ddbb' => 'NULL',
'user' => 'root',
'command' => 'Query',
'time' => 'NULL',
'status' => 'NULL',
))
```

## Vps\_Control\_GetCpuLoadByHosting

**Description:**

Returns cpu load by hosting information.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **array**

Example:

```
array(array('hosting' => 'vl328.dinserver.com',
'used_cpu' => 0.04,
'percent_used_cpu' => 100,
),array('hosting' => 'others',
'used_cpu' => 0,
'percent_used_cpu' => 0,
))
```

## Vps\_Control\_GetSystemLoad

**Description:**

Returns cpu load by hosting information.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **struct**

Example:

(Example too long, see web.)

## Vps\_Control\_KillMysqlProcess

**Description:**

Kill mysql active processes.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**pid** (**int**): Process id. The parameter is required. Parameter value must be a valid integer.

**Response:**

 type: **void**

## Vps\_GetAllHostingsFullReport

**Description:**

Returns a summary of the space, transfer data and plan per hosting.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: **struct**

Example:

'None'

## Vps\_GetLicenses

### Description:

Returns licenses information.  
 Only for managed virtual servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**

Example:

```
array('totalLicenses' => 25,
'usedLicenses' => 8,
)
```

## Vps\_Monitorization\_Get

### Description:

Returns monitorization configuration.  
 Only for non-managed virtual servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

### Response:

type: **struct**

Example:

```
array('emails' => array('ejemplo@example.com'),
'activated' => true,
)
```

## Vps\_Monitorization\_Activate

### Description:

Activate monitorization.  
 Only for non-managed virtual servers.  
 Version: 1.0

### Parameters:

**serverName** (**string**): Server name. The parameter is required.

**emails** (**array**): Array of emails. The parameter is required.

### Response:

type: **void**

## Vps\_Monitorization\_Deactivate

### Description:

Deactivate monitorization.  
 Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **void**

## Vps\_GetServerInfo

**Description:**

Returns VPS server info.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

```
array('name' => 'vl328.dinserver.com',  
'main_ip' => '82.98.167.9',  
'ips' => array(),  
)
```

## Vps\_HostingSetup\_Get

**Description:**

Returns an user custom hosting setup.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**setupName** (**string**): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Vps\_HostingSetup\_GetAll

**Description:**

Returns all available hosting setups.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Vps\_HostingSetup\_GetOptions

**Description:**

Returns all available hosting setup options.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Vps\_HostingSetup\_Create

**Description:**

Creates a new user custom hosting setup.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**setupName** (**string**): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**options** (**struct**): Setup options ( as returned by Vps\_HostingSetup\_Get ). The parameter is required.

**Response:**

type: **void**

## Vps\_HostingSetup\_Delete

**Description:**

Deletes a existing user custom hosting setup.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**setupName** (**string**): Setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: **void**

## Vps\_HostingSetup\_Update



**Description:**

Updates an existing user custom hosting setup.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**setupName** (**string**): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**options** (**struct**): New setup options ( as returned by `Vps_HostingSetup_Get` ). The parameter is required.

**Response:**

type: **void**

## Vps\_GetMonitorizedPorts

**Description:**

Returns monitorized ports with their status.

Only for unmanaged virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **array**

Example:

(Example too long, see web.)

## Vps\_PowerStatus\_Unpause

**Description:**

Unpause paused server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **void**

## Vps\_PowerStatus\_TurnOff

**Description:**

Turn off server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

 type: `void`

## Vps\_PowerStatus\_Get

**Description:**

Get server status.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**
`serverName` (`string`): Server name. The parameter is required.

**Response:**

 type: `string`

Example:

`'Idle'`

## Vps\_PowerStatus\_TurnOn

**Description:**

Turn on server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**
`serverName` (`string`): Server name. The parameter is required.

**Response:**

 type: `void`

## Vps\_PowerStatus\_Reactivate

**Description:**

Reactivate suspended server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**
`serverName` (`string`): Server name. The parameter is required.

**Response:**

 type: `void`

## Vps\_PowerStatus\_Suspend

**Description:**

Suspend server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

`serverName` (string): Server name. The parameter is required.

**Response:**

type: void

## Vps\_PowerStatus\_ElectricRestart

**Description:**

Electric restart of server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

`serverName` (string): Server name. The parameter is required.

**Response:**

type: void

## Vps\_PowerStatus\_Pause

**Description:**

Pause server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

`serverName` (string): Server name. The parameter is required.

**Response:**

type: void

## Vps\_PowerStatus\_Restart

**Description:**

Soft restart server.

Only for non-managed virtual servers.

Version: 1.0

**Parameters:**

`serverName` (string): Server name. The parameter is required.

**Response:**

type: void

## Vps\_GetDataTransferInfo

**Description:**

Returns data transfer summary.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**Response:**

type: *struct*

Example:

(Example too long, see web.)

## Vps\_Nfs\_Get

**Description:**

Returns NFS server name

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name The parameter is required.

**Response:**

type: *struct*

Example:

'None'

## Vps\_MailReport\_Set

**Description:**

Set consumption email report.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**emailTo** (*string*): Destination emails separated with commas. The parameter is required. Parameter value must be a valid email address.

**periodicity** (*string*): Report periodicity. The parameter is required. Accepted parameter values: "daily", "weekly", "monthly".

**Response:**

type: *void*

## Vps\_MailReport\_Get

**Description:**

Returns consumption email report status.

Only for managed virtual servers.

Version: 1.0

**Parameters:**

**serverName** (*string*): Server name. The parameter is required.

**Response:**

type: *struct*

Example:

