



CARMA

Version: 3.0.1

Date: 2024-12-13

Stealth Software Technologies, Inc.

Table of Contents

1. Using CARMA with RACE	1
1.1. Custom config options	1
1.1.1. The <code>--dynamic-only</code> option.....	1
1.1.2. The <code>--allow-multicast</code> option.....	1
1.1.3. The <code>--leader-relay-only</code> option.....	1
1.1.4. The <code>--mix-size</code> option.....	1
1.1.5. The <code>--inter-server-direct-only</code> option.....	2
1.1.6. The <code>--send-retry-count</code> option.....	2
1.1.7. The <code>--send-retry-window</code> option.....	2
1.1.8. The <code>--send-timeout</code> option.....	2
1.2. Release notes.....	3
2022-02-14: <code>carma:r20:prod</code> for RACE 2.0.0	3
2021-12-02: <code>carma:r42:prod</code> for RACE 1.6.0	3
2021-10-20: <code>carma:r263:prod</code> for RACE 1.5.0	3
2021-09-29: <code>carma:r257:prod</code> for RACE 1.5.0	4
2021-09-29: <code>carma:r255:prod</code> for RACE 1.5.0	4
2021-09-21: <code>carma:r32:prod</code> for RACE 1.5.0	5
2021-08-17: <code>carma:r56:prod</code> for RACE 1.4.0	5
2021-08-17: <code>carma:r41:prod</code> for RACE 1.4.0	5
2021-07-22: <code>carma:r215:prod</code> for RACE 1.3.0	6
2021-07-22: <code>carma:r213:prod</code> for RACE 1.3.0	6
2021-07-21: <code>carma:r210:prod</code> for RACE 1.3.0	6
2021-07-12: <code>carma:r199:prod</code> for RACE 1.3.0	7
2021-07-09: <code>carma:r150:prod</code> for RACE 1.3.0	7
2021-07-05: <code>carma:r93:prod</code> for RACE 1.3.0	7
2021-06-17: <code>carma:r4:prod</code> for RACE 1.3.0	8
2021-05-17: <code>carma:r230:prod</code> for RACE 1.1.5	8
2021-05-14: <code>carma:r216:prod</code> for RACE 1.1.5	8
2021-05-10: <code>carma:r199:prod</code> for RACE 1.1.5	9
2021-05-09: <code>carma:r190:prod</code> for RACE 1.1.5	9
2021-05-07: <code>carma:r160:prod</code> for RACE 1.1.5	9
2. Implementing a TA2 plugin using the C API.....	10
2.1. The <code>create_plugin_ta2</code> function.....	10
2.2. The <code>destroy_plugin_ta2</code> function.....	10
2.3. The <code>kestrel_ta2_plugin</code> structure	11
2.3.1. The plugin opaque pointer.....	13
2.3.2. The <code>name</code> string	13
2.3.3. The <code>activateChannel</code> function	13
2.3.4. The <code>closeConnection</code> function	13
2.3.5. The <code>createLink</code> function	13
2.3.6. The <code>deactivateChannel</code> function	14
2.3.7. The <code>destroyLink</code> function	14
2.3.8. The <code>flushChannel</code> function	14
2.3.9. The <code>init</code> function	14
2.3.10. The <code>loadLinkAddress</code> function	14
2.3.11. The <code>onUserAcknowledgementReceived</code> function	15
2.3.12. The <code>onUserInputReceived</code> function	15
2.3.13. The <code>openConnection</code> function	15
2.3.14. The <code>sendPackage</code> function	15
2.3.15. The <code>shutdown</code> function	15

1. Using CARMA with RACE

This repository provides information for all RACE performers about how to use CARMA with RACE.

1.1. Custom config options

These are the custom options that can be specified for CARMA using the `--tal-custom-args` option of the `rib deployment config generate` command:

1.1.1. The `--dynamic-only` option

`--dynamic-only[=<regex>]`

The `--dynamic-only` option specifies that all nodes whose names match `<regex>` will only use dynamic links for non-LT_RECV links, except for the purpose of sending messages to the registration servers. In other words, the matching nodes will never call the `getLinksForPersonas` function for LT_SEND or LT_BIDI links for nodes that are not registration servers.

Specifying this option without `<regex>`, i.e., as `--dynamic-only`, is equivalent to specifying `--dynamic-only=".*"`, which matches all nodes.

This option can be specified multiple times to specify multiple regular expressions to match against.

