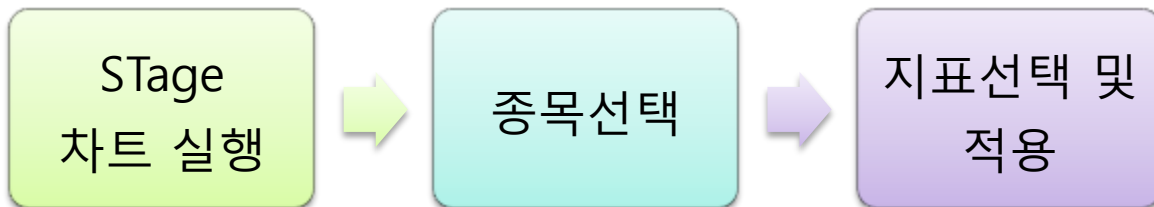


## 사용자 함수 옵션 설정안내

이번 STage 업데이트를 통해 사용자 함수의 옵션이 추가되었습니다. 지표 내의 함수 연산에 관련한 속성으로 Simple / Series / Auto 의 속성값을 갖게 됩니다. 자세한 설명은 아래의 내용을 참고하십시오.

### 1. STage 지표와 함수 작동원리

STage 차트 및 지표를 사용하는 과정은 다음과 같습니다.



이렇게 지표가 적용되면 실시간 시세가 수신됨에 따라 지표를 실행하며 이에 맞춰 매매신호를 발생시킵니다. 이 때 스테이지 지표는 지표가 적용된 스테이지 차트의 종목의 새로운 틱 시세가 갱신될 때마다 실행됩니다. 새로운 시세가 도착할 때마다 스테이지는 해당 지표의 연산을 실행합니다.

지표를 실행하는 과정에서 사용자함수 및 시스템함수를 만나게 되면 해당함수를 호출하여 해당 함수의 연산을 실행하게 됩니다.

스테이지 지표와 함수의 작동원리는 개략적으로 이와 같습니다.

## 2. 지표 내 조건문과 함수호출

다음과 같은 경우를 살펴보겠습니다. 다음 지표는 IF 문, IF~Else 문, IF~ElseIF~Else 문을 포함하고 있습니다. 이와 같이 조건문이 사용된 지표의 경우 조건문의 참, 거짓에 따라 특정 블록문이 실행되지 않게 됩니다.

```
value1 = close

If value1 > 100 Then

    value2 = FuncA( C, 5 )                [1]

ElseIf value1 < 50 Then

    value2 = FuncB( C, 5 )                [2]

Else

    value2 = 0                            [3]

End If
```

위의 예제에서  $value1 > 100$  을 만족할 경우 [1] 구간만을 실행하며  $value1 < 50$  을 만족할 경우 [2] 구간만을 실행합니다. 이외의 경우에는 [3] 구간만 실행하게 됩니다. [1] 구간이 실행될 때에는 FuncA 함수의 호출이 실행되며 [2] 구간이 실행될 때에는 FuncB 의 호출이 실행됩니다. 반면 [3] 구간이 실행되는 경우 FuncA, FuncB 두 함수의 모두 호출이 실행되지 않게 됩니다.

통상적으로 이러한 것은 큰 문제가 되지 않습니다. 하지만 어떤 함수들은 목적에 맞는 기능을 수행하기 위해 매 봉마다 실행이 되어 합니다.

다음에서 좀 더 자세히 살펴보겠습니다.

**사용자함수 : \_FuncA**

```
var:v1(0)
v1 = v1 + 1
_FuncA = v1
```

**지표**

```
If barindex % 3 = 0 Then
    value1 = _FuncA
End If
```

위의 사용자함수 \_FuncA 는 실행될 때 마다 v1 을 1 씩 증가시키며 그 값을 리턴합니다. 지표의 barindex % 3 은 barindex 를 3 으로 나눈 나머지 값을 구하는 것으로 If barindex % 3 = 0 Then 은 barindex 가 3 의 배수일 때 IF 문을 실행하게 됩니다.

