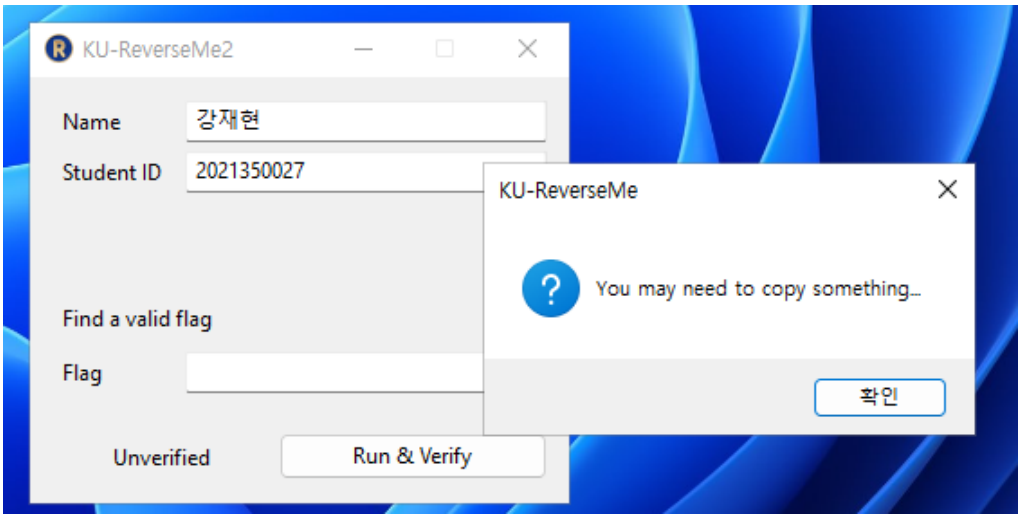
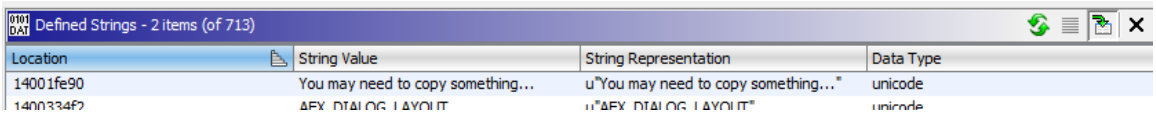
[3] Verify your inputs by pressing ‘Run&Verify’ button. How did you do that?

- 힌트로 **GetDlgItemText**가 주어졌다.
- 일단 Run&Verify 버튼을 누르면 아래와 같은 메시지창이 띄워진다. 뭐를 카피하라는 건지 모르겠다…
- 지금부터 분석하는 파일은 x64이다.



1. 메시지박스의 문장 “You may need to copy something”을 찾는다.

- IDA에서는 검색이 안됐지만, Ghidra에서는 문자열을 찾을 수 있었다.



- 해당 문자열의 위치는 `14001fe90`이었고, 그의 XREF는 `FUN_140008b80:140008cf0`이다.

```
                    u_You_may_need_to_copy_something..._14001fe90    XREF[1]:     FUN_140008b80:140008cf0(*)
14001fe90 59 00 6f        unicode     u"You may need to copy something..."
          00 75 00
          20 00 6d ...
```

2. GetDlgItemText

- GetDlgItemTextA와 GetDlgItemTextW의 XREF는 `FUN_14000a610:14000a63d`. `FUN_140008e60:140008ecf`이다.

```
                     *************************************************************
                     *                    POINTER to EXTERNAL FUNCTION           *
                     *************************************************************
                     UINT __fastcall GetDlgItemTextA(HWND hDlg, int nIDDlgIte...
          UINT               EAX:4          <RETURN>
          HWND               RCX:8          hDlg
          int                EDX:4          nIDDlgItem
          LPSTR              R8:8           lpString
          int                R9D:4          cchMax
                     334  GetDlgItemTextA  <<not bound>>
                     PTR_GetDlgItemTextA_14001f420                  XREF[1]:     FUN_140008e60:140008ecf(R)
     14001f420 ea bf 02        addr       USER32.DLL::GetDlgItemTextA
               00 00 00
               00 00
```

```
                     *************************************************************
                     *                    POINTER to EXTERNAL FUNCTION           *
                     *************************************************************
                     UINT __fastcall GetDlgItemTextW(HWND hDlg, int nIDDlgIte...
          UINT               EAX:4          <RETURN>
          HWND               RCX:8          hDlg
          int                EDX:4          nIDDlgItem
          LPWSTR             R8:8           lpString
          int                R9D:4          cchMax
                     335  GetDlgItemTextW  <<not bound>>
                     PTR_GetDlgItemTextW_14001f4a0                  XREF[1]:     FUN_14000a610:14000a63d(R)
     14001f4a0 fc bf 02        addr       USER32.DLL::GetDlgItemTextW
               00 00 00
               00 00
```

