

QUICK REFERENCE FOR LEGION 1.8 COMMANDS

CONTEXT SPACE

legion_cat <context path1> <context path2> ... <context pathN> [-debug] [-help]	Print the contents of a Legion file object to standard output.
legion_cd <context path> [-debug] [-help]	Change the current working context.
legion_context_add <object LOID> <context name> [-debug] [-help]	Give an object a context name.
legion_context_create <context path> [-debug] [-help]	Create a new sub-context.
legion_cp [-r][-v][-m][-p][-localsrc] [-localdest] [-V <vault context path>] <source path> <destination path> [-debug] [-help]	Copy a Legion file object
legion_export_dir [-v] [-help] [-noserver] [-autorehash] <local directory path> <target context path>	Export a local directory to context space without copying it to Legion space.
legion_export_dir_quit <target context path>	Pauses <code>legion_export_dir</code> for the specified context path.
legion_export_dir_rehash <target context path>	Checks an exported local directory for changes.
legion_import_tree [<unix directory path> [<legion context path>]] [-debug] [-help]	Copy a local Unix directory tree into a Legion context.
legion_list_names {[-c <object context path>] -l <object LOID> } [-debug] [-help]	List an object's context names.
legion_ln <context path> <new alias> [-debug] [-help]	Give an object an additional context name.
legion_ls [-laLRAdqv] [<context path>] [-debug] [-help]	List a context's contents.
legion_mkdir <context path> [-debug] [-help]	Create a new sub-context.
legion_mv <context path> <new context path>[-debug] [-help]	Change an object's context name.
legion_pwd [-debug] [-help]	Print the current context path
legion_rm [-r] [-f] [-v] <context path list> [-debug] [-help]	Remove one or more context name[s].

STARTING UP / SHUTTING DOWN THE SYSTEM

legion_create_class [-c <context path>] [-sc <scheduler context path>] [-sl <scheduler LOID>] [-debug] [-help]	Create a new instance of the class <code>VanillaMetaClass</code> .
legion_init_arch [-debug] [-help]	Create and register implementation objects for commonly used classes in the current architecture.
legion_init_Apps	Initializes the Applications package.
legion_init_Extra	Initializes the Extra package.
legion_init_HPC	Initializes the High Performance Computing package.
legion_initialize	Create basic classes and implementations in a new Legion system.

(cont'd)

legion_make_setup_script [-o <script basename>] [-OPR <OPR dir name>] [-L <\$LEGION dir name>] [-debug] [-help]	Generates a Legion setup script for your system.
legion_print_config [-debug] [-help]	Print the “well-known” LegionClass binding in the current Legion configuration.
legion_setup_state [-i] [-debug] [-help]	Create OPRs for basic Legion system objects.
legion_shutdown [-local] [-f] [-i] [-debug] [-help]	Shut down a Legion system.
legion_shutdown_class {[-c <context path> -l <class LOID>} [-debug] [-help]	Deactivate a class and all of its instances.
legion_starthost [-L <\$LEGION>] [-O <\$LEGION_OPR>] [-A <\$LEGION_ARCH>] [-B <path>] [-N <context name>] [-U <user id>] [-C <host class>] <new host name> [<compatible vault list>] [-debug] [-help]	Create a new host object. Start up a system
legion_startup [-local]	Create a new vault object
legion_startvault [-L <\$LEGION>] [-O <\$LEGION_OPR>] [-A <\$LEGION_ARCH>] [-N <context name>] [-U <user id>] [-C <vault class>] <host name> [<compatible host list>] [-debug] [-help]	