Barindex	0	1	2	3	4	5	6	7
IF 문 실행	O	X	X	O	X	X	O	X
_FuncA 호출	O	X	X	O	X	X	O	X
_FuncA 내부 v1 의 값	1	1	1	2	2	2	3	3
_FuncA 리턴값	1			2			3	

스태이지 차트의 첫봉의 barindex 는 0 의 값을 가지며 이때 IF 문이 실행되며 \_FuncA 함수호출이 실행되어 값을 리턴하게 됩니다. 반면 barindex 1, 2 의 경우에는 IF 문이 실행되지 않으며 \_FuncA 의 연산도 이루어지지 않게 됩니다.

경우에 따라서는 함수 실행이 매 봉마다 이루어져야 할 수도 있습니다. 다음은 스테이지 시스템함수 MAFC 입니다. MAFC 는 Moving Average 를 MA 함수보다 빠른 속도로 계산하는 함수입니다. 이 함수는 로직 상 매 봉마다 실행되어야 정상적인 결과를 리턴을 갖습니다.

Parameter: Price(Numeric), Length(Numeric)

Variables: Summation(0), Counter(0), LoopCount(1), Initialized(False)

If !Initialized Then

    Summation = Price[0]

    LoopCount = 0

    For counter = 1 To Length - 1

        Summation = Summation + Price[counter]

        LoopCount += 1

    End For

    If LoopCount = Length Then

        Initialized = True

    End If

Else

    Summation = Summation + Price - Price[Length]

End If

MAFC = 0

If Length > 0 Then

    MAFC = Summation / Length

End If

### 3. Simple / Series / Auto 모드

이번 업데이트를 통해 스테이지에서는 이러한 현상을 효과적으로 다룰 수 있도록 함수의 속성을 다음과 같은 세 가지로 분류하였습니다.

#### (1) Simple

Simple 모드는 기존의 방식으로 함수호출을 실행합니다. 지표에서 If 문 등의 조건문 분기로 의해 함수호출이 이루어지지 않을 경우 함수 실행을 하지 않습니다.

#### (2) Series

Series 모드는 지표 내의 분기와 관계없이 매 봉마다 함수 자체적으로 연산을 수행합니다.

#### (3) Auto

Auto 모드는 스테이지에서 함수의 특성을 분석하여 Simple 모드, Series 모드 실행을 자동으로 결정합니다. 스테이지에서 Auto 모드의 함수를 Series 로 판단하는 경우는 Auto 모드의 함수 내에 함수 자기 자신에 대한 과거참조가 있을 경우 입니다. 아래와 같은 Auto 속성 함수는 Series 로 판단됩니다.

#### 사용자함수 : \_FuncTest

```
Input: Price(Numeric)
_FuncTest = ( Price + _FuncTest[1] ) / 2
```

Simple 모드와 Series 모드가 적용된 함수를 예를 통해 살펴보겠습니다.

#### 사용자함수 : \_FuncA - Simple 모드

```
var:v1(0)
v1 = v1 + 1
_FuncA = v1
```

#### 사용자함수 : \_FuncB - Series 모드

```
var:v1(0)
v1 = v1 + 1
_FuncB = v1
```

**지표**

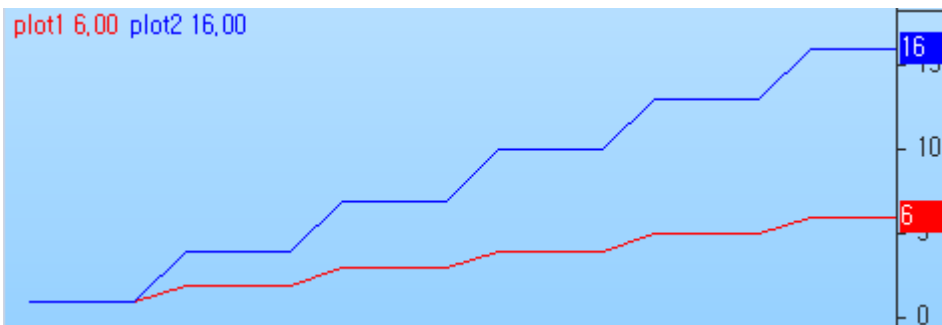
```

If barindex % 3 = 0 Then
    value1 = _FuncA
    value2 = _FuncB
End If

Plot1(_FuncA)
Plot2(_FuncB)
    
```