1.1.2. The `--allow-multicast` option

`--[no-]allow-multicast`

When enabled, multicast links may be used. When disabled, multicast links will not be used.

Default: disabled.

1.1.3. The `--leader-relay-only` option

`--[no-]leader-relay-only`

When enabled, only the leader of a management committee will send a message to the mailbox server of a recipient client. When disabled, every member of a management committee will send a message to the mailbox server of a recipient client.

Default: enabled.

1.1.4. The `--mix-size` option

--mix-size=<n>

Specifies the number of messages a management committee must receive before performing a mix, which causes the messages to be processed by the committee and sent to the recipient clients through their mailbox servers.

<n> must be at least 4 and must be a square.

Default: 4.

1.1.5. The **--inter-server-direct-only** option

--[no-]inter-server-direct-only

When enabled, all non-client nodes will only use direct links to send messages to each other. When disabled, all non-client nodes will use both direct and indirect links to send messages to each other.

Default: enabled.

1.1.6. The **--send-retry-count** option

--send-retry-count=<n>

Specifies the number of times a message will attempt to be resent.

<n> must be nonnegative.

Default: 0.

1.1.7. The **--send-retry-window** option

--send-retry-window=<n>

Specifies the maximum number of seconds to wait between message resend attempts. The actual number of seconds is probabilistically selected for each resend attempt.

<n> must be nonnegative.

Default: 180.

1.1.8. The **--send-timeout** option

--send-timeout=<n>

Specifies the `sendTimeout` value to use when calling the `openConnection` function of the RACE SDK.

<n> must be nonnegative.

Default: 600.

1.2. Release notes

2022-02-14: **carma:r20:prod** for RACE 2.0.0

Hi all, we've released **carma:r20:prod** for RACE 2.0.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-2.0.0-carma-r20>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-2.0.0-carma-r20>

The documentation is also attached to this message as a PDF.

- ▮ Integrated CARMA with RACE 2.0.0 and RiB 2.0.0.

2021-12-02: **carma:r42:prod** for RACE 1.6.0

Hi all, we've released **carma:r42:prod** for RACE 1.6.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.6.0-carma-r42>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.6.0-carma-r42>

The documentation is also attached to this message as a PDF.

- ▮ Integrated CARMA with RACE 1.6.0 and RiB 1.6.0.

2021-10-20: **carma:r263:prod** for RACE 1.5.0

Hi all, we've released **carma:r263:prod** for RACE 1.5.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.5.0-carma-r263>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.5.0-carma-r263>

The documentation is also attached to this message as a PDF.

- ▮ Adjusted all x86-64 builds so that CARMA will run on any x86-64 CPU without requiring any non-ubiquitous extensions such as AVX.

- ▮ No code has changed since the previous release.

2021-09-29: `carma:r257:prod` for RACE 1.5.0

Hi all, we've released `carma:r257:prod` for RACE 1.5.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.5.0-carma-r257>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.5.0-carma-r257>

The documentation is also attached to this message as a PDF.

- ▮ Fixed a minor bug where genesis would fail if the number of servers was too low. Genesis should now work again down to 7 servers.

2021-09-29: `carma:r255:prod` for RACE 1.5.0

Hi all, we've released `carma:r255:prod` for RACE 1.5.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.5.0-carma-r255>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.5.0-carma-r255>

The documentation is also attached to this message as a PDF.

- ▮ CARMA should now automatically recover from broken connections that are reported via the `CONNECTION_CLOSED` or `CONNECTION_INIT_FAILED` events. New connections should now be opened in response to these events.
- ▮ CARMA now has its first iteration of registration servers. The current behavior is that when a client node detects that it has no links to its mailbox server, it will create one or more dynamic links and send them to its mailbox server via the registration servers. Note that this only applies to `CT_INDIRECT` channels.
- ▮ As a summary of this update, CARMA should still function normally when given no genesis options, primarily relying on the `getLinksForPersonas` function to discover links. In order to test dynamic links (for a `CT_INDIRECT` channel), genesis options such as `--ta1-custom-args='--dynamic-only="client"'` (matches all clients) or `--ta1-custom-args='--dynamic-only="client-0000[1-3]"` (matches clients 1 to 3) can be used.
- ▮ Added the `--dynamic-only[=<regex>]` genesis option. Currently, this option makes the matching nodes only use dynamic links for non-`LT_RECV` links, except for the purpose of sending messages to the registration servers. In other words, the matching nodes will never call the `getLinksForPersonas` function for `LT_SEND` or `LT_BIDI` links for nodes that are not registration servers.

