xingAPI DLL 개발가이드

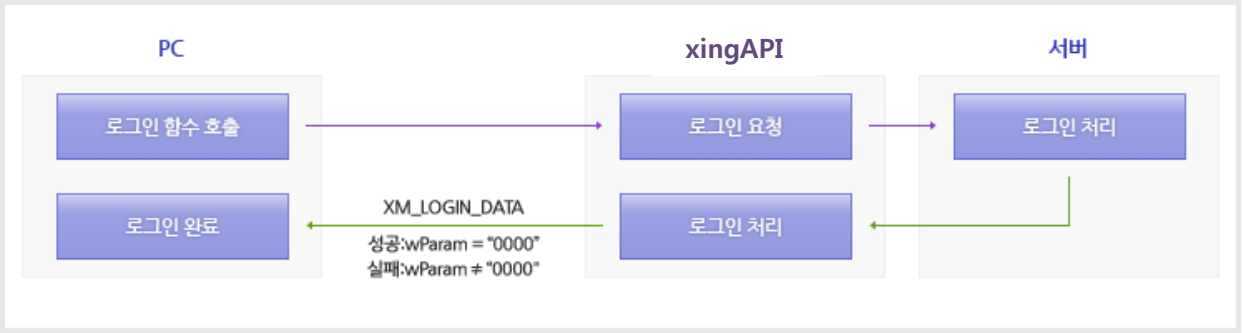
프로그램 흐름도


1. 서버연결 : 당사 서버와 연결
2. 로그인 : 당사 서버에 로그인(아이디/비밀번호, 공인인증)
3. 데이터처리 : 조회성TR/실시간TR을 이용하여 데이터 조회 및 처리
4. 로그아웃 : 당사 서버에 로그아웃
5. 서버연결종료 : 당사 서버와 연결된 세션을 종료

로그인

서버와 연결된 이후, 아이디 및 공인인증을 처리합니다.
xingAPI는 복잡한 로그인 과정을 서버로 요청하며, 서버에서 받은 결과값을 Window Message로 전송합니다.

