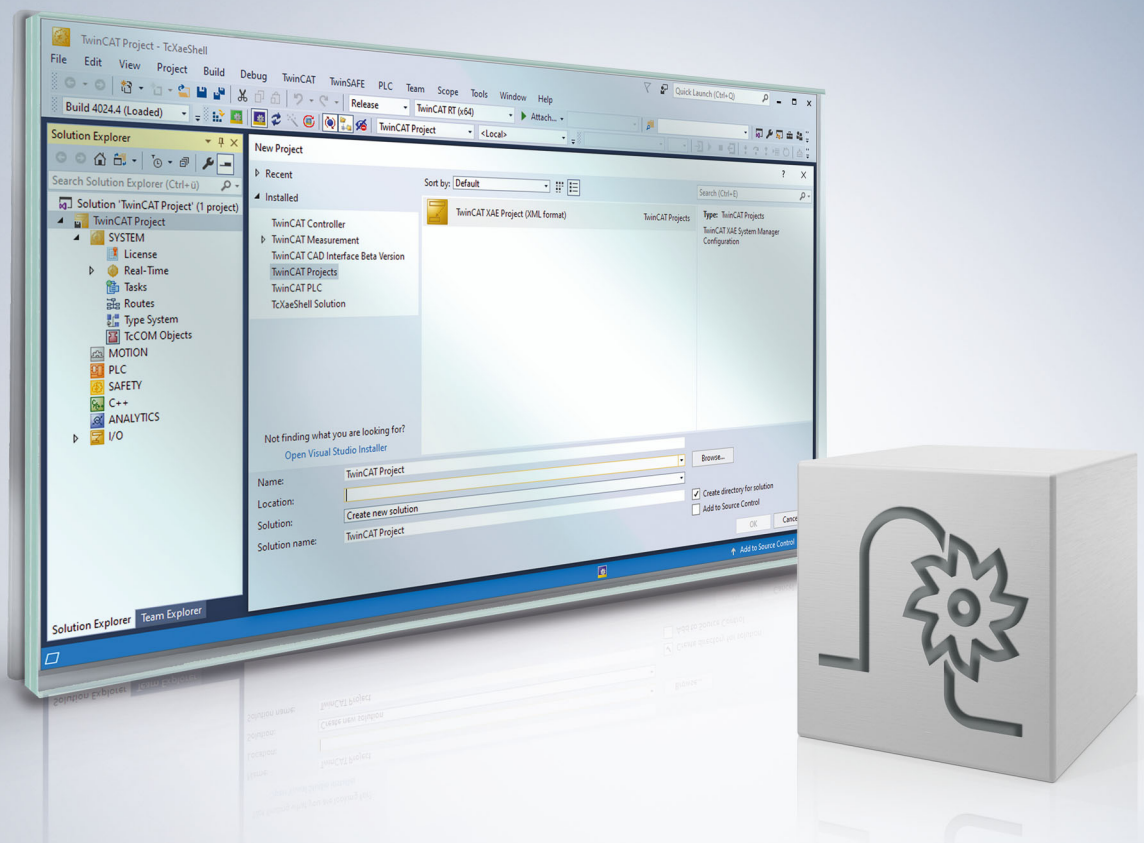
BECKHOFF New Automation Technology

Functional description | EN

TF5200 | TwinCAT 3 CNC

CNC Program encryption



12/18/2023 | Version: 1.05

Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT technology is patent protected, in particular by the following applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

NOTICE

Restriction or error

This icon describes restrictions or warns of errors.

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

General example

Example that clarifies the text.

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

1 Overview

Task

Controller/machine manufacturers supply encrypted NC programs that end-users are not allowed to modify and cannot view.

The NC kernel processes encrypted NC programs.

Properties

An encrypted NC program is recognised by its file extension. A key used for encryption and decryption must be defined for every file extension. Every file extension and the associated key define an encryption group.

A file is recognised as encrypted if its extension matches one of the encryption group file extensions. The CNC uses the associated key to decrypt the file automatically during NC program decoding.

The **program ISGCrypter**) must be used. The examples shown here are taken from the program.

