

# Internet ToolKit

## Versions 4.3, 5 & 6

Release notes

July 13, 2020

Published by e-Node worldwide  
[www.e-node.net/itk](http://www.e-node.net/itk)

### System requirements

As of version 4.3, Internet ToolKit comes through 3 different channels, each matching a 4D version range and SSL context.

- **Versions 4.3 and 4.x above** for use with 4D v14.4-4D v15 (including 4D v15R4), includes openssl 1.1.1d
- **Version 5.x** for use with 4D v16 (including v15R5) - 4D v17 (including 4D v17R4)  
64-bit only version 5.x available for 4D 17.4 and above
- **Version 6.x** for use with 4D v18 and above, 64-bit only, does not contain openssl (uses what is provided by 4D)

64-bit versions (5.x 64 and 6.x) are signed and notarized for macOS 10.15 and above.

Version 4 license keys will work on versions 5 and 6.

ITK requires MacOS 10.7.5 or higher and Windows 7 or better.

### Versions 4.3.3, 5.0.3 & 6.0.3 – New Commands

- `ITK_UDPRcvBlob`
- `ITK_UDPSendBlob`

### Versions 4.3.2, 5.0.2 & 6.0.2 – Change

- 64-bit versions (5.x 64 and 6.x) are signed and notarized for macOS 10.15 and above.

### Versions 4.3.1, 5.0.1 & 6.0.1 – Change

- Fixed a serious bug with SSL streams when using `ITK_TCPUnRcv` or non-empty endString with `ITK_TCPRcv/ITK_TCPRecvBlob`

## Versions 4.3, 5 & 6 – Changes

- *ITK\_TCPRecvBlob* now returns control immediately when using timeout 0 - previously incorrectly interpreted as "no timeout"
- Removed calls to SSL use certificate file with "ITKccert.pem" and "ITKccert.pfx" in *ITK\_TCPOpen*
- Removed calls to SSL CTX use certificate chain file/SSL use certificate file with "ITKcert.pem" in *ITK\_TCPListen*

Many internal changes due to upgrading the openssl version.

Some algorithms are unsupported in recent openssl.

- SHA digest is unsupported, SHA1 is used instead (*ITK\_Digest@*)
- SSLv2 and some other algorithms are not supported (read [openssl pages](#) to find out what is supported by version 1.1.1)

## Versions 4.3, 5 & 6 – New Features

- Added constants for protocol selection:
  - kITKSSLNoTLSv1\_2:65536:L
  - kITKSSLNoTLSv1\_3:131072:L
  - kITKSSLNoDTLSv1:262144:L
  - kITKSSLNoDTLSv1\_2:524288:L
- Old protocol constants disable lower-level protocols:
  - kITKSSLNoSSLv2 does nothing, openssl 1.1 does not contain SSLv2 code
  - kITKSSLNoTLSv1 disables TLSv1
  - kITKSSLNoTLSv1\_1 disables TLSv1 and TLSv1\_1
- Added constants do not disable lower-level protocols:
  - kITKSSLNoTLSv1\_2 disables only TLSv1\_2
  - kITKSSLNoTLSv1\_3 disables only TLSv1\_3
  - kITKSSLNoDTLSv1 disables only DTLSv1
  - kITKSSLNoDTLSv1\_2 disables only DTLSv1\_2

New feature: added support for SNI (a TLS extension) for *ITK\_TCPListen*

- Added command *ITK\_SSLSetSNICert*(&L;&T;&T;&T;&T;&T):L
  - \$1 - integer = the id, use zero for default certificate (used automatically when *ITK\_TCPListen* is called), use -1 to clear all entries in the map
  - \$2 - text = domain, domain to match with SNI request from client (the name has to exactly match the request from client), use empty string to clear entry with provided \$1 id
  - \$3 - text = path to the certificate file (PEM format)
  - \$4 - text = path to the private key file (PEM format)