1) 로그인 처리 Diagram

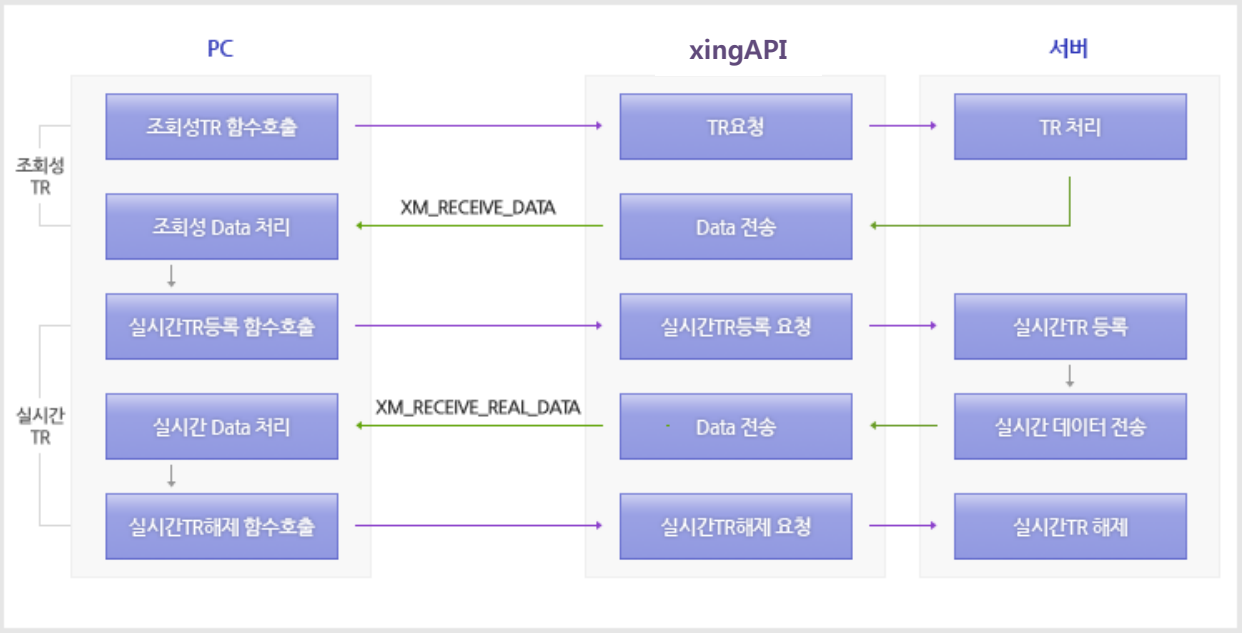


2) 로그인 과정

1. 로그인 함수(ETK_Login()) 호출
2. 로그인 인증 성공/실패는 로그인 호출 시에 등록한 윈도우로 메시지(XM_LOGIN) 전송
3. 메시지의 wParam의 값이 "0000" 이면 성공, 그 외에는 실패

일반 데이터 조회

1) 데이터 조회 Diagram



2) 데이터 조회 과정

1. 조회TR 요청(ETK_Request())
2. 조회TR에 대한 응답 : 조회TR 요청 시 등록한 윈도우로 XM_RECEIVE_DATA 메시지 전송
3. 조회 결과 데이터 처리
4. 실시간 데이터가 필요하면 실시간 등록 이후에 사용하며 사용이 완료된 이후에 실시간 해제 호출
5. 실시간 데이터 등록은 수신 받기를 원할 때 등록해도 되지만, 조회 Data를 받은 이후에 하는 것을 추천

※ TR이란?

- Transaction의 약자로 서버에 데이터를 요청하고 데이터를 받는 일련의 행동을 일컫습니다.
- TR은 조회 데이터용과 실시간 데이터용의 2가지가 존재합니다.

※ 조회TR과 실시간TR 을 이용해 서버로부터 트레이딩에 관련된 모든 데이터를 받을 수 있습니다.

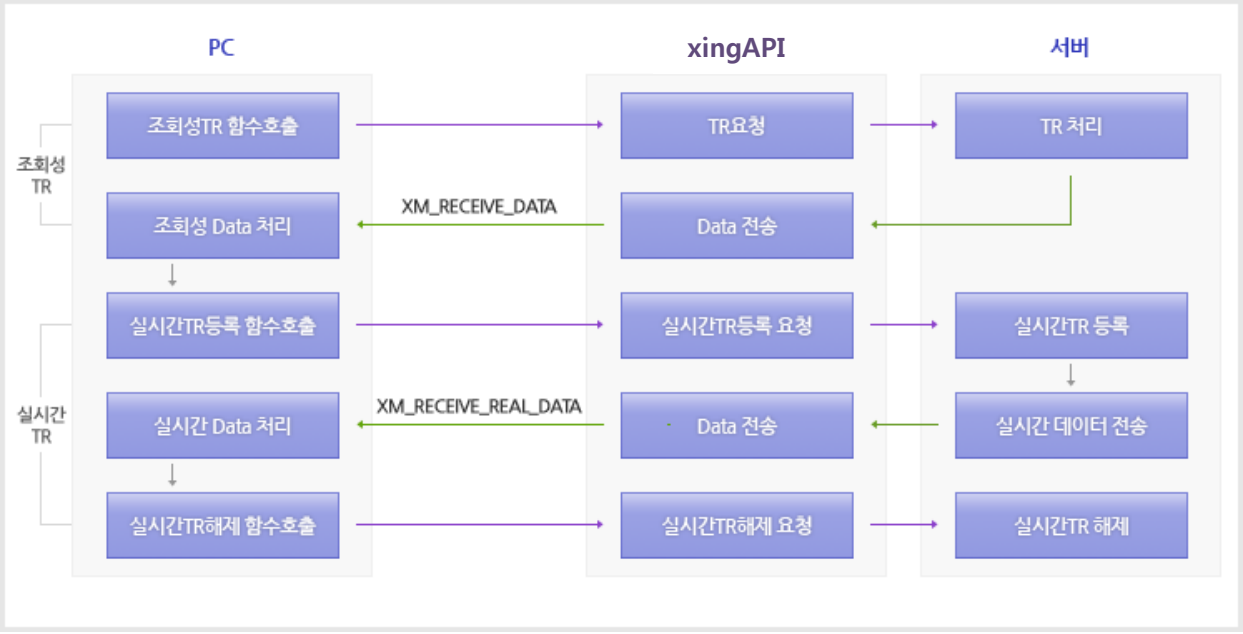
- 조회TR : 서버로 데이터 요청 당시의 데이터를 전송
- 실시간TR : 서버로 데이터 요청(ETK_Advise()) 이후에 데이터가 변경될 때마다 데이터를 전송하며 데이터 요청종료(ETK_Unadvise())를 하면 데이터 전송을 멈춤
  요청 당시의 데이터를 전송해주지 않으므로 시세 등 실시간으로 변하는 데이터를 받기 위해서는 조회TR을 요청하여 당시의 데이터를 받은 이후에 실시간TR을 요청하여 실시간 데이터를 받아야만 함