Display

An NC program encrypted by the user can be decrypted by the program **ISG Crypter** if the required key is known.

Alternatively, the user can create a user-defined interface (HMI) that includes the "ISGEncrypt.dll" or "ISGEncrypt_x64.dll" library (for a 64-bit application) and integrate the "decode_cnc_file()" method there.

Parameterisation

The user can define 3 different encryption groups. The keys are transmitted by CNC objects [▶ 20] to the NC kernel at controller start-up or before program start.

The file extensions assigned to the keys are configured by P-CHAN-00283 [▶ 19].

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

Initialisation

The following steps are required to use an encrypted NC program:

1. An NC program is encrypted with an individual key and saved to a folder.
2. The file extension is entered in the channel parameter list as an encrypted file type for the corresponding channel.
3. In parallel to the file extension, the associated key used to encrypt the file is entered in the NC kernel. Make the entry by using a write operation to a CNC object. This can be executed by the PLC.

Sequence

When an NC program is invoked, the NC kernel detects from the extension whether it is encrypted. If the NC program is detected as encrypted, the kernel decrypts it using the specified key. If the file extension is defined as not encrypted, the program is processed as a normal NC program.

NOTICE

If the key is incorrect, the file is still decrypted. The NC kernel attempts to process the file and this then results in a syntax error.

CNC diagnosis

Entries of NC program parts in the CNC diagnostic data "diag_data.txt" are encrypted by a key from the controller manufacturer, i.e. they are not visible to users.

