

[VPL 강좌 08] List 액티버티 구현하기 (기본과정)

작성자: Microsoft Robotics Group 김영준 수석 (yjoonkim@microsoft.com)

VPL에서는 배열 형태의 구조를 구현하기 위해 List 액티버티를 사용할 수 있습니다. List 액티버티는 1차원 적인 형태의 배열을 지원하며, 2차원 이상의 배열 구조는 C# 프로그래밍을 통해 2차원 이상의 구조를 가지는 서비스를 개발함으로써 구현이 가능합니다.

이번 강좌에서는 List 액티버티와 List Functions 액티버티를 활용하여 배열과 같이 여러 개의 데이터를 처리하는 방식에 대해 설명을 하며, 또한 List Functions가 가지는 다양한 메소드에 대해 설명을 합니다.

먼저, List 액티버티를 활용하는 방법에 대해 설명을 합니다. 일반적으로 List 액티버티는 맨 처음 List 액티버티를 초기화 하기 위해 사용되며, 그 다음 부터는 List 형태의 변수에 저장되어, 변수를 가지고 나머지 부분에서 사용이 됩니다. 그리고 이러한 변수에는 List Functions 액티버티를 통해서 값을 추가하거나 삭제 또는 다양한 연산을 적용할 수 있습니다.

아래 예는 string 타입의 List 액티버티를 생성하여 변수에 저장시키는 방법을 보여줍니다. 먼저, List 액티버티를 추가하고 타입을 string으로 지정합니다.

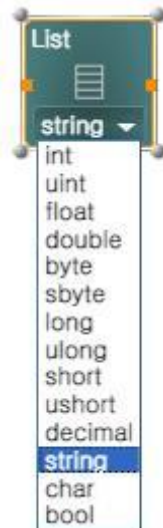


그림 1

그 다음으로는 우측에 Variable 액티버티를 추가하고 아래와 같이 List of string 타입으로 지정합니다.

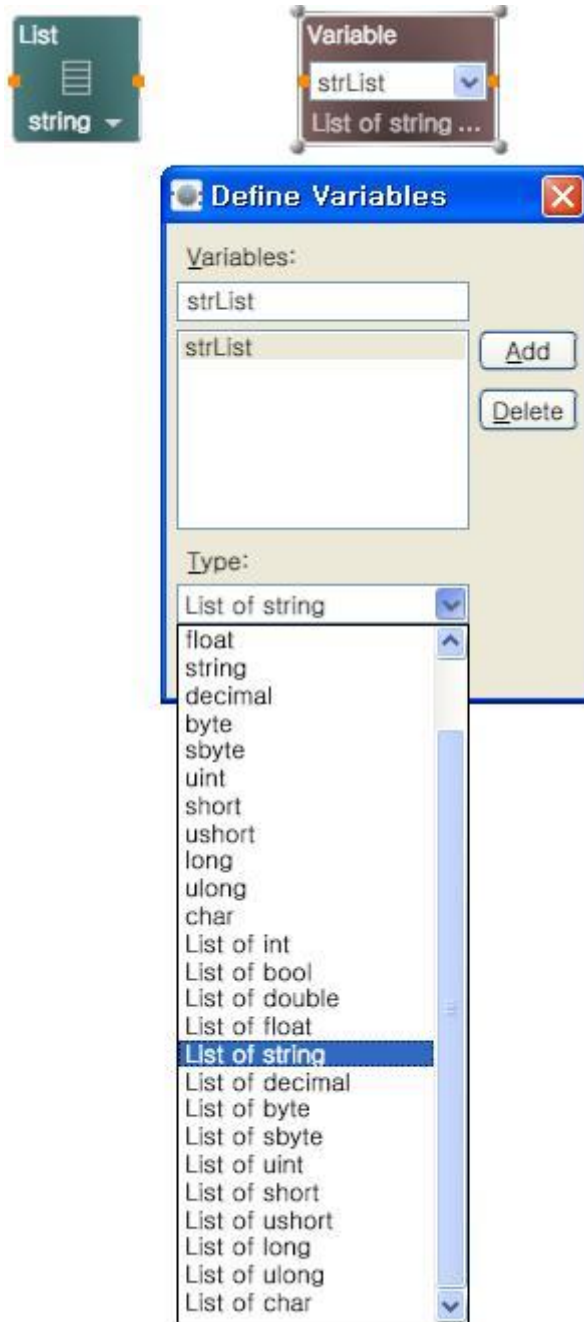


그림 2

자 이제 두 개의 액티비티들을 연결하면 아래와 같은 연결창이 표시되며, List 객체를 초기화 하여 Variable에 저장을 해야 하기 때문에, 여기서 우측의 항목을 SetValue로 선택합니다.