단일 데이터(조회 TR) 처리

※ 조회TR

- 서버에 데이터를 요청했을 때의 현재 데이터를 전송합니다.
- 요청하여 데이터를 전송한 이후에 데이터가 변경이 되었을 경우 재전송이 이루어지지 않습니다.
- 여러 TR을 하나의 윈도우에서 사용하는 경우 데이터를 요청했을 때 획득하는 Request ID로 구별 합니다.
- Data를 받은 이후에는 상황별로 데이터를 해제해 주어야 합니다.
- Data는 Data -> Message -> Release 순으로 오게 됩니다.
- System Error는 Data 처리 이전에 System 문제로 인하여 Data 처리가 불가능할 경우에 발생하며 이 경우엔 Data/Message/Release 가 전송되지 않습니다.

1) 조회성TR 처리 Diagram



1. 조회성 TR 전송 : Request ID 획득
2. 조회성 TR 수신 : 조회성TR 전송 시 등록한 윈도우로 XM_RECEIVE_DATA 메시지 전송

2) XM_RECEIVE_DATA 메시지 정보

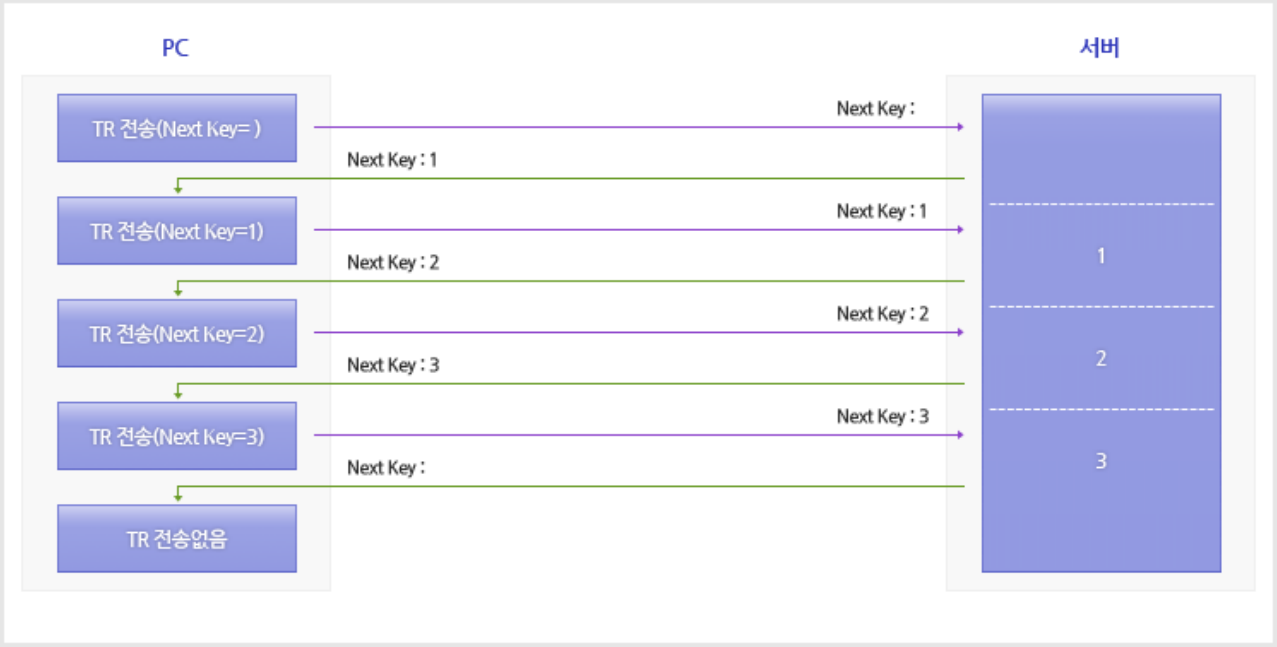
WPARAM	내용	설명
1	Data	- 요청한 Data를 전송 - Data가 없거나 에러가 발생한 경우 수신되지 않을 수 있음
2	Message	- Data 처리에 대한 Message - Data 처리에 성공을 해도 실패를 해도 Message는 발생 - Message Data를 처리하고 더 이상 사용하지 않을 경우 ETK_ReleaseMessageData()를 호출하여 Message를 해제하여야 함
3	System Error	- Data 처리 이전에 System 문제로 인한 Error가 발생할 경우에 전송 - Message Data를 처리하고 더 이상 사용하지 않을 경우 ETK_ReleaseMessageData()를 호출하여 Message를 해제하여야 함 - System Error이 발생할 경우 Release가 전송되지 않으며 ETK_ReleaseMessageData()에서 자동으로 Request ID를 해제해 줌
4	Release	- Data 처리가 완료된 경우에 전송 - Request ID를 해제하기 위해 ETK_ReleaseRequestData()를 호출하여야 함