This option can be specified multiple times to specify multiple regular expressions to match

against. Omitting `=<regex>` is equivalent to specifying `".*"`, which will match all nodes.

Note that since RACE names its client nodes `race-client-XXXXX`, specifying `--dynamic-only="client"` is an easy way to match all client nodes.

NB: Only client nodes have dynamic link support at this time. Matching a non-client node will cause that node to be unable to send messages, which may cause dropped end-to-end messages.

- ▮ Replaced the `--mb-threshold`, `--mc-threshold`, `--mc-good-size`, and `--num-mc-groups` genesis options with the `--mb-count`, `--mc-count`, and `--mc-group-size` options. Also added the `--reg-count` genesis option to specify the expected number of servers that will become registration servers.

These options should usually be omitted so that they use their default values. The default values are documented in the attached PDF.

2021-09-21: `carma:r32:prod` for RACE 1.5.0

Hi all, we've released `carma:r32:prod` for RACE 1.5.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.5.0-carma-r32>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.5.0-carma-r32>
- ▮ Integrated with RACE 1.5.0 and RiB 1.5.0.

2021-08-17: `carma:r56:prod` for RACE 1.4.0

Hi all, we've released `carma:r56:prod` for RACE 1.4.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.4.0-carma-r56>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.4.0-carma-r56>
- ▮ OpenTracing has been re-enabled.

2021-08-17: `carma:r41:prod` for RACE 1.4.0

Hi all, we've released `carma:r41:prod` for RACE 1.4.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.4.0-carma-r41>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.4.0-carma-r41>

- ▮ Integrated CARMA with RACE 1.4.0 and RiB 1.4.0.
- ▮ Added the Android AArch64 build. The CARMA client and server plugins are now available on all three platforms: Linux x86-64, Android x86-64, and Android AArch64.
- ▮ OpenTracing is temporarily disabled due to interfacing difficulties introduced by RACE 1.4.0.

2021-07-22: `carma:r215:prod` for RACE 1.3.0

Hi all, we've released `carma:r215:prod` for RACE 1.3.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r215>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.3.0-carma-r215>
- ▮ Fixed a null pointer bug that was detected by an assertion inside CARMA during performance testing (`std::vector::data()` may return `nullptr` when `size()` is zero).

2021-07-22: `carma:r213:prod` for RACE 1.3.0

Hi all, we've released `carma:r213:prod` for RACE 1.3.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r213>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.3.0-carma-r213>
- ▮ Fixed an oversight where a multicast link that was initially discovered via a unicast `getLinksForPersonas()` call would never be amended to be marked multicast after learning more information about the link.
- ▮ Fixed an oversight where certain multicast send opportunities were being ignored when the actual receiver set is a strict superset of the desired receiver set.
- ▮ Fixed an unlikely bug where certain multicast sends could possibly send to an incorrect receiver set, i.e., a receiver set that is not a superset of the desired receiver set, which could cause a stalled mix.

2021-07-21: `carma:r210:prod` for RACE 1.3.0

Hi all, we've released `carma:r210:prod` for RACE 1.3.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r210>
- ▮ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.3.0-carma-r210>

- ‡ The `--no-allow-multicast` option now properly disables all multicast behavior, including disabling any multicast calls to the `getLinksForPersonas` SDK function.
- ‡ The input and output of all calls to the `getLinksForPersonas` SDK function are now logged.

2021-07-12: `carma:r199:prod` for RACE 1.3.0

Hi all, we've released `carma:r199:prod` for RACE 1.3.0.

- ‡ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r199>
- ‡ Documentation (for everyone): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/integration-info/-/tree/race-1.3.0-carma-r199>
- ‡ Changed the default `sendTimeout` value used when calling the `openConnection` function of the RACE SDK from 60 seconds to 600 seconds, and added the `--send-timeout=<n>` option to `generate_configs.sh` to control this value.
- ‡ Added the initial Android build.

2021-07-09: `carma:r150:prod` for RACE 1.3.0

Hi all, we've released `carma:r150:prod` for RACE 1.3.0.

- ‡ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r150>
- ‡ Fixed an issue where `generate_configs.sh` was requesting unnecessary links compared to what the nodes actually use.

2021-07-05: `carma:r93:prod` for RACE 1.3.0

Hi all, we've released `carma:r93:prod` for RACE 1.3.0.