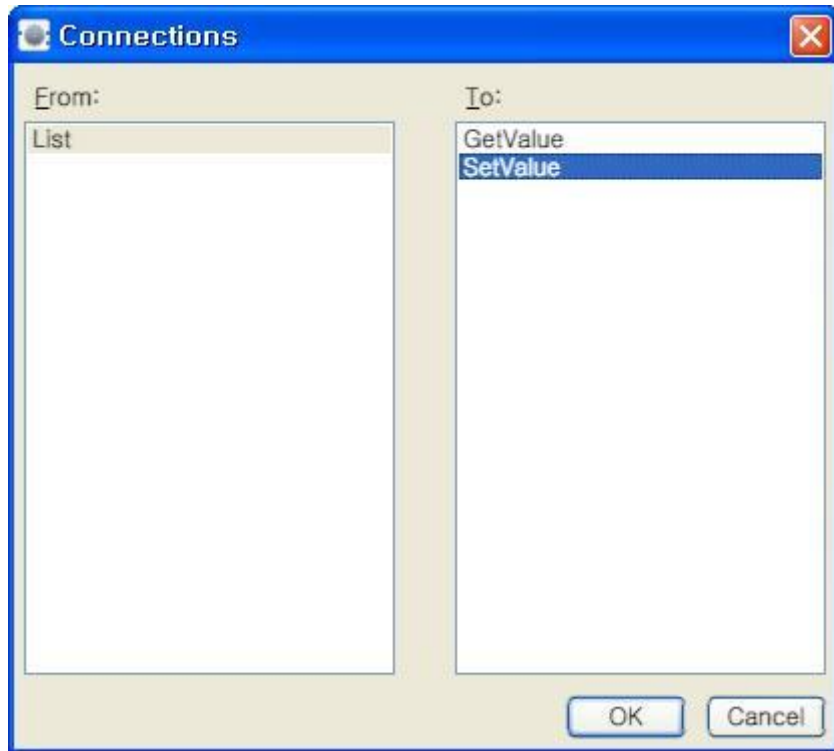


그림 3

List 액티버티를 초기화하고 strList라는 이름의 Variable 변수에 저장시키는 다이어그램은 아래와 같이 간단히 구현됩니다.



그림 4

자 이제 다음으로, Variable에 저장되어 있는 List에 값을 저장하는 과정을 설명합니다. 먼저, Variable에 저장되어 있는 값으로부터 정확한 List 타입으로 추출해 내기 위해 Calculate 액티버티를 추가하고 Variable 액티버티에 연결합니다. 연결 후에, 입력창을 마우스로 클릭하면 맨 아래에 strList 항목이 표시되며, 이 항목을 마우스로 선택합니다. 그리고 List Functions 액티버티도 추가하여 calculate 액티버티 우측에 위치시킵니다. 이때 List Functions의 여러 기능들 중 Append 항목으로 선택해 놓습니다 (기본 선택값임).

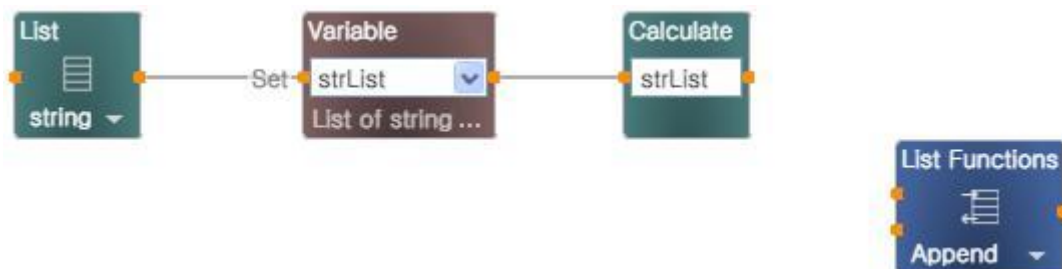


그림 5

만약, Calculate 액티버티를 거치지 않고 Variable 액티버티를 그대로 List Functions의 List 입력점에 연결하면 List Functions 액티버티에서 입력된 값을 List 타입으로 인식하지 못합니다. 이러한 문제를 해결하고 정확한 타입을 매칭시키기 위하여 Calculate 액티버티를 거쳐 List Functions 액티버티에 연결을 해야 합니다.