SECURITY

legion_add_acl [[-c <object context path> -l <object LOID>] { -s <filename> } [-debug] [-help]	Adds in a new access control list.
legion_add_implicit_params [[-c <AuthenticationObject context path> -l <AuthenticationObject LOID>] { -s <filename> } [-debug] [-help]	Add implicit parameters to an object or environment.
legion_change_acl [[-c <object context path> -l <object LOID>] <function pattern> <modifications>	Change an access control list (ACL) for a set of functions on a specified Legion object.
legion_change_owner [-v] [-r] [[-c <object context path> -l <object LOID>] [[-c <target owner context path> -l <target owner LOID>]] [-debug] [-help]	Change an object’s owner.
legion_change_permissions [+ -rwx] [-v] <group/user context path> <target context path> [-debug] [-help]	Change an object’s read, write, and execute permissions
legion_create_user <user id> [-debug] [-help]	Create a new Legion user id.
legion_create_user_object [[-c <class context path> -l <class LOID>] [-h <host for new object> -v <vault for new object>] [-f <implicit parameter file>] [-z <password>] <user name> [-debug] [-help]	Create a new user object.
legion_get_acl [[-c <object context path> -l <object LOID>] [-debug] [-help]	Print the access control list of an object.
legion_get_implicit_params [[-c <object context path> -l <object LOID>] [-debug] [-help]	Print the implicit parameters of an object or of the current environment.
legion_init_security	Create the initial user (called by default /user/admin) in a new Legion system.
(cont’d)	

legion_login [<user id path> -l <user id LOID>] [-debug] [-help]	Log into a running Legion system.
legion_logout [-debug] [-help]	Log out of Legion
legion_modify_parameters [-debug] [-help] <argument>	Update current session's implicit parameters.
legion_passwd <user id context> -l <user id LOID>] [-debug] [-help]	Change a user password.
legion_set_acl {[[-c] <object context path> -l <object LOID>] [-s <filename>]} [-debug] [-help]	Set the access control list of a Legion object.
legion_set_implicit_params {[[-c] <object context path> -l <object LOID>] [-s <filename>]} [-debug] [-help]	Set an AuthenticationObject's or current environment's implicit parameters.
legion_set_message_security [-c <AuthenticationObject context path> -l <AuthenticationObject LOID>] {Off Protected Private} [-help]	Updates the message layer security setting by changing an AuthenticationObject's or current environment's "MessageSecurity" implicit parameter.
legion_show_acl {[[-c] <object context path> -l <object LOID>] [<function pattern>]} [-showLoids]	Display an object's access control list (ACL) in human-readable form.
legion_whereis {[[-c] <object context path> -l <object LOID>] [-debug] [-help]}	Return a specified object's host and vault object.
legion_whoami [-debug] [-help]	Print current user name, if you are currently logged in to a security-enabled system.

OBJECTS

legion_activate_object {[[-c] <context path> -l <object LOID>] [-debug] [-help]}	Activate an object (move it to an active state).
legion_allow_activation [-entire_class] [-debug] [-help] {[-c] <context path> -l <LOID>}	Activate an object. Used in association with <code>legion_deactivate_instances</code> or <code>legion_deactivate_object's -stay_down</code> flag.
legion_cat <context path> [-debug] [-help]	Print a Legion file object's contents to standard output.
legion_classof {[[-c] <context path> -l <object LOID>]} [-debug] [-help]	Display the LOID of an object's class.
legion_copy_class [-v] [-h] {[[-c] <template context> -l] <template LOID>] <new class path> [-a <attribute>]* {-mc [-c <metaclass context> -l <metaclass LOID>]}	Copy a class object.
legion_create_object {[[-c] <class context path> -l <class LOID>] <new object context path> [-h <host name on which to place new object>] [-v <vault on which to place new object>] [-H <context path of preferred host class>] [-V <context path of preferred vault class>] [-Ch <context containing list of preferred hosts>] [-Cv <context containing list of preferred vaults>] [-d <recorder context path> <debug session name>] [-debug] [-help]}	Create a new instance of a class.
legion_create_object_r {[[-c] <class context path> -l <class LOID>] <context path> <host name> <host architecture> <\$LEGION> <\$LEGION_OPR> <\$LEGION_OPA> <binary path> <user id>] [-debug] [-help]}	Create a new object using the <code>rsh</code> (remote shell) mechanism. Note that <code>ssh</code> can be used in place of <code>rsh</code> .
legion_deactivate_object [stay_down] [-debug] [-help] {[-c] <object context name> -l <object LOID>}	Deactivate an object (move it to an inert state).
legion_destroy_object {[[-c] <context path> -l <object LOID>] [-debug] [-help]}	Delete an object's LOID.

