

QUICK-START GUIDE

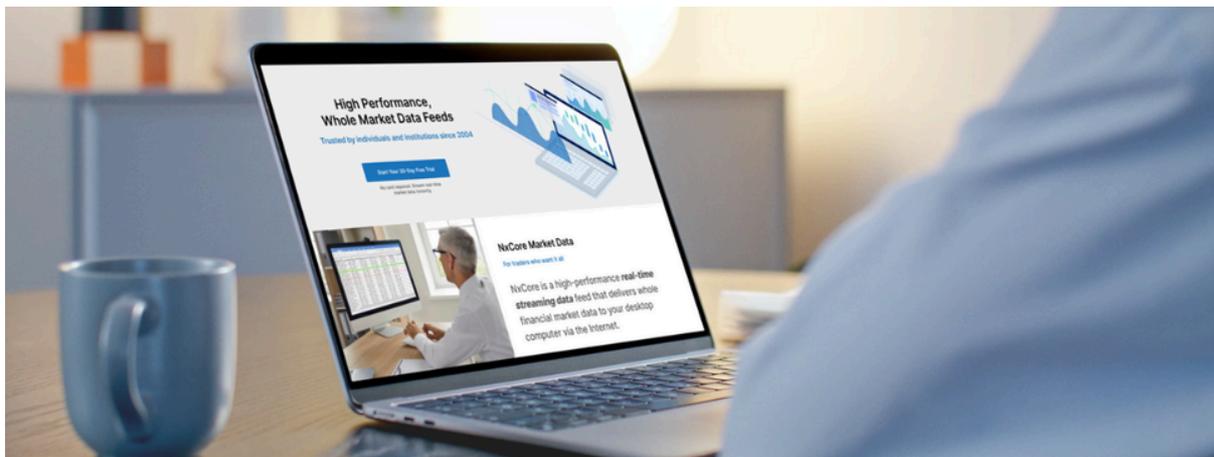
Document last updated on: 01/28/2026

Purpose

Get up and running with NxCore in minutes using its callback-driven API. This guide walks you through installation, linking the API library, running your first streaming callbacks, and verifying output — with examples for relevant languages.

Prerequisites

- NxCore API Downloaded (for C, C#, C++, Python, Java)
- Basic familiarity with your development environment
- Optional: sample tape files for testing



Step 1: Install & Link NxCore

C example:

```
#include "stdio.h"

#include "NxCoreAPI_Wrapper_C++.h"

NxCoreClass NxCore;

//define a callback for trade messages

int TradeCallback(const NxCoreSystem* sys, const NxCoreMessage* msg) {

    if (msg->MessageType == NxMSG_TRADE) {

        const NxCoreTrade& trade = msg->coreData.Trade;

        char* symbol = msg->coreHeader.pnxStringSymbol->String;

        double price = NxCore.PriceToDouble(trade.Price, trade.PriceType);

        printf("Trade received: %s @ $%.2f\n", symbol, price);

    }

    return NxCALLBACKRETURN_CONTINUE;

}

int main(int argc, char* argv[]) {

    if (NxCore.LoadNxCore("path/to/library"))

        int returnValue = NxCore.ProcessTape("path/to/sample"."tape", 0, 0, 0, TradeCallback);

    else

        printf("loading library failed\n");

    return 0;

}
```

Notes

- Include NxCoreAPI_Wrapper_C++.h and link the NxCore library according to the [Basic Concepts documentation](#)
- TradeCallback is invoked for each trade message in the tape.

- Replace "path/to/sample.tape" with the path to your tape file.

Other languages

Download and link the NxCore library / API per instructions on the [NxCore Basic Concepts](#)

Step 2: Set Up Your First Callback

NxCore streams data via callbacks — your application defines a callback function to handle messages (trades, quotes, etc.). For example:

- Use the [Introduction to Message Callbacks](#) guide to learn how callbacks work in various languages. nxcapi.com
- For a concrete “First Trade” callback in C, C++, C#, Java or Python — see the [Getting Started Trades example](#) (multi-language) or C++-focused version at [BasicTrade C++ guide](#)

Example (C-style pseudocode, simplified):

```
int OnNxCoreCallback(const NxCoreSystem* sys, const NxCoreMessage* msg) {  
  
    if (msg->MessageType == NxMSG_TRADE) {  
  
        // extract trade data from msg->coreData.Trade and msg->coreHeader  
  
        // e.g. symbol, price, size, timestamp  
  
    }  
  
    return NxCALLBACKRETURN_CONTINUE;  
  
}  
  
NxCore.ProcessTape("path/to/tape.nx2", 0, 0, 0, OnNxCoreCallback);
```

Step 3: Run Sample Tapes

- Use the built-in tape-processing function (e.g. **NxCore.ProcessTape**) to run sample or historical tape files. See the [ProcessTape documentation](#) for details on parameters, flags, historical vs real-time feed handling, and callback behavior.

- For performance tuning, filtering, and deeper control over quotes/trades/options — you may reference additional docs (e.g. “UserData & Performance” guides) after you’re comfortable with basic callbacks.

Step 4: Verify Output

- Confirm your callback is invoked and printing/logging data as expected (e.g. trades appear with symbol, price, size, timestamp)
- Ensure no errors during tape processing (library load, tape open, callback execution)
- Optionally: visualize or log results for further analysis

Next Step

Primary:

[Handle your first callback → Message Introduction & Example Guide](#)

Secondary:

[Explore full NxCore API Reference → NxCore Documentation Home](#)

Related resources:

- [NxCore Basic Concepts / Overview](#)
- [First Callback Guide \(Trades, multi-language\)](#)
- [BasicTrade C++ callback example](#)
- [Tape Playback & ProcessTape details](#)
- [UserData & Performance / advanced data handling concept](#)