```
array('periodicidad' => 'nunca',
'direccionDeDestino' => "",
)
```

## Vps\_MailReport\_SetOff

### Description:

Deactivate consumption email report.  
Only for managed virtual servers.  
Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: void

## Vps\_GetUsedSpace

### Description:

Returns used space.  
Only for managed virtual servers.  
Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

### Response:

type: struct

Example:

```
array('total_space' => 71,
'space_limit' => 10240,
'homes_space' => 0.1240234375,
'web_space' => 66.9951171875,
'webmail_space' => 0,
'bds_space' => 3.171875,
'mail_space' => 0.396484375,
'ftp_space' => 0.0078125,
'others_space' => 0,
)
```

## Vps\_Hosting\_GetAll

### Description:

Returns existing hostings on Vps.  
Only for managed virtual servers.  
Version: 1.0

### Parameters:

**serverName** (string): Server name. The parameter is required.

**onlyActiveHostings** (bool): True to add filter: return only active hostings. False or empty, execute without filter. The parameter is optional. Parameter value must be a boolean ( true or false ).

**Response:**

 type: **array**

Example:

```
array('xen.delira.eu','java.delira.eu','soportedinahosting.com','dominiopersonal.com','nophp.com','probaformacion.soportedinahosting.com','pruebapersonal.com','correodeprueba.com')
```

## Vps\_Hosting\_Activate

**Description:**

Activates an existing but inactive hosting.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

 type: **void**

## Vps\_Hosting\_Create

**Description:**

Creates a new hosting.

Only for managed virtual servers.

1.0 -&gt; 1.1: Added an existing setup name validation.

Version: 1.1

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**setUpName** (**string**): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**setZones** (**bool**): True to set new zones, false to leave them unchanged. Default: true. The parameter is required. Parameter value must be a boolean ( true or false ).

**Response:**

 type: **void**

## Vps\_Hosting\_Delete

**Description:**

Deletes an existing hosting.

Only for managed virtual servers.

Version: 1.0

**Parameters:**
**serverName** (**string**): Server name. The parameter is required.

**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

 type: **void**

## Vps\_Hosting\_Deactivate

**Description:**

Deactivates an existing and active hosting.  
Only for managed virtual servers.  
Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

type: **void**

## Vps\_Hosting\_GetSetupName

**Description:**

Returns hosting setup name.  
Only for managed vps servers.  
Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**domainName** (**string**): Hosting domain name. The parameter is required.

**Response:**

type: **string**

Example:  
'None'

## Vps\_Hosting\_ChangeSetup

**Description:**

Changes setup of a existing hosting.  
Only for managed virtual servers.  
Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.  
**domainName** (**string**): Hosting domain name. The parameter is required.  
**setupName** (**string**): Existing setup name. The parameter is required. Parameter length must be between 1 and 30 characters.

**Response:**

type: **void**

## Vps\_GetBandwidthConsumption

**Description:**

Returns bandwidth consumption summary.  
Only for non-managed virtual servers.  
Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**month** (**int**): Month number. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 1 and 12 inclusive.

**year** (**int**): Year. The parameter is required. Parameter value must be a valid integer. Parameter value must be between values 2002 and 2013 inclusive.

**Response:**

type: **struct**

Example:

(Example too long, see web.)

## Vps\_GetSpaceAndTransferReportPerHosting

**Description:**

Returns a summary of the space and transfer data per hosting.

Version: 1.0

**Parameters:**

**serverName** (**string**): Server name. The parameter is required.

**Response:**

type: **struct**

Example:

'None'