Flow chart

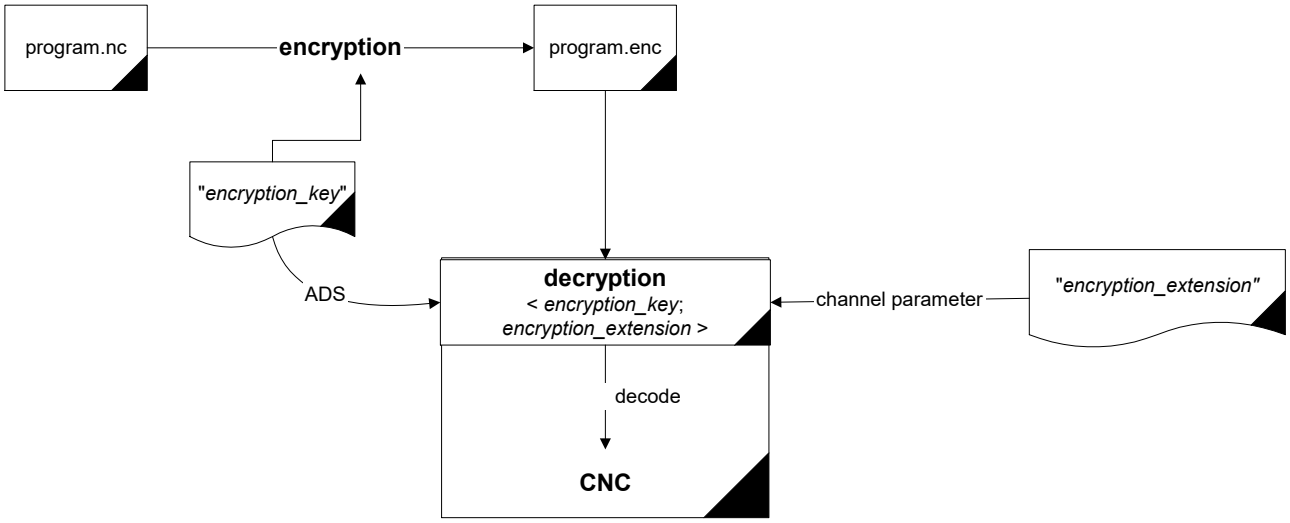


Fig. 1: Encryption/decryption flow chart of an NC program

HMI process

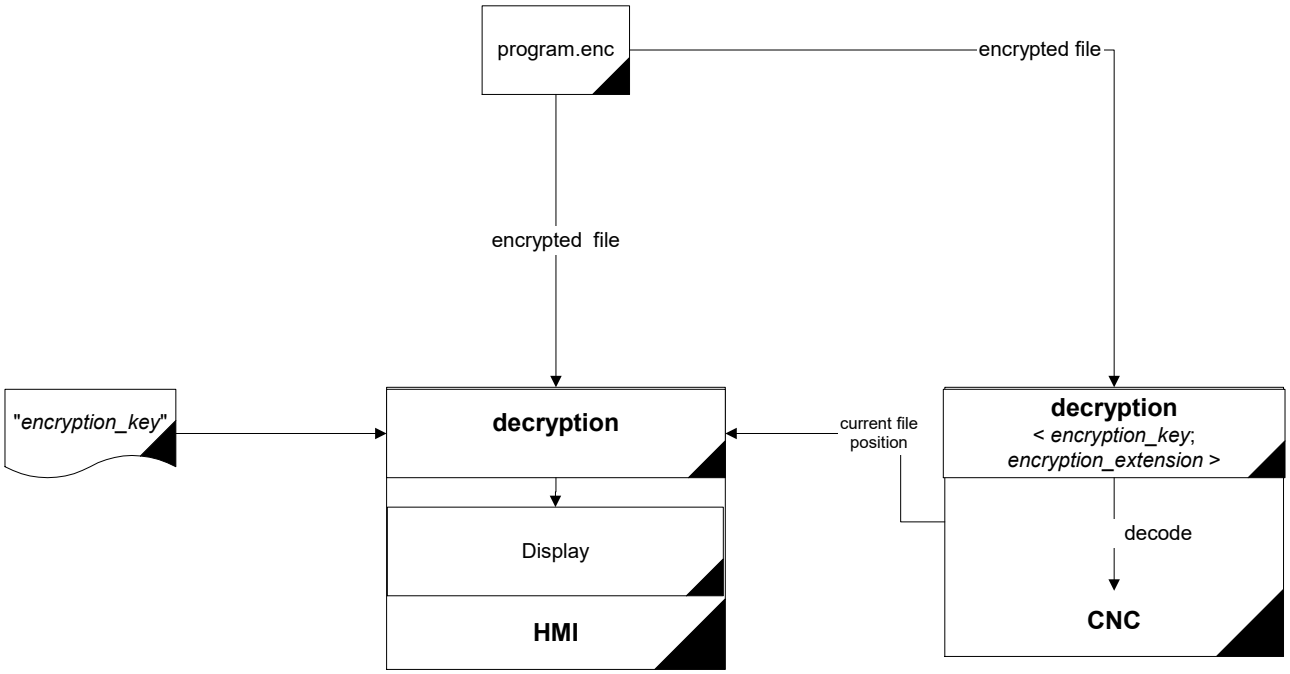


Fig. 2: Displaying an encrypted NC program

3 Encryption groups and configuration

Groups

The user can define 3 different encryption groups for the NC kernel. Each of these groups consists of a pair comprising a key and a file extension. A key can contain a maximum of 56 characters plus '\0'.

A file extension must consist of 1 to 3 characters. When the NC kernel loads an NC program, it checks whether the extension of the NC program is entered in one of these 3 groups. If this is the case, the NC kernel decrypts the NC program with the key belonging to the associated group.

Default configuration

Users can use the groups as they wish. The pair assignment of key and file extension is depicted in the table below:

Group	Key	Channel parameters
1	mc_encryption_key_0	encryption_extension[0] P-CHAN-00283
2	mc_encryption_key_1	encryption_extension[1] P-CHAN-00283
3	mc_encryption_key_2	encryption_extension[2] P-CHAN-00283

3.1 Channel parameter list

Parameterisation options for file extensions in the channel parameter list

The table below shows an example of a setting for the extensions in the channel parameter list. The file extensions for groups 1 to 3 (index 0, 1, 2) can be set.

Channel parameters	Value
encryption_extension[0]	enc
encryption_extension[1]	od
encryption_extension[2]	e

A further group also exists. This group is permanently specified by the controller or machine manufacturer and is used for the encryption of user-created NC programs (e.g. CNC cycles). These NC programs have the extension “ecy”.

You are advised not to re-assign the file extension “ecy“ for user-created definitions.