- ‡ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r93>
- ‡ Added multiple management committees. Each committee maintains a separate mix counter, which has a side effect of making it more difficult to determine when a `ClrMsg` will make it through to the final recipient. We recommend sending 4 copies of every `ClrMsg` (the default mix size) to ensure that a mix always occurs when testing.
- ‡ The default `generate_configs.sh` settings should now be reasonable in arbitrarily large deployments, with the total number of links being reduced by up to a factor of 100.
- ‡ Multicast links will be used by certain parts of the CARMA protocol if available.
- ‡ Dynamic links are not being utilized yet.

- ▮ The `carma-draw` utility is being updated to accomodate the new changes. See the work in progress image attached.

2021-06-17: `carma:r4:prod` for RACE 1.3.0

Hi all, we've released `carma:r4:prod` for RACE 1.3.0.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.3.0-carma-r4>
- ▮ Basic integration with RACE 1.3.0.
- ▮ Not yet calling any new functions added since RACE 1.1.5.

2021-05-17: `carma:r230:prod` for RACE 1.1.5

Hi all, we've released `carma:r230:prod` for RACE 1.1.5.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.1.5-carma-r230>
- ▮ Updated `generate_configs.sh` to output more information for visualization purposes.
- ▮ Updated the node graph "jellyfish" visualization to show channel information and to make the client-mailbox mapping more visible for large deployments.
- ▮ This release does not affect plugin functionality.

2021-05-14: `carma:r216:prod` for RACE 1.1.5

Hi all, we've released `carma:r216:prod` for RACE 1.1.5.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.1.5-carma-r216>
- ▮ New `generate_configs.sh` option: `--mix-size=n`. Specifies the mix size. The default value is `--mix-size=4`. `n` must be at least 4 and must be a square.
- ▮ New `generate_configs.sh` option: `--mc-direct-only`. Specifies that only direct links should be used between management committee (MC) servers.
- ▮ New `generate_configs.sh` options: `--mb-threshold=a/b` and `--mc-threshold=a/b`. Specifies the fraction of servers that will be probabilistically selected as mailbox (MB) and management committee (MC) servers. The default values are `--mb-threshold=9/32` and `--mc-threshold=16/32`. `a` must be nonnegative, `b` must be a power of two, and the sum of both thresholds must be at most 1.
- ▮ Packages reported as `PACKAGE_RECEIVED` will now stop any further resend attempts. Packages reported as `PACKAGE_SENT` also currently stop any further resend attempts.

2021-05-10: **carma:r199:prod** for RACE 1.1.5

Hi all, we've released **carma:r199:prod** for RACE 1.1.5.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.1.5-carma-r199>
- ▮ Fixed a logging bug where **EncPkg** content hashes differed between senders and receivers.
- ▮ This release does not affect functionality, only logging correctness.

2021-05-09: **carma:r190:prod** for RACE 1.1.5

Hi all, we've released **carma:r190:prod** for RACE 1.1.5.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.1.5-carma-r190>
- ▮ Use any number of clients, at least 10 servers, and at least one indirect channel. The minimum number of servers is actually 6, but using fewer servers increases the probability that all server selection trials will fail during config generation.
- ▮ Fixed a bug that could cause the time field of a **ClrMsg** to be incorrect.
- ▮ Fixed a cross-node race condition that could cause **ClrMsg**'s to shadow each other, resulting in some **ClrMsg**'s being dropped and others being duplicated.
- ▮ Refined the tracing events for better log analysis, in particular regarding **EncPkg** routing.

2021-05-07: **carma:r160:prod** for RACE 1.1.5

Hi all, we've released **carma:r160:prod** for RACE 1.1.5.

- ▮ Source code (for those with access): <https://gitlab.race.twosixlabs.com/race-ta1-stealth/carma-source/-/tree/race-1.1.5-carma-r160>
- ▮ Use any number of clients, at least 10 servers, and at least one indirect channel. The minimum number of servers is 6, but using fewer servers increases the probability that all server selection trials will fail during config generation.
- ▮ OpenTracing is now (re)integrated into our updated MPC protocol.
- ▮ Links are now probabilistically prioritized by various metrics.
- ▮ Sends for which success has not been reported via the **onPackageStatusChanged** API will now be probabilistically attempted to be resent up to 5 times over 5 minutes, sometimes trying other links.
- ▮ An issue where messages over a certain size were immediately failing to send has been fixed. Messages up to 140 bytes in size are now supported again. Please note that persona names are currently limited to 32 bytes.