3) Request ID란?

1. 서버에 데이터를 전송하게 되면 부여받게 되는 ID이며, 서버로부터 데이터를 받을 때 ID 값을 같이 받게 되므로 전송한 값에 대한 수신값을 찾을 수 있습니다.
2. Request ID는 xingAPI에서 관리하므로 수신이 완료가 되면 해제해 주어야 합니다.
3. Request ID는 0~255사이의 값을 사용하며, 해제하지 않으면 255개의 전송이 이루어진 이후에는 전송이 불가능합니다.

연속조회

연속조회는 많은 양의 Occurs 데이터를 한번에 전송(수신)하지 않고 나누어서 전송(수신)하는 것을 일컫습니다.
주식시세 과거 데이터등은 한번에 보내기에는 방대한 양이므로 연속조회를 이용합니다.
체결/잔고 등의 Database를 이용하는 경우에도 한번에 많은 양의 Query가 이뤄지면 DB가 Lock이 걸리므로 연속조회를 이용하여 데이터를 보냅니다.



1) 연속조회 기본개념

- 초기 전송시 Next Key는 SPACE로 전송
- 서버에서 다음 데이터가 있으면 다음 데이터의 시작위치를 Next Key로 전송
- 서버로부터 Next Key를 받으면 다음 TR전송시 받은 Next Key를 같이 전송
- 서버로부터 Next Key를 받지 못하였다면 더 이상 연속데이터는 없음