Barindex	0	1	2	3	4	5	6	7
IF 문 실행	O	X	X	O	X	X	O	X
_FuncA 내부 v1 의 값	1	1	1	2	2	2	3	3
_FuncB 내부 v1 의 값	1	2	3	4	5	6	7	8
_FuncA 리턴값	1			2			3	
_FuncB 리턴값	1			4			7	

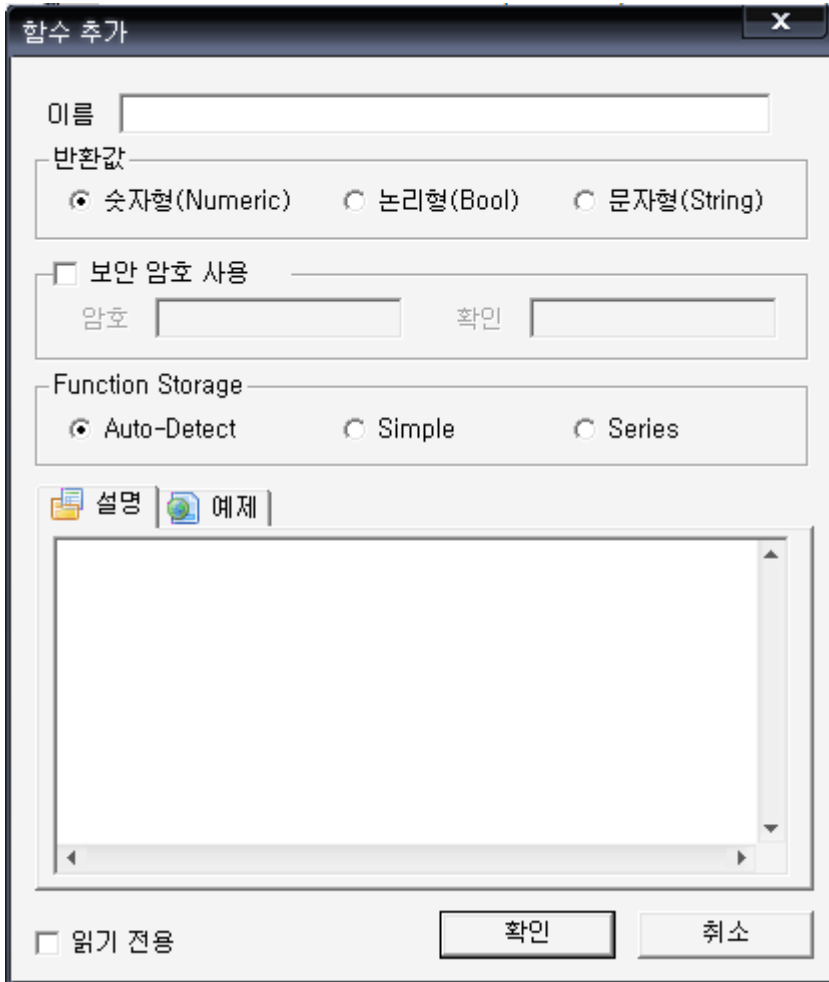
\_FuncA 는 Simple 모드로 If 문이 만족되지 않는 봉에서 함수호출 및 연산이 실행되지 않습니다. 반면 \_FuncB 는 Series 모드로 If 문과 별개로 매 봉에서 연산이 실행되 내부의 v1 값이 매 봉마다 업데이트가 됩니다. 그리하여 Barindex=3 에서 \_FuncA 의 리턴값은 2 \_FuncB 의 리턴값은 4 를 갖게되며, Barindex=6 에서 \_FuncA 의 리턴값은 3 \_FuncB 의 리턴값은 7 을 갖게됩니다.