2. Implementing a TA2 plugin using the C API

To implement a TA2 plugin using the Kestrel C API, you must implement the `create_plugin_ta2` and `destroy_plugin_ta2` functions.

Kestrel will always call `plugin->shutdown` if you indicated success from `create_plugin_ta2`, even if the plugin is not initialized.

2.1. The `create_plugin_ta2` function

```
void create_plugin_ta2(  
    kestrel_ta2_sdk * sdk,  
    kestrel_ta2_plugin * plugin  
);
```

The `create_plugin_ta2` function creates an instance of the TA2 plugin, filling in the `kestrel_ta2_plugin` structure pointed to by `plugin` as follows:

- ▮ `plugin->plugin` should be an opaque pointer to the plugin instance.
- ▮ `plugin->name` should point to a null-terminated string that contains the plugin name.
- ▮ All function pointers in `*plugin` should point to the appropriate plugin functions.

The `sdk` parameter points to a `kestrel_ta2_sdk` structure that the plugin instance can use throughout its lifetime to make calls to Kestrel. The structure contains a set of function pointers but no opaque pointer to the SDK instance, as the `sdk` pointer itself serves as the opaque pointer. When the plugin instance calls an SDK function, it should always pass the `sdk` pointer as the first argument. For example, the plugin instance could call `sdk->get_entropy(sdk, p, n)` to fill `p` with `n` random bytes.

To indicate failure, the `create_plugin_ta2` function should set `plugin->plugin` to a null pointer. In this case, Kestrel will ignore all of the function pointers in `*plugin`, but Kestrel will still inspect `plugin->name`. If `plugin->name` is not a null pointer, it should point to an error message that explains why `create_plugin_ta2` failed. The error message should be null-terminated and encoded in UTF-8. Although failure is being indicated, Kestrel will still call `destroy_plugin_ta2(plugin)`, allowing any necessary cleanup to be performed on `plugin->name`.

Although the name `create_plugin_ta2` is used for this function throughout this manual, you should give this function a unique name of your own choosing to help prevent name collisions with other code. The usual approach is to use a unique prefix on this name (and any other names your plugin uses). For example, if your plugin were named `Foo`, you could use `foo_` as a unique prefix and name this function `foo_create_plugin_ta2`.

Note that the `create_plugin_ta2` function corresponds to the `createPluginTa2` function from the RACE C++ API.

2.2. The `destroy_plugin_ta2` function

```
void destroy_plugin_ta2(  
    kestrel_ta2_plugin * plugin  
);
```

The `destroy_plugin_ta2` function destroys an instance of the TA2 plugin. The `plugin` parameter points to a `kestrel_ta2_plugin` structure that was originally filled in by the `create_plugin_ta2` function.

Kestrel will always call the `destroy_plugin_ta2` function exactly once on a `kestrel_ta2_plugin` structure, even if the `create_plugin_ta2` function indicated failure by setting `plugin->plugin` to a null pointer. This allows the TA2 plugin to perform any necessary cleanup if it allocated an error message into `plugin->name`.

Although the name `destroy_plugin_ta2` is used for this function throughout this manual, you should give this function a unique name of your own choosing to help prevent name collisions with other code. The usual approach is to use a unique prefix on this name (and any other names your plugin uses). For example, if your plugin were named `Foo`, you could use `foo_` as a unique prefix and name this function `foo_destroy_plugin_ta2`.

Note that the `destroy_plugin_ta2` function corresponds to the `destroyPluginTa2` function from the RACE C++ API.