2) 연속키

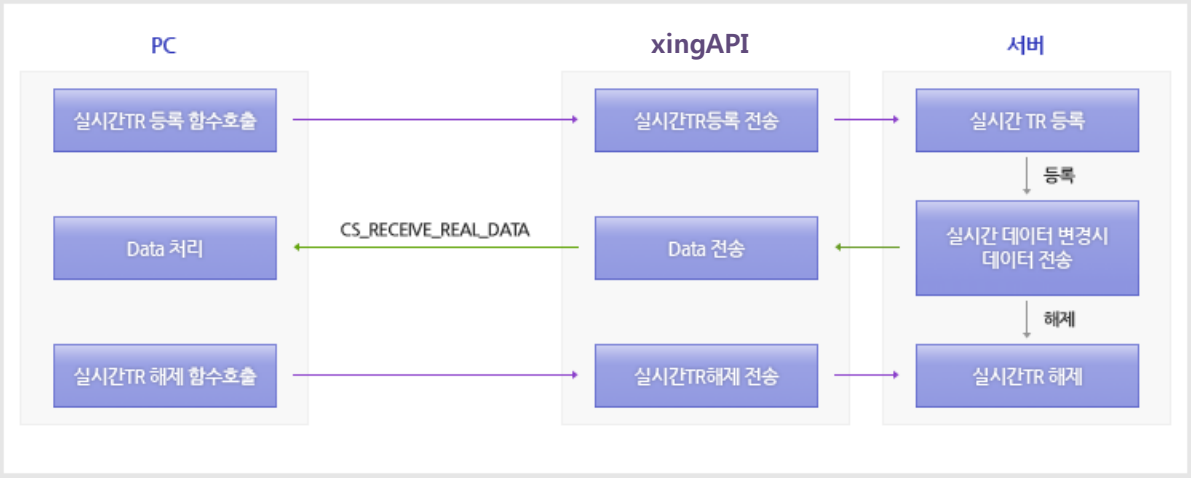
- TR코드가 5자리인 경우에는 연속키는 OutBlock의 값을 InBlock에 넣어주고
- TR코드가 10자리인 경우에는 연속키는 Request 시에 넣어주게 됩니다.

실시간 데이터 조회 (실시간TR 처리)

※ 실시간TR

- 서버에 데이터를 요청한 이후에 데이터가 변경이 되었을 경우 전송이 이루어집니다.
- 데이터를 더 이상 받고 싶지 않을 경우에는 요청취소를 해야 합니다.
- 요청한 시점의 데이터는 전송하지 않으므로 조회TR을 이용하여 현재의 데이터를 가져와야 합니다.

1) 실시간TR 처리 Diagram



1. 실시간TR을 등록합니다.
2. 실시간 데이터가 변경되면 해당 데이터를 등록했을 때 설정한 윈도우로 XM_RECEIVE_REAL_DATA 메시지를 통해서 전송합니다.
3. 더 이상 필요가 없다면 실시간TR을 해제합니다.

2) 실시간 TR을 등록하는 시점은?

1. 실시간TR을 등록할 당시의 데이터는 전송하지 않으므로, 조회TR을 전송하여 데이터를 받아야 합니다.
2. 조회TR을 서버로 전송함과 동시에 실시간TR을 등록하면 조회TR을 수신 전에 실시간TR을 받을 수 있고
3. 조회TR에 대한 수신을 받고 나서 실시간TR을 등록하게 되면 수신후의 등록시점까지 변경되는 데이터는 받을 수 없다는 문제가 있습니다.
4. 이 두 가지 중에 어떤 것을 택하느냐는 개발자의 선택입니다.
5. HTS 에서는 대부분 수신을 받은 이후에 실시간TR을 등록하지만 상황에 따라서는 조회TR을 전송함과 동시에 등록하는 경우도 있습니다.

TR 속성

xingAPI 에서는 TR 을 다루기 위한 몇 가지 속성이 있습니다.
이 속성을 숙지하셔야 xingAPI DLL 개발이 가능합니다.

1) Data Type
TR의 필드를 채우고 읽어오기 위한 Data의 형식에 대해 설명합니다.

2) Attribute
TR의 필드에 대한 속성 Byte 를 설명합니다.

2) Block Mode
TR은 여러 개의 Block 으로 이루어져 있으며 이 Block을 어떻게 처리하는지에 대해 설명합니다.

3) Occurs
OutBlock 중에는 일련의 데이터 배열로 이루어지는 경우가 있습니다. 이 처리 방법에 대해 설명합니다.

4) 연속조회
OutBlock 의 Occurs 데이터가 너무 많아 연속조회를 해야 하는 경우에 대해 설명합니다.

Data Type

TR의 필드에는 Data Type 이 존재합니다.
Data Type 에 따라 전송/수신 시 처리 방식이 다릅니다.