3. FUN_140008b80

- 먼저 문자열이 들어있던 함수부터 보겠다.

```
void FUN_140008b80(longlong param_1)

{
  int iVar1;
  LPCWSTR *ppWVar2;
  longlong lVar3;
  wchar_t *pwVar4;
  WCHAR *pWVar5;
  undefined auStack_178 [32];
  wchar_t *local_158;
  HANDLE local_150;
  size_t local_148;
  ulonglong *local_140;
  undefined8 *local_138;
  LPCWSTR *local_130;
  ulonglong local_128;
  undefined8 local_120;
  WCHAR local_118 [64];
  wchar_t local_98 [64];
  ulonglong local_18;

  local_18 = DAT_14002d080 ^ (ulonglong)auStack_178;
  pwVar4 = local_98;
  for (lVar3 = 0x80; lVar3 != 0; lVar3 = lVar3 + -1) {
    *(undefined *)pwVar4 = 0;
    pwVar4 = (wchar_t *)((longlong)pwVar4 + 1);
  }
  pWVar5 = local_118;
  for (lVar3 = 0x80; lVar3 != 0; lVar3 = lVar3 + -1) {
    *(undefined *)pWVar5 = 0;
    pWVar5 = (WCHAR *)((longlong)pWVar5 + 1);
```

```
  }
  FUN_14000a610((HWND *)(param_1 + 8),1000,local_118,0x40);
  if ((local_118[0] != L'\0') && (iVar1 = FUN_14000a6a0((HWND *)(param_1 + 8)), iVar1 != 0)) {
    local_150 = GetClipboardData(0xd);
    if (local_150 != (HANDLE)0x0) {
      local_158 = (wchar_t *)GlobalLock(local_150);
      local_148 = wcslen(local_158);
      if (local_148 < 0x40) {
        wcsncpy_s(local_98,0x40,local_158,local_148);
      }
      else {
        wcsncpy_s(local_98,0x40,local_158,0x3f);
      }
      GlobalUnlock(local_150);
    }
    CloseClipboard();
    iVar1 = wcscmp(local_98,local_118);
    if (iVar1 != 0) {
      local_140 = &local_128;
      local_138 = &local_120;
      local_130 = (LPCWSTR *)FUN_140009e80(local_140,0x80);
      ppWVar2 = (LPCWSTR *)FUN_140002160(local_138,L"You may need to copy something...");
      FUN_140009ef0((HWND)0x0,*ppWVar2,*local_130,0x20);
    }
  }
  FUN_14000b800(local_18 ^ (ulonglong)auStack_178);
  return;
}
```

- 이 대화 상자를 띄우지 않으려면 if문에서 `iVar==0`이면 된다. 즉, `wcscmp(local_98, local_118)=0`이면 된다.
- `local_98`은 `local_158`의 값을 0x40 만큼 복사해온다.

  ```
    if (local_148 < 0x40) {
          wcsncpy_s(local_98,0x40,local_158,local_148);
        }
        else {
          wcsncpy_s(local_98,0x40,local_158,0x3f);
        }
  ```

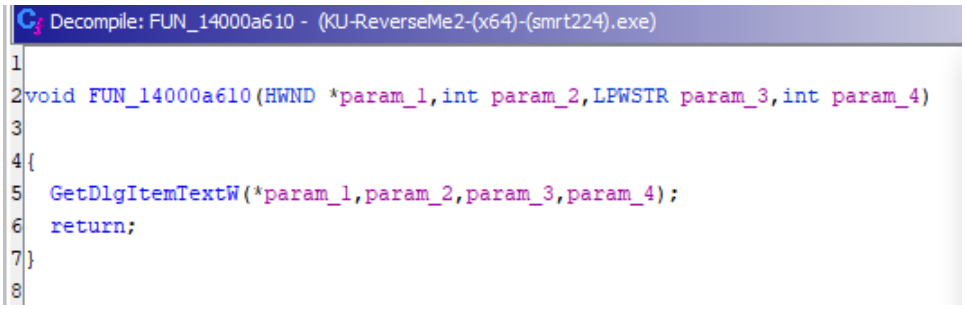
  - 그러면 local_158은 무엇일까?