(cont'd)

legion_get_host {[[-c] <object context path> -l <object LOID>} [-debug] [-help]	Return the LOID of an object's host
legion_get_vault {[[-c] <object context path> -l <object LOID>} [-debug] [-help]	Return the LOID of an object's vault.
legion_list_attributes {[[-c] <context path> -l <object LOID>} [-L] [<attribute name>] [-debug] [-help]	List an object's attributes.
legion_list_invocations {[[-c] <object context path> -l <object LOID>} [-debug] [-help]	Return current pending and started invocations for an object
legion_object_info {[[-c] <object context path> -l <object LOID>} [-v] [-debug] [-help]	Get information (host, vault, OA, owner, etc.) about an object
legion_ping {[[-c] <context path> -l <object LOID>} [-timeout <seconds>] [-debug] [-help]	Ping an object to see if it is active.
legion_set_backup_vaults {[-c] <instance context path> -l <instance LOID>} [-nodeactivate]] {[-a -d] [-c <vault1 context path> -l <vault1 LOID>} [-a -d] [-c <vault2 context path> -l <vault2 LOID>} ... [-a -d] [-c <vaultn context path> -l <vaultn LOID>]] {-Cv <vault context path> -n <total number of vaults>}	Set one or more backup vaults for an object.
legion_set_binding_agent [-unset] [-make_default] [-make_default_only] {-l <object context path> -c <object LOID>}	Set or unset a binding agent as the user's current and/or future login session binding agent.
legion_set_host {[[-c] <instance context path> -l <instance LOID>} {[-c] <host context path> -l <host LOID>} [-debug] [-help]	Move an object to a different host object.
legion_refresh_local_cache	Manually refresh some high-use objects' bindings and high-use contexts' LOIDs in your current login session.
legion_set_vault {[[-c] <instance context path> -l <instance LOID>} {[-c] <vault context path> -l <vault LOID>} [-permanent] [-debug] [-help]	Move an object's OPR to a different vault object.
legion_set_worm {[[-c] <object context path> -l <object LOID>}	Tell Legion that an object uses WORM syntax.
legion_skcc_set_class_vaults <SKCC class context path> {[-a -d] [-c <vault1 context path> -l <vault1 LOID>} [-a -d] [-c <vault2 context path> -l <vault2 LOID>} ... [-a -d] [-c <vaultn context path> -l <vaultn LOID>]]	Set an allowable set of backup vaults on which to store the specified SKCC class's replicated state.
legion_skcc_set_defaults {[-l <SKCC class LOID> -c <SKCC class context path>} <number of default vaults>	Specifies default level of state replication for instances of an SKCC class.
legion_synch_vaults {[[-c] <instance context path> -l <instance LOID>} [-nodeactivate]]	Synchronize an object's primary and backup vaults.
legion_2drm <file name>	Removes a 2D file object from Legion.
legion_unset_worm {[[-c] <instance context path> -l <instance LOID>}	Tell Legion that an object does not use WORM syntax.
legion_update_attributes {[[-c] <object context path> -l <object LOID>} [-a <new attribute>] [-d <attribute>] [-r <old attribute> <new attribute>] [-debug] [-help]	Change an object's attributes.

