

# StreamTransformationFilter

From Crypto++ Wiki



A **StreamTransformationFilter** allows a Symmetric Cipher to participate in Pipelining. The filter also handles details such as padding.

## Example

In the example below, `CBC_Mode< Rijndael >::Encryption` is the `StreamTransformation` object (by way of derivation in `CBC_Mode`).

```
byte key[ Rijndael::DEFAULT_KEYLENGTH ],
    iv[ Rijndael::BLOCKSIZE ];
...
// Encryptor
CBC_Mode< Rijndael >::Encryption e( key, sizeof(key), iv );

// Encryption
CryptoPP::StringSource( PlainText, true,
    new CryptoPP::StreamTransformationFilter( e,
        new CryptoPP::StringSink( CipherText )
    ) // StreamTransformationFilter
); // StringSource
```

To specify a padding scheme, use an overloaded constructor.

```
...
// Encryption
CryptoPP::StringSource( PlainText, true,
    new CryptoPP::StreamTransformationFilter( e,
        new CryptoPP::StringSink( CipherText ),
        BlockPaddingScheme::NO_PADDING
    ) // StreamTransformationFilter
); // StringSource
```

Retrieved from "<http://www.cryptopp.com/wiki/StreamTransformationFilter>"

Categories: Sample | Filter

- This page was last modified on 7 March 2009, at 00:58.