자 이제, Calculate 액티버티의 출력점을 List Functions 액티버티의 첫번째 입력 점에 연결합니다. 이 때 아래와 같이 연결창이 표시되면, 우측의 항목을 List로 선택합니다.

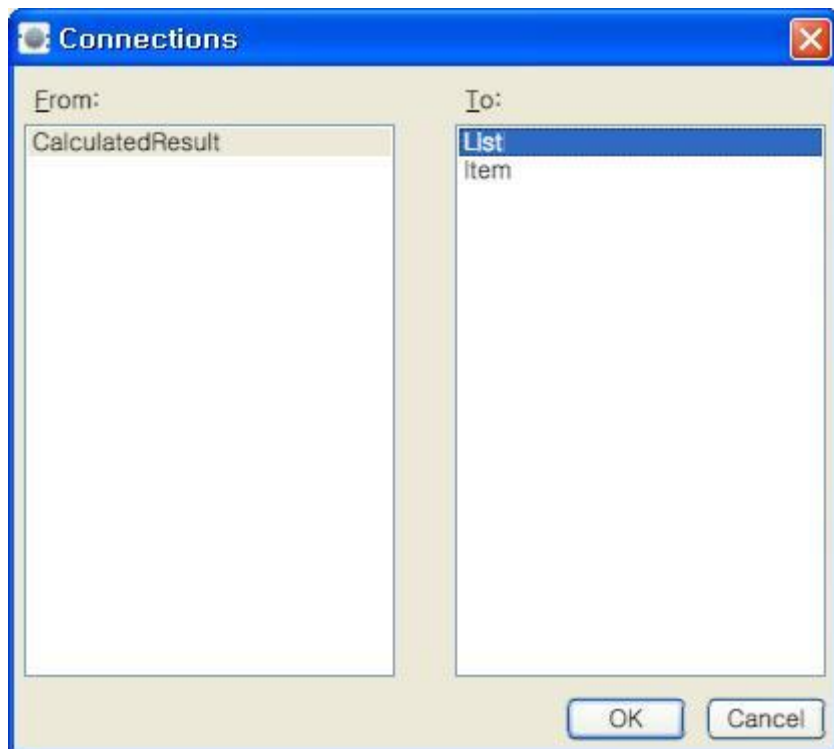


그림 6

그 다음으로는 Data 액티버티를 추가한 다음 string 타입으로 지정하고 입력창에 "Hello"라고 입력을 합니다. 입력한 후에 Data 액티버티의 출력점을 List Functions 액티버티의 나머지 하나 입력점에 연결을 합니다.

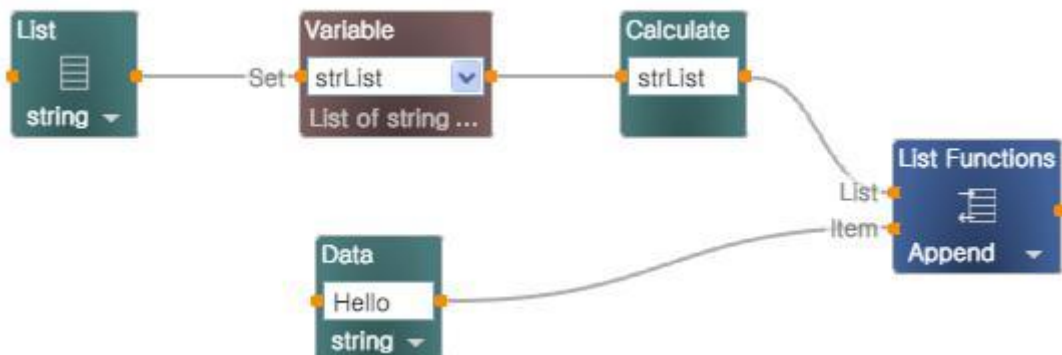


그림 7

이제 List 액티버티를 strList 라는 이름의 List of string 타입을 가지는 변수로 초기화를 하였으며, 이 변수에 "Hello" 라는 문자열을 추가하였습니다.

이 과정을 거쳤다 해서, 변수에 값이 자동으로 반영되는 것은 아닙니다. List Functions 액티버티는 해당 값을 추가한 후에 추가된 List를 결과값으로 리턴할 뿐 변수에 자동으로 저장하지 않습니다. 따라서, 아래와 같이 다시 변수에 SetValue 형태로 연결을 해주어야 합니다. Variable은 이전에 있는 Variable 액티버티를 마우스로 선택한 다음, 키보드에서 Ctrl+C, Ctrl+V를 연속으로 눌러 해당 액티버티를 복사하면 됩니다.