CLASSES

legion_activate_instances {[-c <class context path> -l <class LOID>} [-debug] [-help]	Activate a class's instances.
legion_allow_activation [-entire_class] [-debug] [-help] {[-c] <context path> -l <LOID>}	Activate an object. Used in association with the -stay_down option (legion_deactivate_instances or legion_deactivate_object).
legion_classof {-l <object LOID [-c] <context path>} [-debug] [-help]	Display the LOID of an object's class.
legion_deactivate_instances [-stay_down] [-debug] [-help] {[-c] <class context name> -l <class LOID>}	Deactivate a class's instances.
legion_destroy_instances {[-c] <class context path> -l <class LOID>} [-debug] [-help]	Destroy all of a class's instances' LOIDs.
legion_exports_interface {[-c] <context path> -l <LOID>} {-w <well-known class type> -f <function signature>}+ [-debug] [-help]	Create an interface for an object.
legion_FTPd [-help] [-v] [-p <port number>] [-a] [-m <max connections>] [-t <transfer block size>] [-o <max outstanding invocations>] [-d] [-w] [-e] [-rt <RMI timeout>] [-dt <detail timeout>]	Starts the Legion FTP daemon from the command line.
legion_get_interface {[-c] <context path> -l <class LOID>} [-debug] [-help]	Retrieve an interface for objects of a given class.
legion_list_implementations [-v] {[-c] <class context path> -l <class LOID>} [-debug] [-help]	List a class's implementation objects
legion_list_instances {[-c] <class context path> -l <class LOID>} [-debug] [-help]	List the instances of a class.
legion_shutdown_class {[-c] <class context path> -l <class LOID>} [-debug] [-help]	Deactivate a class and all of its instances.

LEGIONCLASS

legion_add_class_mapping <metaclass LOID> <class LOID> [-debug] [-help]	Tell Legion that a given class is another's class's metaclass.
legion_combine_domains [-v] <list of domain cookie files> [-debug] [-help]	Combine one or more Legion domains
legion_create_implementation <binary path name> <architecture> {[-c] <class context path> -l <class LOID>} {[-c] <object context path>} [-nc] [-help] [-v] [-a <attribute>] [-debug] [-help]	Create a new implementation object.
legion_generate_domain_cookie [-o <cookie output filename>] [-debug] [-help]	Generate a domain cookie.
legion_list_domains [-debug] [-help]	List domains connected to the current domains
legion_print_domain_cookie [-i <cookie input file>] [-debug] [-help]	Display the current domain's cookie file.

RUNNING APPLICATIONS

legion_bfs <file> [-o <out file>] [-Backend] [-help]	Parses Fortran programs containing Legion (BFS) pseudo-comments.
legion_ft_initialize	Initializes the CheckpointStorage object.
legion_link [-CC <compiler>] [-Fortran] [-FC <compiler>] [-pvm] [-mpi] [-L<library path>] [-<library>] [-v] [-o <output file>] [-bfs] <object file list> [-debug] [-help]	Links together a set of object files and libraries to produce an executable program and automatically binds the produced executables to the Legion libraries.
legion_nq [-help] [-w] [-v] [-a <arch>] [-h <host context path>] [-in <context name>] [-out <context name>] [-IN <local file name>] [-OUT <local file name>] [-f <options file>] [-novrun] [-t <minutes>] [-n <nodes>] [-r <minutes>] [-p <priority>] [-d] <program class> [<arg1> <arg2> ... <argn>]	Submits jobs to a Legion JobQueue object.
legion_make_idl [-notrans] [-noclient] [-v] [-noreg] [-s <suffix string>] [-run] <application name> [-debug] [-help]	Compiles Legion stubs and implementation code of CORBA client and server classes.
legion_make [-v] [-a <architecture>] [-h <host context path>] [-e <make command>] [-OUT <remote file>] [<arg1> <arg2> ... <argn>] [-debug] [-help]	Compile programs on a remote host.
legion_make_multi [-v] [-a <architecture>] [-e <make command>] [<arg1> <arg2> ... <argn>] [-debug] [-help]	Compile a program on multiple platforms.
legion_manage_job [-help] [-k] [-cp <priority>] [-fr] [-nc] [-q <context name>] <ticket number>	Manage a job started with <code>legion_nq</code> .
legion_manage_queue [-help] [-purge] [-maxrs <slots>] [-q <context name>]	Manage a Legion JobQueue object.
legion_mpi_debug [-q] [-c] <instance context path> [-help]	Examine the state of MPI objects and print their message queues.
legion_mpi_probe [-help] [-debug] [-v[erbose]] [-all] [-list] [-in <context path name>] [-out <context path name>] [-IN <local file name>] [-OUT <local file name>] [-showscratch] [-stat <remote file name>] <pid context name>	Checks an MPI program started on a remote host with <code>legion_mpi_run</code> .
legion_mpi_register <class name> <binary local path name> <platform type> [-help]	Register an MPI program for use with Legion MPI.
legion_mpi_run {-f <options file> [<flags>]} {-n <number of hosts> [<flags>] <command> [<arguments>]} [-help]	Run an MPI program in Legion (see the Reference Manual for information about <flags>).
legion_mplc [-help]	Compile a Mentat Programming Language program.
legion_mplc_reg_impl <class name> <binary path> <stateless stateful sequential> <arch> [-debug] [-help]	Register the given binary as a stateful, stateless, or sequential Mentat object for a given architecture.
legion_native_mpi_config_host [-wrapper>] [-debug] [-help]	Set native MPI properties on a host.
legion_native_mpi_init [<architecture>] [-debug] [-help]	Install the <code>legion_native_mpi_backend</code> class in a native MPI host.
legion_native_mpi_register <class name> <binary path> <architecture> [-help]	Register a native MPI program with your Legion system.