If you define the file extension “ecy”, it is not possible to use CNC cycles since NC programs encrypted by controller/machine manufacturers cannot be decrypted.

3.2 Setting keys via CNC objects

The keys for encryption groups must be set via CNC objects. Refer to the example below on how to address objects via the index group and index offset.

You can also set them online using the ISG Object Browser of the CNC.

As a security measure, all keys are only displayed hidden.

Access to encryption

All groups are accessible as shown in the example below. The arrays can be written via CNC objects.

Channel 1:

Task: COM

```
IDXGRP      :=16#00120101  ( Channel 1 )
IDXOFFS     :=16#00000094  ( mc_encryption_key_0)
IDXGRP      :=16#00120101  ( Channel 1 )
IDXOFFS     :=16#00000095  ( mc_encryption_key_1)
IDXGRP      :=16#00120101  ( Channel 1 )
IDXOFFS     :=16#00000096  ( mc_encryption_key_2)
```

ADS function block

Transfer takes place using the function block ADSWRITE(). The following applies to the example above:

```
fb_AdsWrite( NETID   :='',
             PORT    :=553,
             IDXGRP  :=16#00120101,
             IDXOFFS :=16#00000094,
             SRCADDR := ADR(mc_encryption_key_0),
             LEN     := SIZEOF (mc_encryption_key_0),
             WRITE   := TRUE
```

);

NOTICE

When you write CNC objects, note that you may have to insert a "\0" at the string end.

4 Library methods

ISGEncryption.dll

This auxiliary DLL includes the methods for encrypting and decrypting NC programs.

- encode_cnc_file()
- decoder_cnc_file()
- get_version()

The library was designed for the European/Western character set. If different character sets are used, it may result in unforeseen side effects.

4.1 Encryption

The specified input file is fully encrypted with the specified key and is saved as the output file.

long **encode_cnc_file** (char *pIn, char *pOut,

char* encryption_key, char* err_buffer)

Parameter

Name	Type	Meaning
pIn	char*	Name for input file
pOut	char*	Name for output
encryption_key	char*	Key
err_buffer	char*	Buffer for error messages: "Key is longer than 56 characters" "No key defined" "Could not open input file" "Could not open output file" If a blank string is entered, no error occurred. A maximum of 256 characters can be transmitted.

Return values

Value	Meaning
-2	Maximum key length exceeded.
-1	Key missing.
0<x	Encryption of x characters successful.

4.2 Decryption

The specified file is decrypted with the specified key and written to the buffer "pDest".

long **decode_cnc_file** (unsigned char *pDest, char *pName, long offset,
long length, char *decryption_key,
char *err_buffer)

Parameter

Name	Type	Meaning

pDest	unsigned char*	Buffer for output
pName	char*	Name for input file
offset	long	Offset for reading in the input file
length	long	Read length
decryption_key	char*	Key
err_buffer	char*	Buffer for the following error messages: "Key is longer than 56 characters" "No key defined" "File offset was negative" "File length was negative" If a blank string is entered, no error occurred. A maximum of 256 characters can be transmitted.

Return values

Value	Meaning
-4	Maximum key length exceeded.
-3	Offset parameter is negative.
-2	Length parameter is negative.
-1	Key missing.
0<x	Decryption of x characters successful.

4.3 Version number

The file version of the dll can be determined with the library method get_version(). This is the same version that is obtainable by right-clicking File => Properties => Details (see the figure below).

