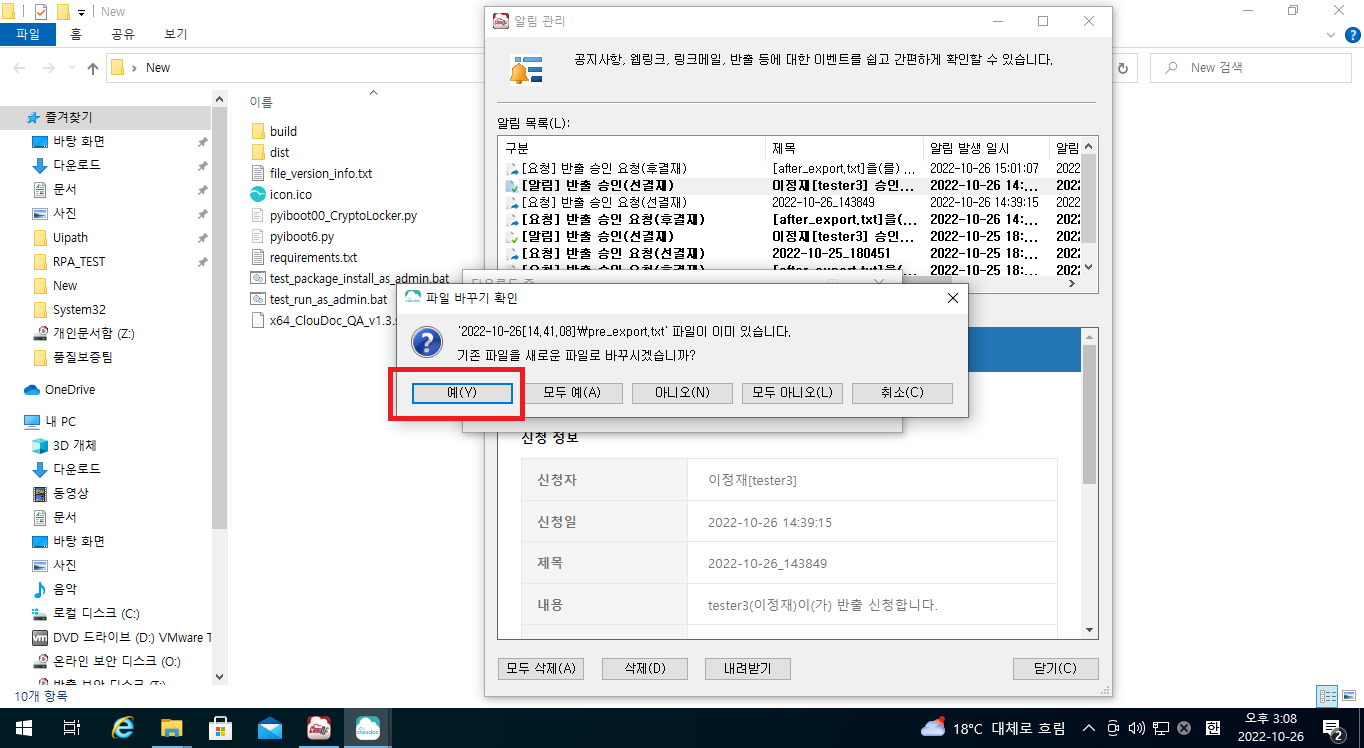
# QA\_autotest Update v1.3

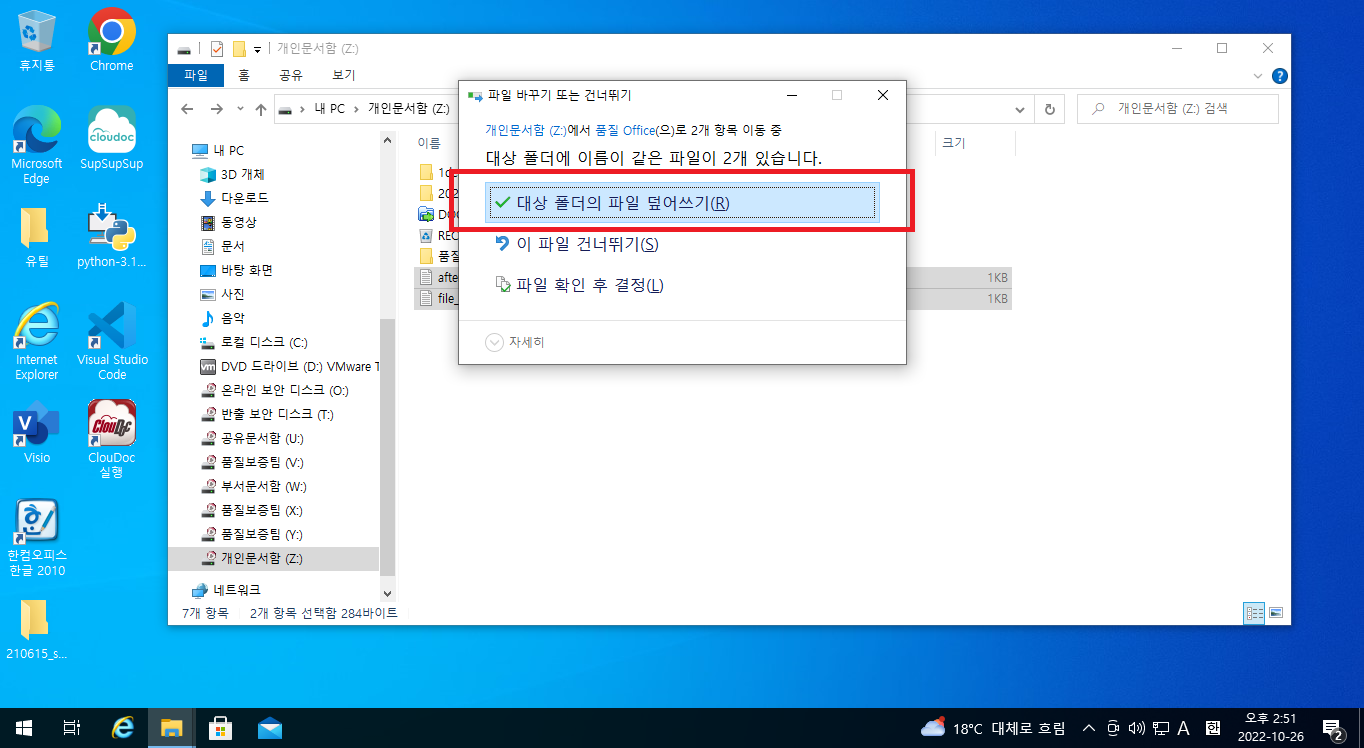
2022. 11. 02 / 작성자 : 이상운

**Step 1. 문서보안등급(dsl) on/off 파일 overwrite 처리 개선**

1. 문서보안등급(dsl) 켜져 있을 때



1. 문서보안등급(dsl) 꺼져 있을 때 (default)



