WNDRVR
SUPPORT NETWORK

WIND RIVER HELIX DEVICE CLOUD WRA API REFERENCE, 1.2

Corporate Headquarters

Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.
Toll free (U.S.A.): +1-800-545-WIND
Telephone: +1-510-748-4100
Facsimile: +1-510-749-2010

For additional contact information, see the Wind River website:

www.windriver.com

For information on how to contact Customer Support, see:

www.windriver.com/support

Wind River Helix Device Cloud WRA API Reference, 1.2

29 January 2019

1. WIND RIVER HELIX DEVICE CLOUD WRA API REFERENCE, 1.2

2. WRA

Name

wra -

Layer

WRA

Routines

wra_gethandle( ) on page - Get a handle to the WR Agent. wra_tm_create( ) on page - Allocate a telemetry object and initialize it. wra_tm_destroy( ) on page - Free the contents of a telemetry object. wra_tm_reset( ) on page - Reset a telemetry object with default attributes. wra_tm_subscribe( ) on page - Subscribe to data telemetry. wra_tm_setvalue_string( ) on page - Set the textual value of a telemetry object wra_tm_getvalue_string( ) on page - Get the textual value of a telemetry object wra_tm_setvalue_int( ) on page - Set the integer value of a telemetry object wra_tm_getvalue_int( ) on page - Get the integer value of a telemetry object attribute. wra_tm_setvalue_bool( ) on page - Set the boolean value of a telemetry object wra_tm_getvalue_bool( ) on page - Get the boolean value of a telemetry object wra_tm_setvalue_double( ) on page - Set the double value of a telemetry object wra_tm_getvalue_double( ) on page - Get the double value of a telemetry object wra_tm_setaux( ) on page - Set the auxiliary telemetry information of a telemetry object wra_tm_getaux( ) on page - Get the auxiliary telemetry information of a telemetry object wra_tm_settimestamp( ) on page - Set the telemetry object time stamp. wra_tm_gettimestamp( ) on page - Get the telemetry object time stamp. wra_tm_post( ) on page - Send a telemetry object to the server. wra_tm_post_default( ) on page - Send a telemetry object to the server. wra_reg_app_ex_handler(DEPRECATED)( ) - Register an application wra_action_subscribe( ) on page - Subscribe to an action. wra_action_unsubscribe( ) on page - Unsubscribe an application action handler. wra_action_wait( ) on page - Wait for action requests from the server. wra_file_download( ) on page - Subscribe to a file download from the server. wra_file_upload( ) on page - Upload the specified file to server. wra_delete_handle( ) on page - wra context from user space.

Description

Wind River HDC Agent API

Include files

wra_types.h

Wra_gethandle( )

Name

wra_gethandle( ) - Get a handle to the WR Agent.

Layer

WRA

Synopsis

```
wra_handle wra_gethandle
    (
```

```
    void
    );
```

Description

Call this function to get a calling context to access the WR Agent.

Erron

WRA_NULL if the Agent is un-initialized.

Returns

A handle to the WR Agent API

Errno

Not Available

See also

wra on page

Wra_tm_create( )

Name

wra_tm_create( ) - Allocate a telemetry object and initialize it.

Layer

WRA

Synopsis

```
wra_tm_handle wra_tm_create
    (
    const char *tmtype,
    const char *tmname
    );
```

Description

Allocate a telemetry object that can then be used in subsequent calls to the wra_tm_* interface functions to set and get its attributes. Calling this function allocates memory from the heap to store the object. It is not necessary to call wra_tm_init after calling this function.

Parameters:

const char *tmtype

[in] The type of telemetry.

const char *tmname

[in] The name of the telemetry object.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_tm_destroy( )

Name

wra_tm_destroy( ) - Free the contents of a telemetry object.

Layer

WRA

Synopsis

```
wra_status wra_tm_destroy
    (
    wra_tm_handle tm
    );
```

Description

Free the contents of a telemetry object when it is no longer needed.

Parameters:

wra_tm_handle tm

[in] A telemetry object whose contents to free.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_tm_reset( )

Name

wra_tm_reset( ) - Reset a telemetry object with default attributes.