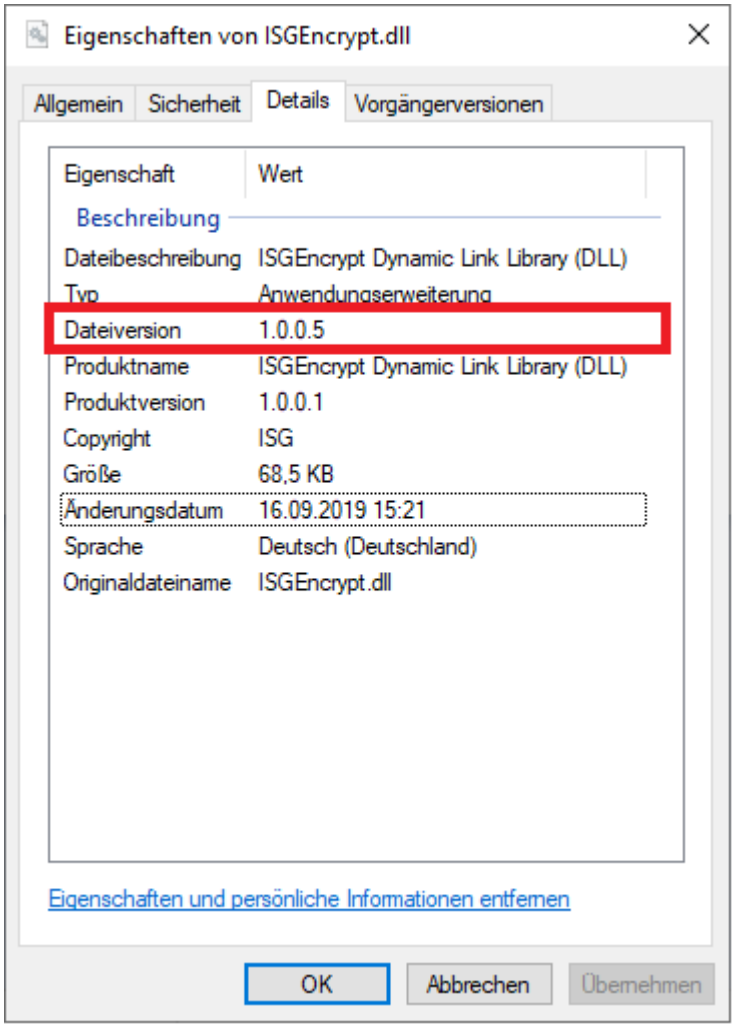


Fig. 3: Determining the dll file version

Parameter

Name	Type	Meaning
pDest	unsigned char*	Buffer for version string

Return values

Value	Meaning
TRUE	Version string successfully copied to the buffer.
FALSE	Version string could not be written to the buffer.

5 Use of the dll in a test GUI

An example of C#/.NET is provided below. This shows the use of ISGEncryption.dll by an example application in the figure below.