    ```
    local_158 = (wchar_t *)GlobalLock(local_150);
    ```

    - **GlobalLock** 함수는 `local_150`을 잠그고, 해당 블록의 첫 번째 바이트에 대한 포인터를 반환하는 함수이다.
    - `local_150`은 `GetClipboardData(0xd)` 값으로, 직전에 내가 복사한 값을 가져오는 것이다.
- 따라서 local_98은 내가 직전에 카피한 값의 0x40바이트 만큼임을 알 수 있다.
- `local_118`은 아래와 같은 함수로 결정이 된다.

```
FUN_14000a610((HWND *)(param_1 + 8),1000,local_118,0x40);
```

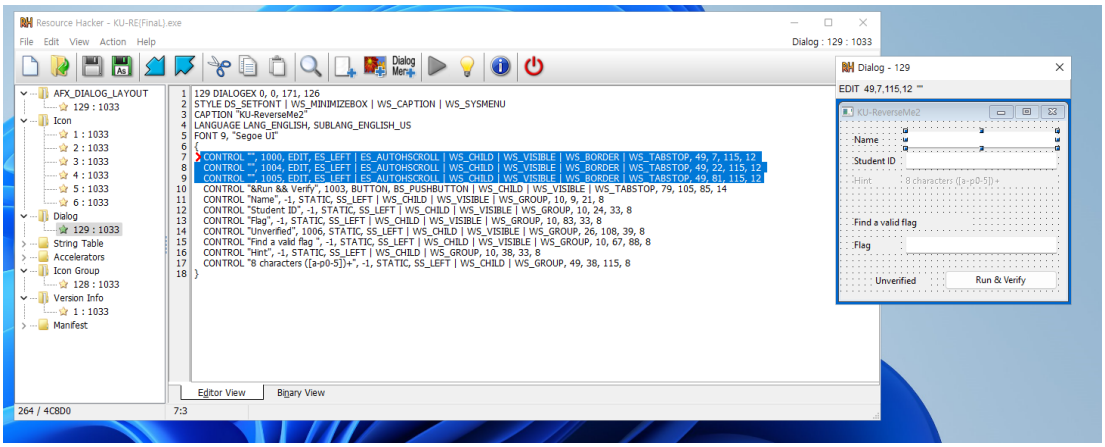
- `FUN_14000a610`을 들어가보면, **GetDlgItemTextW** 자체인 것을 알 수 있다. 따라서 잘 찾아온 것 같다.



- 즉, 아래와 같은 함수로 `local_118`이 결정된다.

```
GetDlgItemTextW((HWND *)(param_1+8), 1000, local_118, 0x40);
```

- 메시지박스에서의 `nID=1000`인 입력란은 **Resource Hacker**로 알 수 있었다.(저번 과제 리뷰 때 배움)

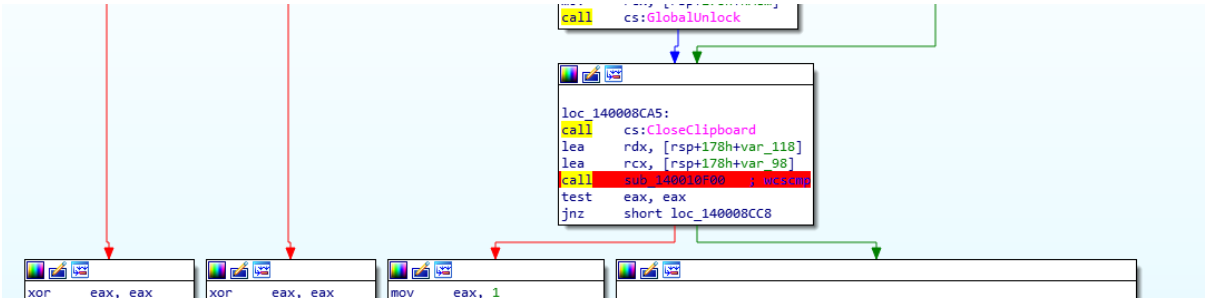


- 따라서 nID=1000의 GetDlgItemTextW는 입력한 Name값이다.