Layer

WRA

Synopsis

```
wra_tm_handle wra_tm_reset
    (
    wra_tm_handle tm,
    const char    *tmname
    );
```

Description

Reset a telemetry object that can then be used in subsequent calls to the wra_tm_* interface functions to set and get its attributes.

Parameters:

tm

[out] A telemetry object to be initialized.

tmname

[in] The name of the telemetry object.

Returns

A valid handle to the reset telemetry object if successful.

Errno

WRA_NULL otherwise

See also

wra on page

Wra_tm_subscribe( )

Name

wra_tm_subscribe( ) - Subscribe to data telemetry.

Layer

WRA

Synopsis

```
wra_status wra_tm_subscribe
    (
    wra_handle    wra_h,
    wra_tm_handle tm,
    const char    *tmname,
    wra_timestamp *tmo
    );
```

Description

Register to receive incoming telemetry data from the server.

Parameters:

wra_handle wra_h

[in] A valid handle to the WR Agent.

wra_tm_handle tm

[in] A valid, initialized handle to a telemetry object. (Only data telemetry is support)

const char *tmname

[in] The name of the telemetry object.

wra_timestamp *tmo

[in] A time out value to specify how long to wait without receiving a telemetry object. A **NULL** value specifies that the application will wait indefinitely.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_tm_setvalue_string( )

Name

wra_tm_setvalue_string( ) - Set the textual value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_setvalue_string
    (
    wra_tm_handle tm,
    const char    *attr,
    const char    *value
    );
```

Description

attribute.

Set the textual value of the specified attribute in a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] The attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_getvalue_string( )

Name

wra_tm_getvalue_string( ) - Get the textual value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_getvalue_string
    (
    wra_tm_handle tm,
    const char    *attr,
    char          **value
    );
```

Description

Get the textual value of the specified attribute from a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to get.

value

[out] A reference to the attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_setvalue_int( )

Name

wra_tm_setvalue_int( ) - Set the integer value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_setvalue_int
    (
    wra_tm_handle tm,
    const char    *attr,
    int           value
    );
```

Description

Set the 32-bit unsigned value of the specified attribute in a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] The attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_getvalue_int( )

Name

wra_tm_getvalue_int( ) - Get the integer value of a telemetry object attribute.

Layer

WRA

Synopsis

```
wra_status wra_tm_getvalue_int
    (
    wra_tm_handle tm,
    const char    *attr,
    int           *value
    );
```

Description

Get the 32-bit unsigned value of the specified attribute from a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to get.

value

[out] A reference to the attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_setvalue_bool( )

Name

wra_tm_setvalue_bool( ) - Set the boolean value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_setvalue_bool
    (
    wra_tm_handle tm,
    const char    *attr,
    bool          value
    );
```

Description

Set the boolean value of the specified attribute in a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] The attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_getvalue_bool( )

Name

wra_tm_getvalue_bool( ) - Get the boolean value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_getvalue_bool
    (
    wra_tm_handle tm,
    const char    *attr,
    bool          *value
    );
```

Description

Get the boolean value of the specified attribute from a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to get.

value

[in] A reference to the attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_setvalue_double( )

Name

wra_tm_setvalue_double( ) - Set the double value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_setvalue_double
    (
    wra_tm_handle tm,
    const char    *attr,
    double        value
    );
```

Description

Set the double value of the specified attribute in a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] The attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_getvalue_double( )

Name

wra_tm_getvalue_double( ) - Get the double value of a telemetry object

Layer

WRA

Synopsis

```
wra_status wra_tm_getvalue_double
    (
    wra_tm_handle tm,
    const char    *attr,
    double        *value
    );
```

Description

Get the double value of the specified attribute in a telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] A reference to the attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_setaux( )

Name

wra_tm_setaux( ) - Set the auxiliary telemetry information of a telemetry object attribute.

Layer

WRA

Synopsis

```
wra_status wra_tm_setaux
    (
    wra_tm_handle tm,
    const char    *attr,
    wra_tm_handle aux
    );
```

Description

The value must resolve to a valid type expected by the attribute, or the operation will not complete successfully.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] A reference to the attribute value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_getaux( )

Name

wra_tm_getaux( ) - Get the auxiliary telemetry information of a telemetry object attribute.