2.3. The `kestrel_ta2_plugin` structure

```
typedef struct kestrel_ta2_plugin kestrel_ta2_plugin;  
  
struct kestrel_ta2_plugin {  
  
    void * plugin;  
  
    kestrel_char * name;  
  
    kestrel_PluginResponse (*activateChannel)(  
        void * plugin,  
        kestrel_RaceHandle handle,  
        kestrel_char const * channelId,  
        kestrel_char const * roleName  
    );  
  
    kestrel_PluginResponse (*closeConnection)(  
        void * plugin,  
        kestrel_RaceHandle handle,  
        kestrel_char const * connectionId  
    );  
  
    kestrel_PluginResponse (*createLink)(  
        void * plugin,  
        kestrel_RaceHandle handle,  
        kestrel_char const * channelId  
    );  
  
    kestrel_PluginResponse (*deactivateChannel)(
```

```
void *          plugin,
kestrel_RaceHandle handle,
kestrel_char const * channelId
);

kestrel_PluginResponse (*destroyLink)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_char const * linkId
);

kestrel_PluginResponse (*flushChannel)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_char const * channelId,
uint64_t        batchSize
);

kestrel_PluginResponse (*init)(
void * plugin
);

kestrel_PluginResponse (*loadLinkAddress)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_char const * channelId,
kestrel_char const * linkAddress
);

kestrel_PluginResponse (*onUserAcknowledgementReceived)(
void *          plugin,
kestrel_RaceHandle handle
);

kestrel_PluginResponse (*onUserInputReceived)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_bool      answered,
kestrel_char const * response
);

kestrel_PluginResponse (*openConnection)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_LinkType linkType,
kestrel_char const * linkId,
kestrel_char const * linkHints,
int32_t          sendTimeout
);

kestrel_PluginResponse (*sendPackage)(
void *          plugin,
kestrel_RaceHandle handle,
kestrel_char const * connectionId,
```

```
uint8_t const *    pkg,  
uint64_t          pkgSize,  
uint64_t          timeoutTimestamp,  
uint64_t          batchId  
);  
  
void (*shutdown)(  
    void * plugin  
);  
  
};
```

The `kestrel_ta2_plugin` structure represents an instance of a TA2 plugin. The structure contains an opaque pointer to the plugin instance, a string pointer, and a set of pointers to the plugin functions. Kestrel stores one of these structures for each TA2 plugin instance it's managing and uses the function pointers to make calls to that plugin instance, always passing the opaque pointer as the first argument. For example, Kestrel will call `plugin->init(plugin->plugin)` to initialize a plugin instance.

2.3.1. The plugin opaque pointer

```
void * plugin;
```

2.3.2. The name string

```
kestrel_char * name;
```

2.3.3. The activateChannel function

```
kestrel_PluginResponse (*activateChannel)(  
    void * plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * channelGid,  
    kestrel_char const * roleName  
);
```

2.3.4. The closeConnection function

```
kestrel_PluginResponse (*closeConnection)(  
    void * plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * connectionId  
);
```

2.3.5. The createLink function

```
kestrel_PluginResponse (*createLink)(
```

```
void *          plugin,  
kestrel_RaceHandle handle,  
kestrel_char const * channelGid  
);
```

2.3.6. The deactivateChannel function

```
kestrel_PluginResponse (*deactivateChannel)(  
    void *          plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * channelGid  
);
```

2.3.7. The destroyLink function

```
kestrel_PluginResponse (*destroyLink)(  
    void *          plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * linkId  
);
```

2.3.8. The flushChannel function

```
kestrel_PluginResponse (*flushChannel)(  
    void *          plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * channelGid,  
    uint64_t         batchId  
);
```

2.3.9. The init function

```
kestrel_PluginResponse (*init)(  
    void * plugin  
);
```

2.3.10. The loadLinkAddress function

```
kestrel_PluginResponse (*loadLinkAddress)(  
    void *          plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * channelGid,  
    kestrel_char const * linkAddress  
);
```


2.3.11. The `onUserAcknowledgementReceived` function

```
kestrel_PluginResponse (*onUserAcknowledgementReceived)(  
    void * plugin,  
    kestrel_RaceHandle handle  
);
```

2.3.12. The `onUserInputReceived` function

```
kestrel_PluginResponse (*onUserInputReceived)(  
    void * plugin,  
    kestrel_RaceHandle handle,  
    kestrel_bool answered,  
    kestrel_char const * response  
);
```

2.3.13. The `openConnection` function

```
kestrel_PluginResponse (*openConnection)(  
    void * plugin,  
    kestrel_RaceHandle handle,  
    kestrel_LinkType linkType,  
    kestrel_char const * linkId,  
    kestrel_char const * linkHints,  
    int32_t sendTimeout  
);
```

2.3.14. The `sendPackage` function

```
kestrel_PluginResponse (*sendPackage)(  
    void * plugin,  
    kestrel_RaceHandle handle,  
    kestrel_char const * connectionId,  
    uint8_t const * pkg,  
    uint64_t pkgSize,  
    uint64_t timeoutTimestamp,  
    uint64_t batchId  
);
```

2.3.15. The `shutdown` function

```
void (*shutdown)(  
    void * plugin  
);
```