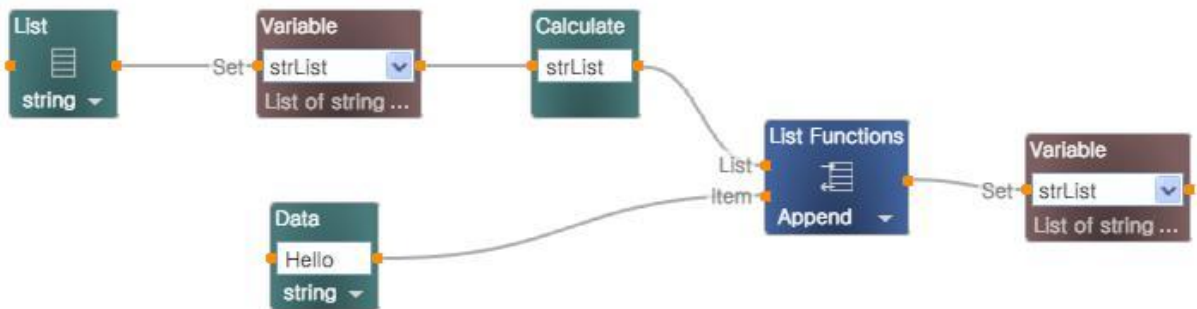


그림 8

이제, 이 변수에 저장되어 있는 값을 읽어서 표시하는 과정을 설명합니다.

아래와 같이 Calculate 액티버티를 추가한 후, 입력창에서 strList를 선택합니다. 그리고 첫 번째 값을 읽어 오기 위해 아래와 같이 strList[0] 로 내용을 수정합니다. List 객체는 일종의 배열과 같은 구조이며, 첫 번째 항목이 0으로 시작합니다.

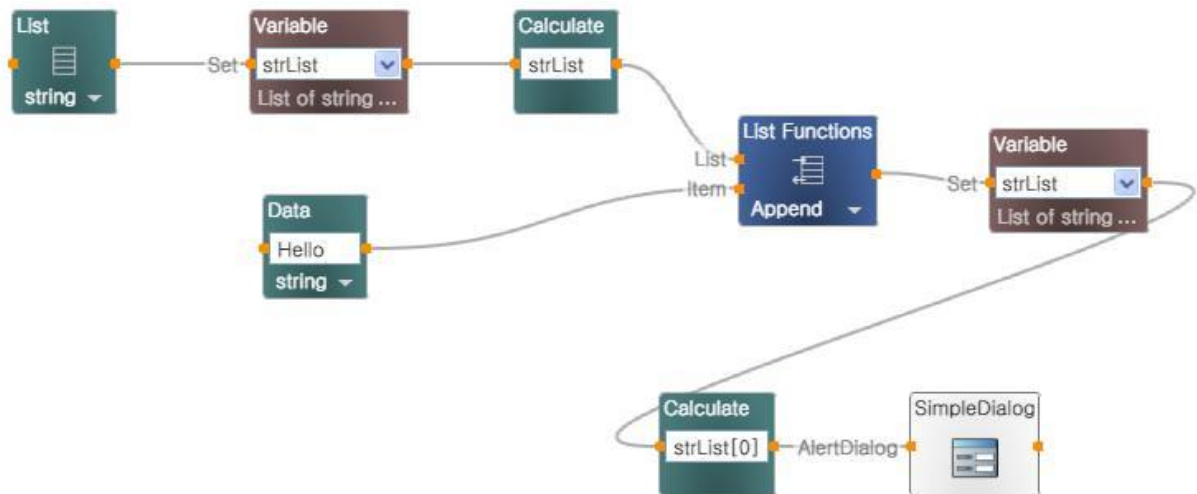


그림 9

List 객체에 저장되어 있는 항목의 갯수는 Count 속성을 이용하여 알아낼 수 있으며, 아래와 같이 해당 변수뒤에 "."을 찍으면 자동으로 Capacity와 Count 속성이 표시되며, 이 중 Count를 선택하면 됩니다. 아래 예제를 실행하면 갯수 1이 표시됩니다.