Layer

WRA

Synopsis

```
wra_status wra_tm_getaux
    (
    wra_tm_handle tm,
    const char    *attr,
    wra_tm_handle aux
    );
```

Description

.

The calling function must provide an initialized value that is expected by the attribute, or the operation will not complete successfully.

Parameters:

tm

[in] A valid, initialized telemetry object.

attr

[in] The attribute to set.

value

[in] The attribute value

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_settimestamp( )

Name

wra_tm_settimestamp( ) - Set the telemetry object time stamp.

Layer

WRA

Synopsis

```
wra_status wra_tm_settimestamp
    (
    wra_tm_handle tm,
    wra_timestamp *timestamp
    );
```

Description

Set the time stamp of the specified telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

tstamp

[in] A valid, initialized time stamp.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_gettimestamp( )

Name

wra_tm_gettimestamp( ) - Get the telemetry object time stamp.

Layer

WRA

Synopsis

```
wra_status wra_tm_gettimestamp
    (
    wra_tm_handle tm,
    wra_timestamp *timestamp
    );
```

Description

Get the time stamp of the specified telemetry object.

Parameters:

tm

[in] A valid, initialized telemetry object.

tstamp

[out] A reference to the time stamp value.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_tm_post( )

Name

wra_tm_post( ) - Send a telemetry object to the server.

Layer

WRA

Synopsis

```
wra_status wra_tm_post
    (
    wra_handle              wra_h,
```

```
    wra_tm_handle           tm,
    wra_service_handle      wra_service_h,
    wra_notification_handle wra_notification_h
    );
```

Description

Transmit the telemetry data by way of the WR Agent.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

tm

[in] The address of the telemetry object.

wra_service_h

[in] A valid service identification handle.

wra_notification_h

[in] A valid notification handle.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_NO_MEMORY if internal resources are exhausted **WRA_ERR_EAGAIN** if the agent is not ready **WRA_ERR_FULL** if the transmission queue is full **WRA_ERR_FAILED** otherwise

See also

wra on page

Wra_tm_post_default( )

Name

wra_tm_post_default( ) - Send a telemetry object to the server.

Layer

WRA

Synopsis

```
void wra_tm_post_default
      (
      wra_h,
```

```
        tm
        )
```

Description

Transmit the telemetry by way of the WR Agent.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

tm

[in] The address of the telemetry object.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_NO_MEMORY if internal resources are exhausted **WRA_ERR_EAGAIN** if the agent is not ready **WRA_ERR_FULL** if the transmission queue is full **WRA_ERR_FAILED** otherwise

See also

wra on page

wra_reg_app_ex_handler(DEPRECATED)( )

name

wra_reg_app_ex_handler(DEPRECATED)( ) - Register an application execution handler.

Layer

WRA

Synopsis

```
int wra_reg_app_ex_handler
    (
    wra_handle         wra_h,
    wra_app_ex_handler app_ex_handler,
    const char         *appname
    );
```

Description

Call this routine to register a handler for an application execution action that is sent from the server. When the server sends an application execution action to the device, it contains an "appname" and an "args" string. The agent spawns a task that invokes the handler registered using this API against the appname. One handler can be registered against more than one appname. You can register handlers against a maximum of 128 unique appnames.

The maximum length of appname is 32 characters.

It is up to the handler to parse the arguments. The "appname" and "args" parameters are not guaranteed to be valid after the handler returns.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

app_ex_handler

[in] Application execution handler to register.

appname

[in] Application name to register the handler against.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_action_subscribe( )

Name

wra_action_subscribe( ) - Subscribe to an action.

Layer

WRA

Synopsis

```
int wra_action_subscribe
    (
    wra_handle         wra_h,
    wra_app_ex_handler app_ex_handler,
    const char         *appname
    );
```

Description

Call this routine to register a handler for an action sent from the server. When the server sends an action to the device, it contains an "appname" and an "args" string. The agent spawns a task that invokes the handler registered the appname. One

handler can be registered against more than one appname. You can register handlers against a maximum of 128 unique appnames.