(cont'd)

<p>legion_native_mpi_run [-v] [-a <architecture>] [-h <host context path>] [-IN <local input file>] [-OUT <local result file>] [-in <Legion input file>] [-out <Legion result file>] [-n <nnodes>] [-t <minutes>] [-legion] [-help] [-debug] <program context path> [<arg 1> <arg 2> ... <arg n>]</p>	<p>Start a native MPI program.</p>
<p>legion_pvm_register <class path name> <binary local path name> <platform type> [-help]</p>	<p>Register a PVM task for use with Legion.</p>
<p>legion_register_program <program class> <executable path> <legion arch> [-debug] [-help]</p>	<p>Register a Legion-independent program for use with Legion.</p>
<p>legion_register_runnable <program class> <executable path> <legion arch> [-debug] [-help]</p>	<p>Register a Legion-linked program for use with Legion.</p>
<p>legion_run [-help] [-debug] [-v[erbose]] [-w] [-a <arch>] [-h <host>] [-block] [-non[-]block] [-d[ir] <dir>] [-D <var=value>] [-in <contextfile>] [-out <contextfile>] [-IN <localfile>] [-OUT <localfile>] [-stdin <localfile>] [-stdout <localfile>] [-stderr <localfile>] [-novrun] [-p[robe] <localfile>] [-setscratch <context>] [-meta <option=value>] [-f <options file>] <program class> [<arg1> ... <argn>]</p>	<p>Remotely execute a registered program in Legion.</p>
<p>legion_run_multi [-help] [-debug] [-v[erbose]] [-z[ero]] [-r[estart]] [-e[xec] command] [-x <exceptfile>] [-a[rch] <architecture>] [-d[ir] <dir>] [-D <var=value>] {-n <number of processors> -s[chedule] <schedfile>} -f[file] <specfile> [-t[ime] <timefile>] [--] <program class> [<arg1> <arg2> ... <argn>]</p>	<p>Run a serial program with specified input file</p>
<p>legion_stateless_add_workers <class name> <worker name1> <worker name2> ... <worker nameN> [-debug] [-help]</p>	<p>Create additional MPL stateless object workers. Adding more workers allows message calls to be handled more efficiently.</p>
<p>legion_stateless_configure <stateless class context path> [-n <number of replicas>] [-Ch <host context path>] [-w <max number work requests>] [-FIUsH] [-debug] [-help]</p>	<p>Configure stateless Legion classes</p>
<p>legion_stateless_remove_workers <class name> <worker name1> <worker name2> ... <worker nameN> [-debug] [-help]</p>	<p>Remove MPL stateless object workers.</p>