- \$5 - text = optional password for the private key file
- \$6 - text = optional path to the chained certificate file (PEM)
- \$0 - integer = result
  - 0 = OK
  - 1 = entry does not exist (when trying to delete some entry)
  - -3 = bad certificate
  - -4 = bad private key/password
- when called, new CTX is created and the certificate is loaded. If OK, the CTX is stored into the SNI map and will be used later
- you can clear some entry by calling it with empty \$2 or \$3 or \$4
- don't forget that clearing the entries has no effect on already connected streams — and clearing entry with id = 0 has no effect on currently running listeners
- Added command to get currently defined domains in the SNI map  
*ITK\_SSLGetSNIDomains(&Y;&Y):L*
  - \$1 - integer or longint array = the id (values of \$1 from calls to *ITK\_SSLSetSNICert*)
  - \$2 - text array = domain (values of \$2 from calls to *ITK\_SSLSetSNICert*)
  - \$0 - integer = result:
    - -1 if the arrays are not of the proper type
    - number of entries otherwise
- Added constants for SNI support (to be used in parameter \$5 when calling *ITK\_TCPListen*):
  - *kITKSSLUseSNI:1048576:L*
  - *kITKSSLUseSNIWithError:2097152:L*
- Added new parameter to *ITK\_TCPListen*
  - \$8 - integer = the SNI map entry ID to use when using *kITKSSLUseSNI* or *kITKSSLUseSNIWithError* in tcpOpt (\$5)
  - when \$5 contains *kITKSSLUseSNI* or *kITKSSLUseSNIWithError*, SNI callback is installed
  - when \$5 contains *kITKSSLUseSNI* or *kITKSSLUseSNIWithError* and entry with id = \$8 exists in SNI map, that certificate is set to the stream, you don't have to call *ITK\_SSLSetCert*
  - when *ITK\_TCPListen* is called with *kITKSSLUseSNI* and client asks for a hostname not found in the SNI map, ITK ignores it (already set certificate is kept)
  - when *ITK\_TCPListen* is called with *kITKSSLUseSNIWithError* and client asks for a hostname not found in the SNI map, ITK returns an error to the client
  - Note: when both *kITKSSLUseSNI* & *kITKSSLUseSNIWithError* are used, the behavior for *kITKSSLUseSNIWithError* is used by ITK
  - -> by definition, 4D provides zero to the plugin if you don't specify this parameter -> thus it still looks for ID = 0 in the SNI map by default

SNI implementation is really simple:

- Use *ITK\_SSLSetSNICert* with \$id = 0 for the "master" certificate to be used with *ITK\_TCPListen*
- Add other certificates (with different \$id) to be used when client asks for different server
- Then call *ITK\_TCPListen* specifying kITKSSLUseSNI
- When kITKSSLUseSNI is specified in call to *ITK\_TCPListen*, callback for SNI is installed
  - when SNI with id = 0 exists, it is used
  - otherwise you have to call *ITK\_SSLSetCert* as usual
  - when SNI callback is made by openssl, the requested server name is searched in domain names in the SNI map - when found, it is used
  - server names are exactly matched (using strcmp), there is no support for sub-domain match
  - could be ITK has to support wildcards ("\*.domain.com" is not handled - exact match is used; workaround: use separate \$id for "xxx.domain.com" and "yyy.domain.com" with identical certificates)

### Versions 4.3, 5 & 6 – Fix

- Fixed *ITK\_TCPChRcv*

### Version 4.2.1

- Fixed a bug from ITK v2.6.1 - *ITK\_TCPRcvBlob* waits for the requested number of bytes
- Fixed a serious bug in *ITK\_TCPReceiveFile* & *ITK\_TCPRecvBlob* when option to not release the stream is not used - could crash later when using the same stream

### Version 4.1 & 4.2

- Changed parameter handling in *ITK\_TCPListen* (parameter 6 was incorrectly used as localPort instead of parameter 7; parameter 6 is now unused)
- Added one call to SSL to use provided password: uses the password when PEM private key is encrypted
- Added constants for SSL protocol options:
  - "kITKSSLNoSSLv2:4096:L",
  - "kITKSSLNoSSLv3:8192:L",
  - "kITKSSLNoTLSv1:16384:L",
  - "kITKSSLNoTLSv1\_1:32768:L",

To be used in calls to *ITK\_TCPOpen* and *ITK\_TCPListen* in the tcpOpt parameter.

Disabling any protocol version disables all older versions. E.g. when you set kITKSSLNoSSLv3 to disable SSLv3, you also disable SSLv2.

Warning: the version of OpenSSL used depends on the OS version (Mac) or 4D version (Win). If you disable kITKSSLNoTLSv1\_1 and your current OpenSSL does not implement TLSv1\_2, there is **no** protocol to be used...

- Fixed a possible crash in SSL

### Upgrading from previous versions

ITK versions 4.3, 5 & 6 are free upgrade from v4.0: existing v4.x licenses will activate any version as of 4.0.

### Contact

Free upgrade requests, sales and license information: [orders@e-node.net](mailto:orders@e-node.net)

Technical issues, discussions and requests: [forums.e-node.net/viewforum.php?f=8](http://forums.e-node.net/viewforum.php?f=8)

ITK web page: [www.e-node.net/itk](http://www.e-node.net/itk)