The maximum length of the appname is 32 characters.

It is up to the handler to parse the arguments. The "appname" and "args" parameters are not guaranteed to be valid after the handler returns.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

app_ex_handler

[in] Application action handler to register.

appname

[in] Application name to register the handler against.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_action_unsubscribe( )

Name

wra_action_unsubscribe( ) - Unsubscribe an application action handler.

Layer

WRA

Synopsis

```
wra_status wra_action_unsubscribe
    (
    wra_handle  wra_h,
    const char* appname
    );
```

Description

The application handler registered against the specified appname is removed, and the application will no longer receive action notifications from the server for the specified appname.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

appname

[in] Application name to unsubscribe from.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise

See also

wra on page

Wra_action_wait( )

Name

wra_action_wait( ) - Wait for action requests from the server.

Layer

WRA

Synopsis

```
wra_status wra_action_wait
    (
    wra_handle    wra_h,
    wra_timestamp *tm
    );
```

Description

Call this function after registering a handler for an action using the wra_action_subscribe function. The application blocks until it receives an action request from server. It executes the action handler in the application context user space or in a dedicated agent thread in case of flat memory. Alternately, the application can specify a timeout, and the function returns if the application does not receive an action before the timeout expires.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

time_out

[in] A **NULL** value specifies that application will block until it receives action request from the server. Alternatively, the application can specify how long to wait for an action request.

Returns

WRA_SUCCESS - if the registered action was successfully executed. **WRA_ERR_EAGAIN** - if a timeout was specified and no action was received within the specified time period.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page

Wra_file_download( )

Name

wra_file_download( ) - Subscribe to a file download from the server.

Layer

WRA

Synopsis

```
wra_status  wra_file_download
    (
    wra_handle    wra_h,
    const char    *filename,
    const char*   dir,
    wra_timestamp *time_out
    );
```

Description

Wait for the server to download the specified file. Optionally, specify how long to wait for the file.

Parameters:

wra_h

[in] A valid handle to WR Agent.

filename

[in] name of the file to download.

dir

[in] The absolute path of the directory where the file is copied when it is received from the server. If a **NULL** value is specified, the file is copied to a preconfigured directory. The agent configuration file specifies the preconfigured directory.

time_out

[in] A **NULL** value specifies that the application will block until it receives the file. Alternatively, the application can specify a timeout, and the function returns if the application does not receive the file before the timeout expires.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_BAD_PARAM - file name or path is too long - destination directory does not exist **WRA_ERR_ETIMEDOUT** - operation timed out **WRA_ERR_FAILED** otherwise

See also

wra on page

Wra_file_upload( )

Name

wra_file_upload( ) - Upload the specified file to server.

Layer

WRA

Synopsis

```
wra_status  wra_file_upload
    (
    wra_handle    wra_h,
    const char    *filepath,
    wra_timestamp *time_out
    );
```

Description

Upload a file to the server. Specify the absolute path of the file name.

Parameters:

wra_h

[in] A valid handle to WR Agent.

filepath

[in] Absolute path of the file from where agent will upload it to server.

time_out

[in] A **NULL** value specifies that the application will block until the agent finishes uploading the file. Alternatively, the application can specify a timeout, and if the file is not uploaded before the timeout expires, the function return code indicates that the file failed to upload due to timing out.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_BAD_PARAM - file does not exist **WRA_ERR_ETIMEDOUT** - operation timed out **WRA_ERR_FAILED** otherwise

See also

wra on page

Wra_delete_handle( )

Name

wra_delete_handle( ) - wra context from user space.

Layer

WRA

Synopsis

```
int wra_delete_handle
    (
    wra_handle wra_h
    );
```

Description

Call this function to cleanup the agent resources (such as queues and subscriptions) during the application life cycle.

Parameters:

wra_h

[in] A valid handle to the WR Agent.

Returns

WRA_SUCCESS if the operation was successful.

Errno

WRA_ERR_FAILED otherwise.

See also

wra on page