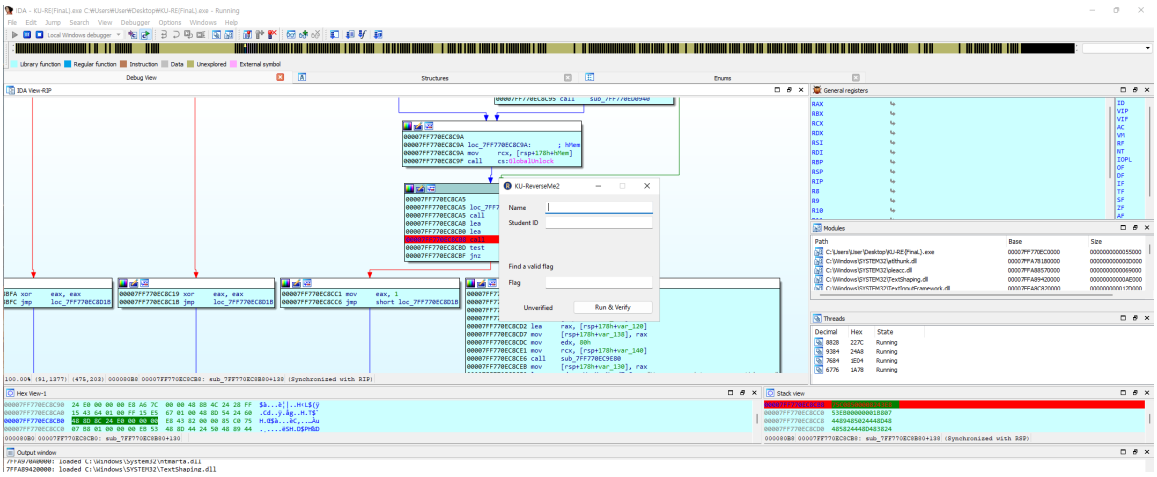
- 따라서 `local_118`은 **Name의 입력값**이다.

4. 디버깅으로 확인

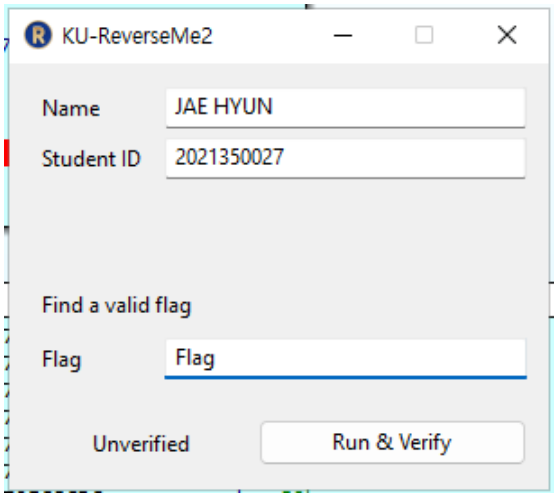
- `wcscmp` 부분에 중단점을 걸고 디버깅을 한다.

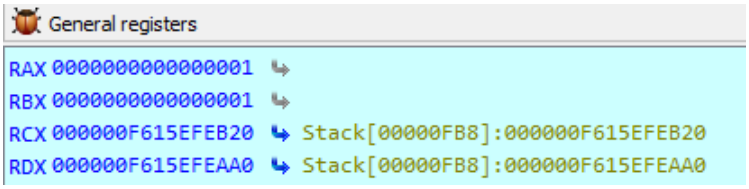


- 여기서 조금 해맸다.. 디버깅을 몇번 안 해봤던 지라... 계속 아래와 같은 화면만 뜨길래 왜이런가 했다.



  - 곰곰히 생각해보니, 이 함수는 내가 Name에다가 입력값을 넣어야 비교대상이 있는 것이니까, 올라온 메시지박스에 값을 넣고 버튼을 누르면 되는 것임을 깨달았다.
  - 따라서 **Name: JAE HYUN**, **ID: 2021350027**, **Flag:Flag**를 입력하고, **JAE HYUN**을 복사했다. 그리고 버튼을 눌렀더니, 아래와 같은 레지스터결과가 나왔다.





wcscmp에 중단점 건 register 상태

- `RCX=F615EFEB20`

```
000000F615EFEB20  002000450041004A
000000F615EFEB28  004E00550059004B
```

- `RDX=F615EFEAA0`

```
000000F615EFEAA0  002000450041004A
000000F615EFEAA8  004E00550059004B
```

- 따라서, `RCX`는 내가 카피한 값, `RDX`는 내가 입력한 Name값임을 다시 확인하게 되었다.

결과

- 버튼을 누르기 이전 카피한 값이, 메시지박스에 입력한 Name값과 같으면 오류 메시지가 뜨지 않는다.
- `Name 입력값 == 직전에 copy한 값`