TTY OBJECTS

<p>legion_direct_output [[-c] <object context path> -l <object LOID>] {[-c] <tty context path> -l <tty LOID>} [-debug] [-help]</p>	<p>Direct an object's output to a Legion tty object.</p>
<p>legion_set_tty <tty context path> [-debug] [-help]</p>	<p>Set a tty object.</p>
<p>legion_tty <tty context path> [-debug] [-help]</p>	<p>Create, set, and watch a tty object.</p>
<p>legion_tty_off [-debug] [-help]</p>	<p>Stop watching the current tty object (works only with tty object created with the <code>legion_tty</code> command).</p>
<p>legion_tty_redirect <object context path> [-debug] [-help]</p>	<p>Redirect the current Legion tty object's output to a specified file object.</p>
<p>legion_tty_unredirect <object context path> [-debug] [-help]</p>	<p>Stop directing the current tty object's output to a specified file object.</p>
<p>legion_tty_watch [[-c] <tty context path> -l <tty LOID>] [-debug] [-help]</p>	<p>Direct the current tty object's output to print to standard output.</p>

THE STATE OF THE SYSTEM

legion_check_system [-v] [-q] [-fix CommandLine -fix Contexts] [-debug] [-help]	Check the status of key elements in your Legion system.
legion_classof {[[-c] <context path> -l <object LOID>} [-debug] [-help]	Display the LOID of an object's class.
legion_create_stat_tree <base context path> [-debug] [-help]	Create a database of current loads on all host objects in a system.
legion_get_accounting_data [-v] [-help] [-debug] {<pull-source1> <pull-source2> ... <pull-sourcen>} {-d <local target directory>} [-ot <seconds>} [-it <seconds>]	Retrieves accounting information logs from specified pull-sources and places them in a specified target directory in local file space.
legion_host_stats {[[-c] <host context path> -l <host LOID>} [-debug] [-help]	Print the number of objects and current load for a particular host object.
legion_list_objects {[[-c] <host context path> -l <host LOID>} [-debug] [-help]	List the Legion objects currently managed by a given host object.
legion_wellknown_class <well-known class name> [-debug] [-help]	Print the LOID of a "well-known" Legion class.
legion_update_accounting_db [-v] <local configuration file path> <local logfile directory>	Parse the data in all accounting information logs in a specified local directory and place the data into a MySQL database.
legion_update_stat_tree [-v] [-help] [-t <repeat delay seconds>] [-site <top site context path>] [-collection <collection context path>]	Update database of current loads on all host objects in a system.

SCHEDULING

legion_config_scheduler {[[-c] <scheduler path> -l <scheduler LOID>} [-get_enactor] [-get_collection] [-set_enactor {[[-c] <Enactor path> -l <Enactor LOID>}] [-set_collection {[[-c] <Collection path> -l <Collection LOID>}] [-debug] [-help]	Configure a Legion scheduler's Collection and Enactor
legion_join_collection {[[-c] <Collection path> -l <Collection LOID>} {[-c] <member path> -l <member LOID>} [-debug] [-help]	Joins objects to a Collection.
legion_leave_collection {-l <Collection LOID> [-c] <Collection path>} {[-c] <member path> -l <member LOID>} [-debug] [-help]	Remove objects from a Collection.
legion_make_schedule {-c <class context path> -l <class LOID>} [-n <number of nodes>] [-f <specification file>] [-q <query>] [-help]	Create a host file for running an instance of a class.
legion_query_collection [-v] {[-c] <Collection context path> -l <Collection LOID>} <query> [-debug] [-help]	Get information on a Collection.
legion_set_scheduler {[[-c] <class context path> -l <class LOID>} {[-c] <Scheduler context path> -l <Scheduler LOID>} [-debug] [-help]	Set a the default Scheduler for a class.
legion_set_scheduler_policy {[[-c] <class context path> -l <class LOID>} <policy> [-debug] [-help]	Set a class's policy for using its default Scheduler.