	String	Int	Float
성격	문자열	정수	실수
길이	문자열의 최대 길이 예) 12 : 최대 12자리의 문자열	정수의 최대 길이 예) 5 : 최대 5자리의 정수	정수부와 소수부로 나뉨 정수부는 소수부 포함 최대 길이 소수부는 소수부 최대 길이 예) 6.2 소수부분은 2자리 정수 부분이 3자리 혹은 4자리인 실수 정수부분에 "." 가 생략되는 경우에는 정수부가 4자리 예) 123.45 => 123.45 123456 => 1234.56
서버 전송/수신	왼쪽 정렬 남는 영역은 ' ' 로 채움 [예] 길이:5 데이터:"ABC" 	오른쪽 정렬 남는 영역은 '0' 으로 채움 [예] 길이:5 데이터:"123"  * 일부 TR의 경우는 필드 값의 변경으로 인하여 뒷 부분이 null로 채워집니다. Ex) t1305의 fpvolume	오른쪽 정렬 남는 영역은 '0' 으로 채움 [예] **<소수점을 포함하지 않는 경우>** 길이:6.2 데이터:12.45  **<소수점을 포함하는 경우 >** 길이:6.2 데이터:12.45 

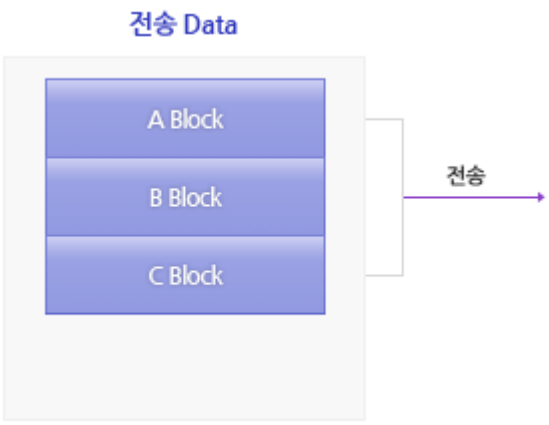
Attribute

해당 필드의 속성을 결정합니다.
Attribute 속성이 있는 TR 이면 필드 뒤에 항시 1바이트의 Attribute를 가집니다.
ex) 7의 길이를 가진 현재가 필드는 7바이트의 현재가와 1바이트의 Attribute로 전송/수신
1 Byte 의 Attribute는 각 비트마다 의미가 있습니다.

	제목	설명	구분값	비고
0번째 비트	Font Resize	화면 확대 시 글꼴크기 변경 여부		사용하지 않음
1번째 비트	Reverse	글자색과 배경색이 바뀌는지 여부	0:바뀌지 않음 1:바뀜	현재가가 상하한가일 경우에 사용됨
2번째 비트	Protect	입력불가 여부	0:입력가능 1:입력불가	사용하지 않음
3번째 비트	Cursor			사용하지 않음
4번째 비트	Color	글자색	1:Black 2:Blue(하락) 3:Red(상승) 4:Green(보합) 5:Brown 6:Cyan 7:LightRed	현재가 등에서 글자색을 표시하기 위해 사용
5번째 비트				
6번째 비트				
7번째 비트	Use	Attribute 사용여부	0:사용하지 않음 1:사용	0으로 세팅될 경우 Attribute는 무시

