

```
'),o.close()>("https://assets.zendesk.com/embeddable_framework/main.js","numxl.zendesk.com"); /*]]>*/
```

## RETURN CODES

Last Modified on 03/23/2016 4:00 pm CDT

- [C/C++](#)
- [.Net](#)

The NumXL different functions attempt to return the most relevant status code to the operation. In C/C++, the status codes (e.g. error codes) are defined in **SFMacros.h** file as preprocessor macros.

By definition, the zero value is used for success (NDK\_SUCCESS), and negative numbers for error codes (e.g. NDK\_FAILED). This leaves us with the flexibility to use positive numbers for additional success information (e.g. warning codes)

```
#define NDK_SUCCESS 0 ///< SUCCESS return code.
#define NDK_FAILED -1 ///< FAILED: Internal error occurred

#define NDK_SDK_UNINITIALIZED -10 ///< FAILED: The API NDK_INIT has not
yet been called
#define NDK_LOG_UNINITIALIZED -11 ///< FAILED: The API NDK_LOG_INIT has
not yet been called
#define NDK_LUC_UNINITIALIZED -12 ///< FAILED: The API NDK_LUC_INIT has
not yet been called
#define NDK_DBM_UNINITIALIZED -13 ///< FAILED: The API NDK_DBM_INIT has
not yet been called
#define NDK_MSG_UNINITIALIZED -14 ///< FAILED: The API NDK_MSG_INIT has
not yet been called

#define NDK_LOG_INIT_FAILED -20 ///< FAILED: The logging system failed
during initialization, check the configuration settings
#define NDK_DB_INIT_FAILED -21 ///< FAILED: Missing or failed to open
the database file
#define NDK_LUC_INIT_FAILED -22 ///< FAILED: Missing or failed to open
the database file

// Initialization error codes
#define NDK_MISSING_CONF -100 ///< FAILED: The configuration file
is missing
#define NDK_BAD_CONF -101 ///< FAILED: Access denied or
corrupted file
#define NDK_CONF_DATAPATH_INVALID -102 ///< FAILED: Invalid datapath value
in the configuration file
#define NDK_DATAPATH_INVALID -103 ///< FAILED: failed to
```

```

retrieve/construct a temp path for logs and intermediate calculation
#define NDK_CONF_PRODID_INVALID      -104  ///< FAILED: Invalid value for
[GLOBALS][PRODUCTID] entry in the conf file
#define NDK_LOGFILE_INUSE            -105  ///< FAILED: Failed to open the
logfile for writing (permission error or file in use)
#define NDK_MISSING_APP_ARG          -106  ///< FAILED: invalid or Null argument
(e.g. AppName for return value)
#define NDK_MISSING_LICENSE_KEY      -107  ///< FAILED: The product license ket
is invalid
#define NDK_INVALID_LICENSE_KEY      -108  ///< FAILED: The product license ket
is invalid
#define NDK_INACTIVE_LICENSE_KEY     -109  ///< FAILED: The license key has yet
to be activated
#define NDK_INVALID_KEY_CODE         -110  ///< FAILED: The license key and code
are not valid
#define NDK_EXPIRED_LICENSE_KEY      -111  ///< FAILED: The license key has
expired
#define NDK_LOW_LICENSE_LEVEL        -112  ///< FAILED: The required license
level is not met by current license

// Runtime error codes
#define NDK_INVALID_ARG              -300  ///< FAILED: an input argument with
unexpected or invalid value.
#define NDK_LENGTH_ERROR             -301  ///< FAILED: The user's buffer is not
big enough or Insufficient input data
#define NDK_INVALID_VALUE            -302  ///< FAILED: Invalid value of an
argument
#define NDK_EMPTY_TIME_SERIES        -303  ///< FAILED: number of non-missing
values is zero
#define NDK_ZERO_INVALID_VARIANCE    -304  ///< FAILED: number of non-missing
values is zero
#define NDK_CALIBRATION_ERROR        -305  ///< FAILED: The Optimizer failed to
converge to a unique solution.
#define NDK_INVALID_MODEL            -306  ///< FAILED: The model's parameters
values did not pass the stability test.
#define NDK_INSUFFICIENT_OBS         -307  ///< FAILED: The number of non-missing
observations is not sufficient to carry on the requested function

// Implementation status
#define NDK_NOTSUPPORTED             -400  ///< FAILED: The required operation is
not currently implemented/supported

// Warnings codes
#define NDK_RET_NAN                  100  ///< WARNING: The function returns an
invalid (i.e. missing) value

```

```

#define NDK_SKIP_INIT                105 ///< WARNING: The DLL is already
initialize, skipping !

#define NDK_KEY_IN_GRACE_PERIOD      1000 ///< INFORMATION: the trial license
key is in the 7-day grace period
#define NDK_KEY_IN_TRIAL_PERIOD      1005 ///< INFORMATION: the trial license
key is in the free trial period
#define NDK_KEY_NOT_IN_TRIAL_PERIOD  1010 ///< INFORMATION: the trial license
key is not in the free trial period
#define NDK_PERP_KEY_ACTIVE          1015 ///< INFORMATION: the perpetual
license key is activated
#define NDK_PERP_KEY_INACTIVE        1020 ///< INFORMATION: the perpetual
license key is not activated
#define NDK_SUB_KEY_ACTIVE           1025 ///< INFORMATION: the subscription
license key is activated
#define NDK_SUB_KEY_INACTIVE         1030 ///< INFORMATION: the subscription
license key is not activated

```

The functions in .Net Wrapper attempt to return the most relevant status code to the operation. The Wrapper library define an enumeration type - NDK\_RETCODE (in NumXLAPI namespace) to capture those values returned by NumXL C- APIs.

```

namespace NumXLAPI
{
    ///< <summary>
    ///< Defines the possible API return values.
    ///< </summary>
    public enum NDK_RETCODE
    {
        ///< <summary> SUCCESS </summary>
        NDK_SUCCESS =                0,
        ///< <summary> FAILED: Internal error occured </summary>
        NDK_FAILED =                  -1,

        // TRUE/FALSE aliases
        ///< <summary> SUCCESS: return value is TRUE. </summary>
        NDK_TRUE=                      0,
        ///< <summary> SUCCESS: return value is FALSE. </summary>
        NDK_FALSE =                    1,
        ///< <summary> FAILED: The API NDK_INIT has not yet been called. </summary>
        NDK_SDK_UNINITIALIZED =        -10,
        ///< <summary> FAILED: The API NDK_INIT has not yet been called</summary>
        NDK_LOG_UNINITIALIZED=         -11,
    }
}

```

```
///
```

```

// Runtime error codes
///

```

```
    NDK_PERP_KEY_ACTIVE      =      1015,  
    ///activated</summary>  
    NDK_PERP_KEY_INACTIVE =      1020,  
    ///activated</summary>  
    NDK_SUB_KEY_ACTIVE=          1025,  
    ///</summary>  
    NDK_SUB_KEY_INACTIVE=      1030,  
    }  
}
```

## See Also

[template("related")]