RESOURCE MANAGEMENT

<p>legion_add_host_account {-l <host object LOID> [-c] <host object context path>} {[-f <mapping file name>] [<Unix user id> [-l <owner LOID> -c <owner context path>]]} [-debug] [-help]</p>	<p>Add one or more mappings between Unix user ids and Legion user ids for a PCD host object.</p>
<p>legion_class_host_list {[-c] <class context path> -l <class LOID>} {[-a -d -t] <host1> <host2> ... <hostn>} [-p] [-debug] [-help]</p>	<p>Change a class's list of acceptable hosts (these will be listed by LOID).</p>
<p>legion_class_vault_list {[-c] <class context path> -l <class LOID>} {[-a -d -t] <vault1> <vault2> ... <vaultn>} [-p] [-debug] [-help]</p>	<p>Change a class's list of acceptable vaults (these will be listed by LOID).</p>
<p>legion_destroy_host [-v] {[-c] <host context path> -l <host LOID>} [-debug] [-help]</p>	<p>Destroy a host.</p>
<p>legion_destroy_vault {[-c] <vault context path> -l <vault LOID>} [-debug] [-help]</p>	<p>Destroy a vault.</p>
<p>legion_host_vault_list {[-c] <host context path> -l <host LOID>} {[-a -d -t] <vault1> <vault2> ... <vaultn>} [-p] [-debug] [-help]</p>	<p>Change a host object's list of compatible vault objects.</p>
<p>legion_instance_host_list {[-c] <instance context path> -l <instance LOID>} {[-a -d -t] <host1> <host2> ... <hostn>} [-p] [-debug] [-help]</p>	<p>Change an object's list of acceptable hosts.</p>
<p>legion_instance_vault_list {[-c] <instance context path> -l <instance LOID>} {[-a -d -t] <vault1> <vault2> ... <vaultn>} [-p] [-debug] [-help]</p>	<p>Change an object's list of acceptable vaults.</p>
<p>legion_list_host_accounts {-l <host object LOID> [-c] <host object context path>} <user id> [-debug] [-help]</p>	<p>List available accounts on a PCD host object.</p>
<p>legion_list_oprs {[-c] <vault context path> -l <vault LOID>} [-debug] [-help]</p>	<p>List all OPRs currently stored in a given vault.</p>
<p>legion_remove_host_account {-l <host object LOID> [-c] <host object context path>} [-debug] [-help]</p>	<p>Remove one or more account mappings from a PCD host object.</p>
<p>legion_set_default_placement {[-c] <class context path> -l <class LOID>} {[-c] <host context path> -l <host LOID>} {[-c] <vault context path> -l <vault LOID>} [-debug] [-help]</p>	<p>Set default placement for a class's instances if it can't contact its associated Scheduler.</p>
<p>legion_set_varch {[-c] <host context path> -l <host LOID>} <arch> [-debug] [-help]</p>	<p>Set a a virtual host's architecture.</p>
<p>legion_set_vrun {[-c] <host context path> -l <host LOID>} <path> [-debug] [-help]</p>	<p>Configure a virtual host object.</p>
<p>legion_starthost {-L <\$LEGION>} [-O <\$LEGION_OPR>] [-A <\$LEGION_ARCH>] [-B <path>] [-N <context name>] [-U <user id>] [-C <host class>] <new host name> [<compatible vault list>] [-debug] [-help]</p>	<p>Create a new host object.</p>

(cont'd)

legion_startvault [-L <\$LEGION>] [-O <\$LEGION_OPR>] [-A <\$LEGION_ARCH>] [-N <context name>] [-U <user id>] <host name> [<compatible host list>] [-debug] [-help]	Create a new vault object.
legion_vault_host_list {[-c] <vault context path> -l <vault LOID>} [{-a -d -t} <host1> <host2> ... <hostn>] [-p] [-debug] [-help]	Change a vault's list of compatible host objects.

DEBUGGING

legion_output_state {[-c] <object context path> -l <object LOID>} StartupState [-debug] [-help]	An output device for debugging and information reporting.
legion_record {-uf <local storage file name> [-c] <recorder context path>} [-name <debug session name>] [-t <interval>] <client application> [<arg1> <arg2> ... <argN>] [-debug] [-help]	Execute an application in record mode.
legion_replay {-uf <local storage file name> [-c] <recorder context path>} -name <debug session name> {-list [-cmd <debugger name>] [-local] <session number>} [-debug] [-help]	Start a replay or playback session for a recorded object.