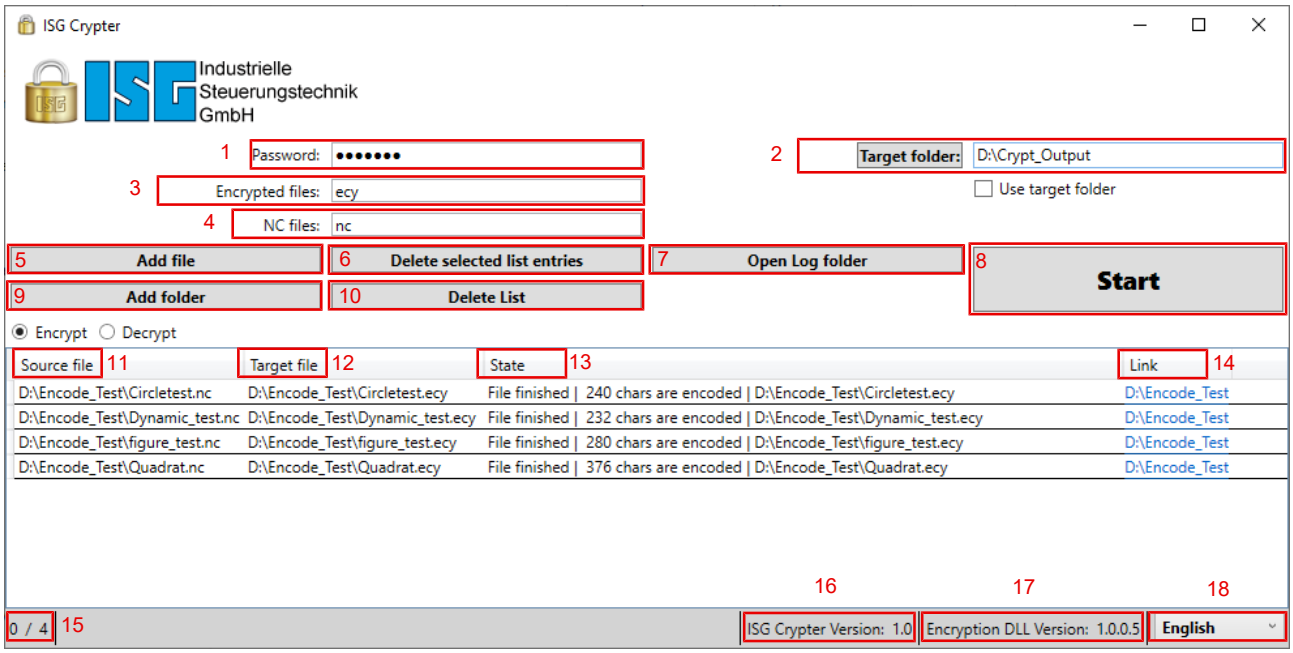


Fig. 4: Description of the GUI (= Graphical User Interface)

Index	Meaning	Description
1	Password	This is the password to encrypt/decrypt the NC program. The password may consist of 1 to 56 characters. However, we recommend min. 8 characters. The password is not stored in the encrypted program. Its validity is not verified later. In other words, an invalid password during decryption does not lead to an abort but the program cannot be correctly encrypted nor is it readable.
2	Output directory	Besides the default directory (= directory of source file), you can select a different folder. If you use a different directory than that of the source file, tick the box below. If selected, all decrypted/ encrypted files are saved here.
3	Encrypted files	File extension for encrypted files; *.ecy is the default. The file extension is also the file filter for decryption. Files with this extension are automatically detected as encrypted and decrypted at start.
4	NC files	Files with this extension are automatically detected as NC files and encrypted at start.
5	Add file	Add one or several files to the list to be processed (see window below).
6	Delete marked list entries	Marked list entries are removed from the program.
7	Open log directory	Open the directory containing the log files. The log files are generated in the selected language when they are encrypted or decrypted.
8	Start	Start of encryption or decryption. All entries in the list are processed, regardless of whether elements are marked in the list or not. First remove files from the list if they are not to be processed.

9	Add folder	Add the contents of an entire folder with the file filter (4) of the list to be processed (see window below).
10	Delete list	Removes all current entries from the encryption/decryption list.
11	Source file	Indicates what course files are encrypted/decrypted.
12	Destination file	Indicates the destination files created for encryption/decryption.
13	Status	The current status for encrypting/decrypting the file.
14	Link	Link to the output file folder.
15	Marked files	Number of marked elements / total number of elements.
16	ISG Crypter Version	Program version.
17	DLL version	Version of the dll for decryption.
18	Language	Set the required language. The possible languages are 'German', 'English', 'Russian', 'Italian', 'Spanish', 'French', 'Japanese' and 'Chinese'. The English terms are kept even when the language is changed.