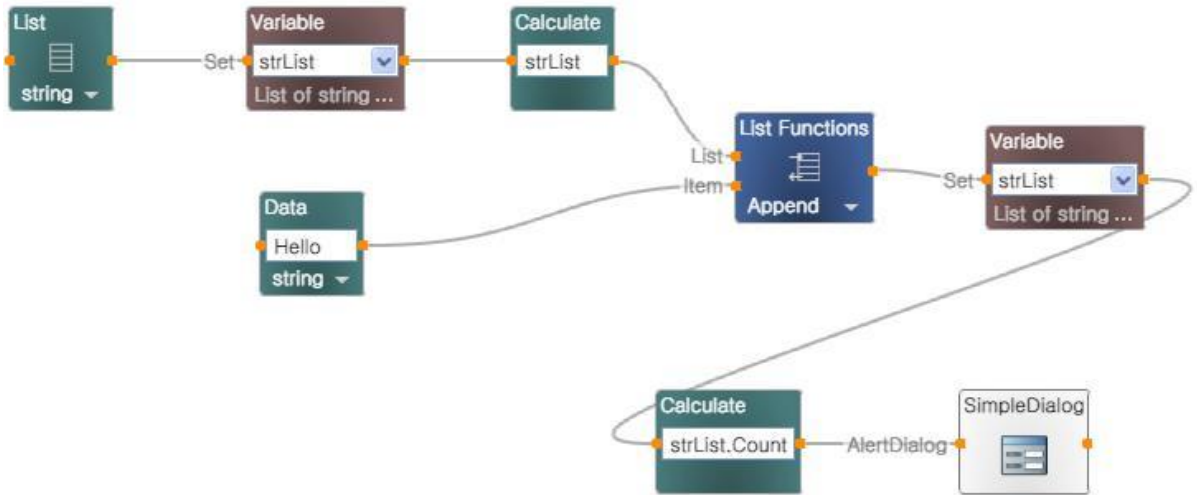


그림 10

그림 9의 다이어그램에서, List 객체의 값을 읽어 올 때, 읽어올 위치를 숫자로 지정하지 않고 아래와 같이 변수를 사용하여 지정이 가능합니다. 아래의 예는 다음에 소개할 반복문에서의 List 구현과정에서 좀 더 사용될 예정이며, 이 번 강좌에서는 기본적인 형태만 소개합니다.

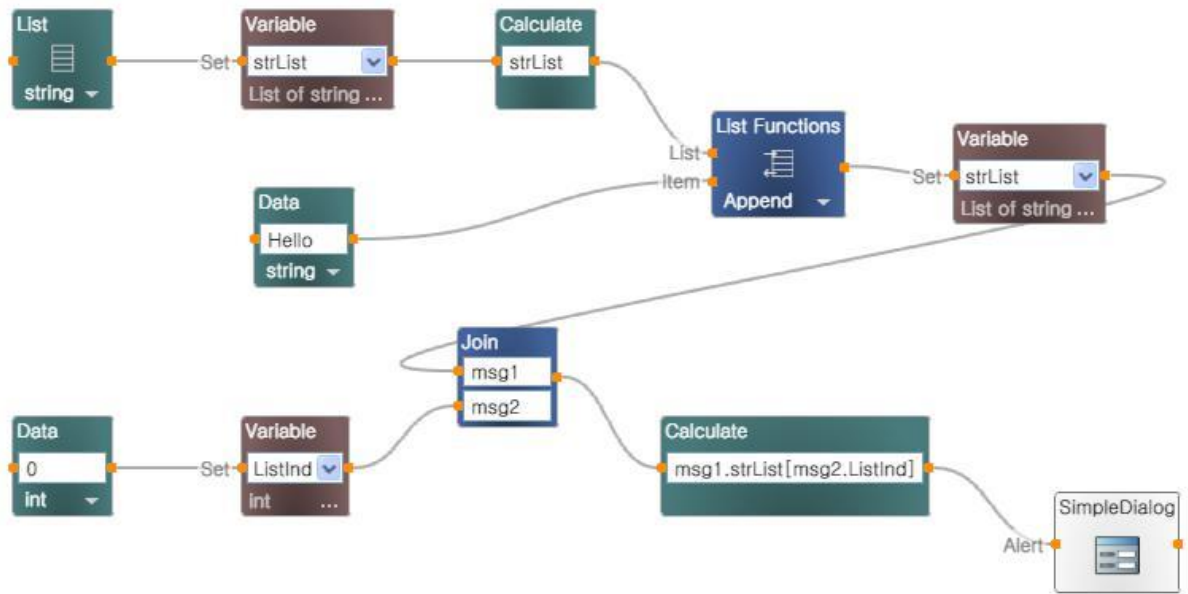


그림 11

위의 다이어그램은 Data에 0을 지정함으로써 첫 번째 값을 읽어오도록 하는 과정을 보여줍니다.