Plot1 은 \_FuncA 의 모습이며 Plot2 는 \_FuncB 의 모습입니다. \_FuncB 의 경우 리턴값이 1 부터 3 씩 증가하여 1, 4, 7, 10, 13, 16 임을 확인할 수 있습니다. 이는 \_FuncB 는 Series 모드로 매 봉마다 연산이 실행되기 때문입니다.

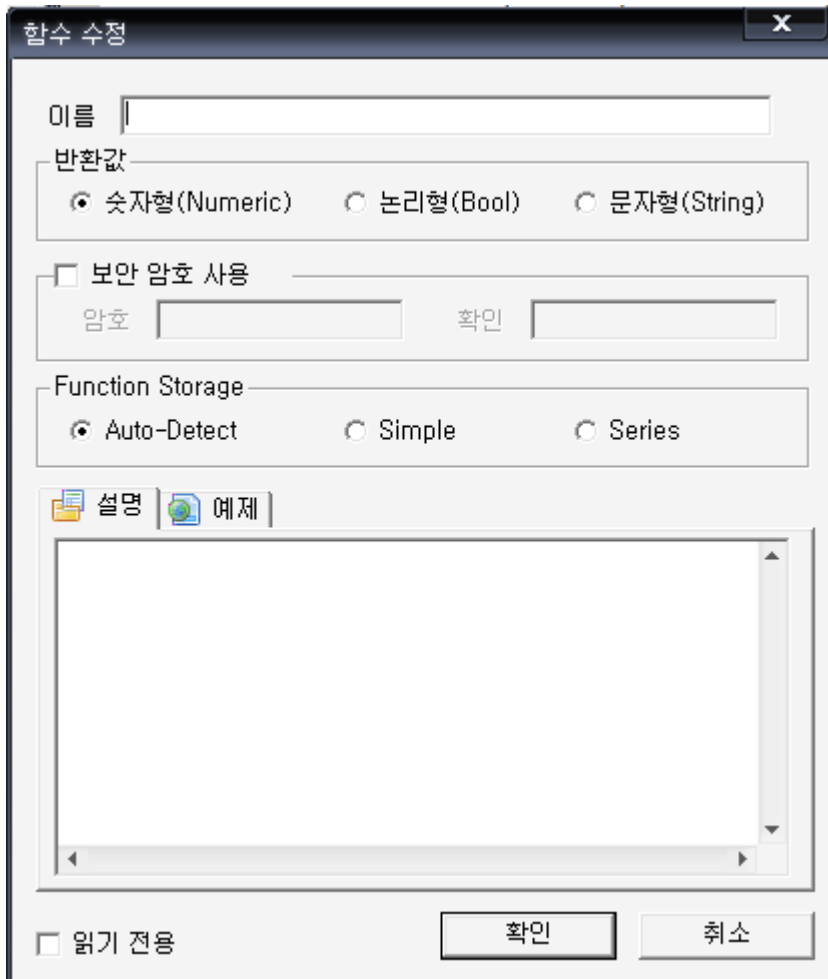
#### 4. 함수의 Simple / Series / Auto 모드 설정 방법

이번 업데이트를 통해 전략편집기에서 [새 함수]를 실행하여 함수를 추가하는 화면이 아래와 같이 변경되었습니다.



중앙의 Function Storage 부분의 Auto-Detect, Simple, Series 옵션이 해당 부분입니다. 함수를 새로 생성할 때 이 부분을 통해 옵션을 선택할 수 있습니다.

또한 지표명을 마우스 우클릭하여 나타나는 함수 수정 화면에서도 Function Storage 옵션을 선택할 수 있습니다.



함수 수정

이름

반환값

숫자형(Numeric)  논리형(Bool)  문자형(String)

보안 암호 사용

암호  확인

Function Storage

Auto-Detect  Simple  Series

설명 | 예제

읽기 전용

확인 취소