5.1 Encryption example

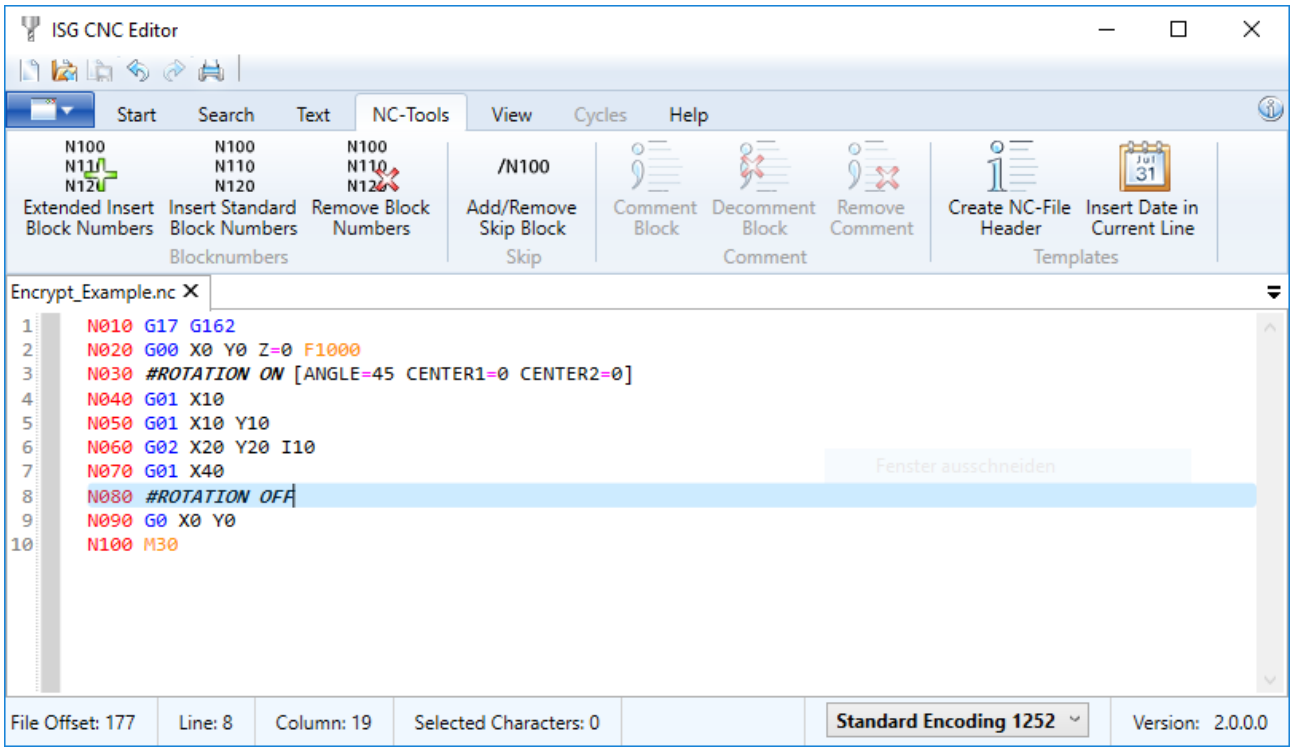


Fig. 5: Vie of the source file with readable code

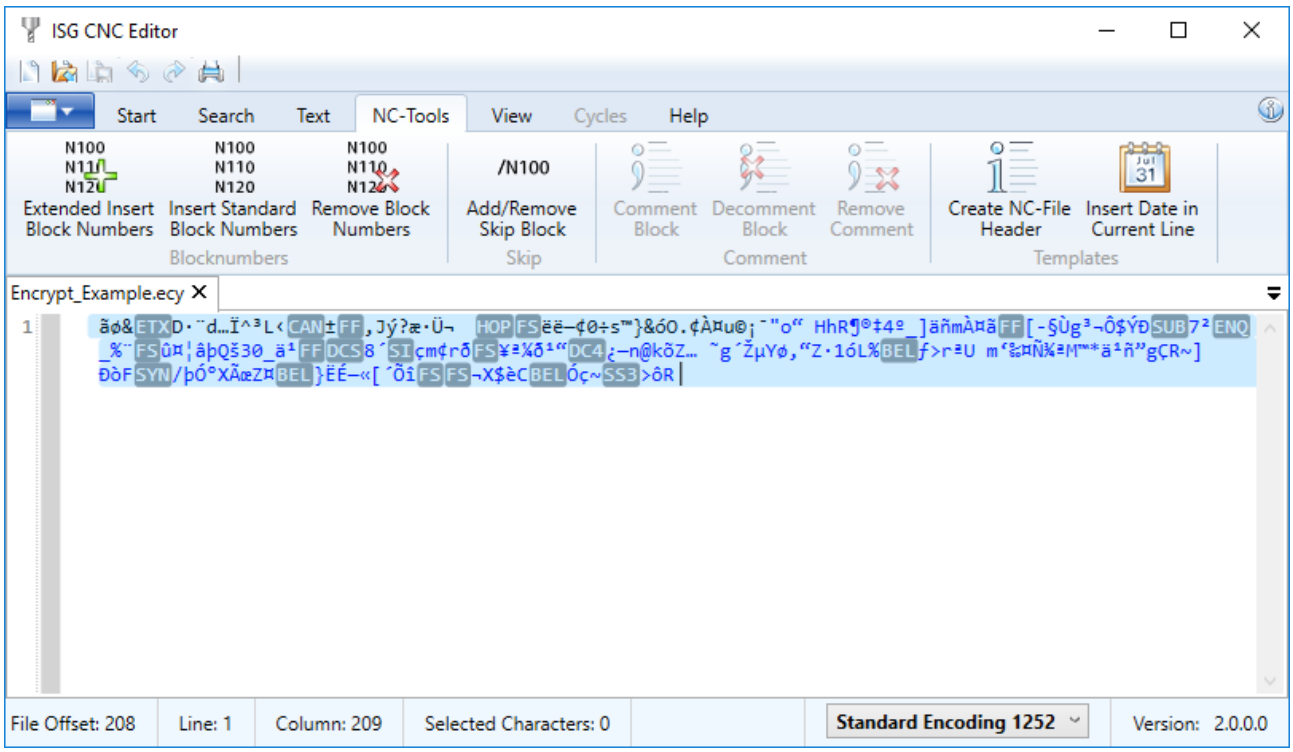


Fig. 6: Encrypted file with encrypted code

Procedure to encrypt a file

The file in the figure “View of the source file with readable code” is encrypted. ISGEncryption.dll must be in the same folder as the example application.

1. In this case, choose "asdf" as the password.
2. Select "Encrypt_Example.enc" (in the same folder as the input file) as the output file.
3. The result should be similar to the figure “Encrypted files”.
4. Select the output file with "Decrypt File".
5. Enter 0 for "Start" and 16 for "Length".
6. Click "Show" should return the same output as in the figure “Source data”.

6 Parameter

6.1 Channel parameters

P-CHAN-00283	Define file extensions to encrypt NC programs
Description	The NC channel can process encrypted NC programs. Encryption is recognised by the file extension. A maximum of 3 self-defined file extensions are available in the channel parameter 'encryption_extension[...]' to configure file extensions. A file extension can consist of one to maximum 3 characters. No distinction is made between uppercase and lowercase letters in the file extension. A check is made whether the extension is entered in one of the 3 groups before opening an NC program. If the check is positive, the NC kernel decrypts the NC program with the key belonging to the related group. Both main programs and global subroutines can be encrypted. For more information about encryption see [FCT-C12 [▶ 8]].
Parameter	encryption_extension[i] where i = 0 2
Data type	STRING
Data range	Maximum of 3 characters
Dimension	----
Default value	*encryption_extension[0]* ---- *encryption_extension[1]* ---- *encryption_extension[2]* ---- *encryption_extension[3]* ecy *