Block Mode

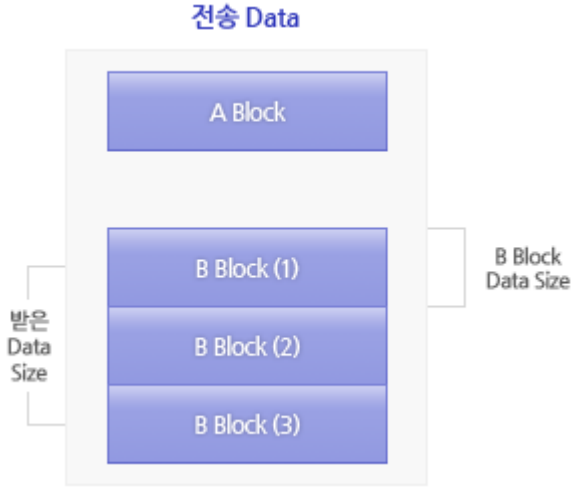
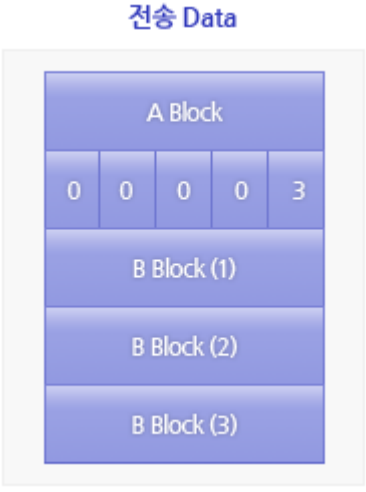
TR의 Block들을 전송/수신시 한번에 보내고/받느냐 나누어서 보내고/받느냐의 방식입니다.
Header Type 이 A 이면 Block Mode 그렇지 않으면 Non Block Mode 입니다.

Block	Non-Block
- Block별로 따로 전송 	- Block을 한꺼번에 전송 
- Block Mode는 Block을 각각 전송합니다. - 따라서 수신측에서는 A Block 을 받아서 처리하고 B Block을 받아서 처리하고 C Block을 받아서 처리하게 됩니다. - Receive 정보에는 Block 명이 있어 이것으로 어느 Block이 수신되었는지 구별할 수 있습니다.	- Non Block Mode는 Block을 한번에 전송합니다. - 따라서 수신측에서는 A Block/B Block/C Block을 한번에 받아서 처리합니다. - Receive 정보에는 Block명이 없으며 B Block은 A Block의 크기만큼을 계산해서 사용해야 합니다

많은 양의 데이터가 한 번에 수신이 될 경우, Block/Non-Block에 상관없이 잘게 쪼개서 수신을 하지만 xingAPI 에서 모아 하나의 데이터를 만든 후에 보내줍니다.
전송도 마찬가지로 많은 양의 데이터를 한번에 전송할 경우 잘게 쪼개서 전송을 합니다.
그러므로 Block 이 쪼개져서 들어오는 일은 없습니다.

Occurs

Occurs는 배열 형식의 데이터를 일컫습니다.
예를 들면 Grid 안에 들어가는 데이터(체결내역/잔고내역등)가 배열 형식의 데이터입니다.
Occurs는 DataMode에 따라 처리방식이 다릅니다.

Block	Non-Block
- Recevice 된 Block의 전체 길이와 Block의 1개의 길이를 나누어 Block의 총 개수를 구함 	- Recevice 된 Block의 첫 부분에 Block Occurs의 개수를 나타내는 5바이트의 문자열을 포함 
- 위의 경우에 수신측에서는 A Block과 B Block을 따로 받으며 B Block은 3개의 Occurs 데이터를 한번에 받게 됩니다. - 이때 수신측에서는 Block안의 Occurs의 개수를 알아야 처리가 가능합니다. - B Block 전체 데이터는 B Block만 존재하므로 받은 데이터 크기를 B Block 하나의 크기로 나누면 됩니다. - B Block 개수 = B Block 전체의 크기 / B Block 한 개의 크기	- 위의 경우에 수신측에서는 A Block 과 B Block 3개의 Occurs 데이터를 한번에 받게 됩니다. - 이때 수신측에서는 Block안의 Occurs의 개수를 알아야 처리가 가능합니다. - Block Mode와는 다르게 A Block이 같이 수신되므로 B Block의 개수를 알수 없습니다.(위의 경우엔 전체 수신 크기에서 A Block의 크기를 빼면 B Block의 크기를 알 수 있지만 C Block도 전송된다고 하고 C Block도 Occurs라고 가정했을 경우엔 Occurs의 갯수를 알 방법은 없습니다.) - 그러므로 Block이 Occurs 인 경우에는 Block이 시작되기 전에 5 바이트의 Block의 Occurs갯수를 의미하는 문자열을 포함합니다

감사합니다