Remarks	* File extensions can be set for the groups 1 to 3 (Index 0, 1, 2). A further group also exists. This group especially is pre-defined by the controller or machine manufacturer and is used for the encryption of self-created NC programs (e.g. cycles). The extension is 'ecy'. It is recommended not to re-use this extension for new user-defined definitions
	Parameterisation example:
	`encryption_extension[0] enc (1st group)`
	`encryption_extension[1] od (2nd group)`
	`encryption_extension[2] d (3rd group)`

6.2 CNC objects

Notes on addressing

<C_{ID}> Channel or channel ID starting with 1

For further information on addressing CNC objects, see [FCT-C13//Description].

Name	**mc_encryption_key_0**		
Description	This object specifies the key for the first encryption group. The encryption group is defined by the parameter P-CHAN-00283 [▶ 19] and refers to the specified file extension. This key acts on encryption_extension[**0**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Index offset**	0x94
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

Name	**mc_encryption_key_1**		
Description	This object specifies the key for the second encryption group. The encryption group is defined by the parameter P-CHAN-00283 and refers to the specified file extension. This key acts on encryption_extension[**1**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Indexoffset**	0x95
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

Name	**mc_encryption_key_2**		
Description	This object specifies the key for the third encryption group. The encryption group is defined by the parameter P-CHAN-00283 [▶ 19] and refers to the specified file extension. This key acts on encryption_extension[**2**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Indexoffset**	0x96
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

7 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Index

P

More Information:
www.beckhoff.com/TF5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com