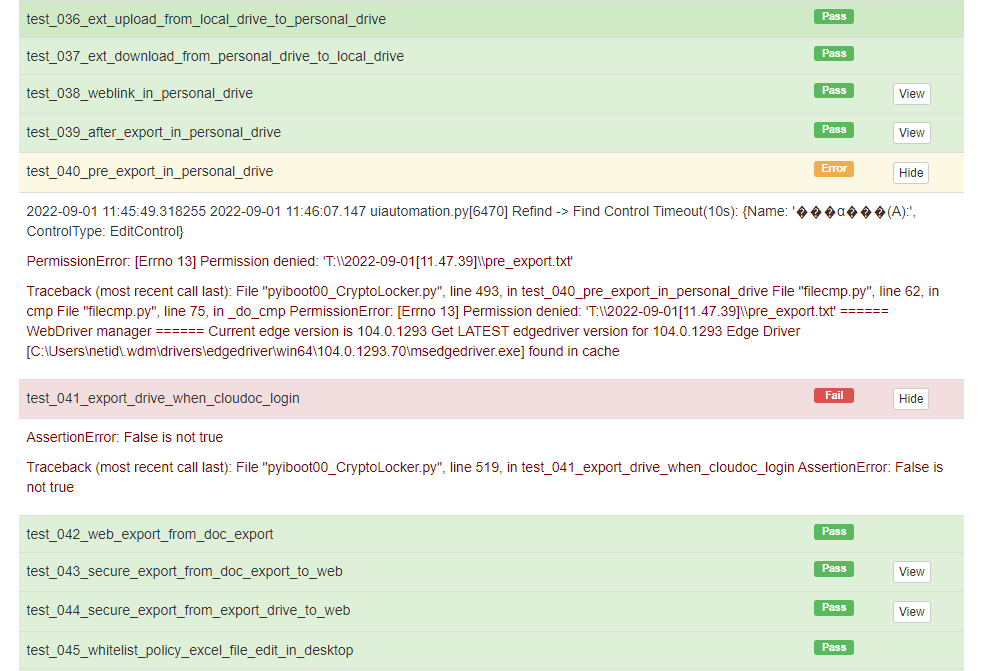
1. **업데이트 내용**
2. if overwrite==True:
3. window = uiautomation.WindowControl(searchDepth=2, Name='파일 바꾸기 확인')
4. window.ButtonControl(Name="예(Y)").Click()
5. win32gui.PostMessage(win2.\_hWnd, win32con.WM\_CLOSE, 0, 0)

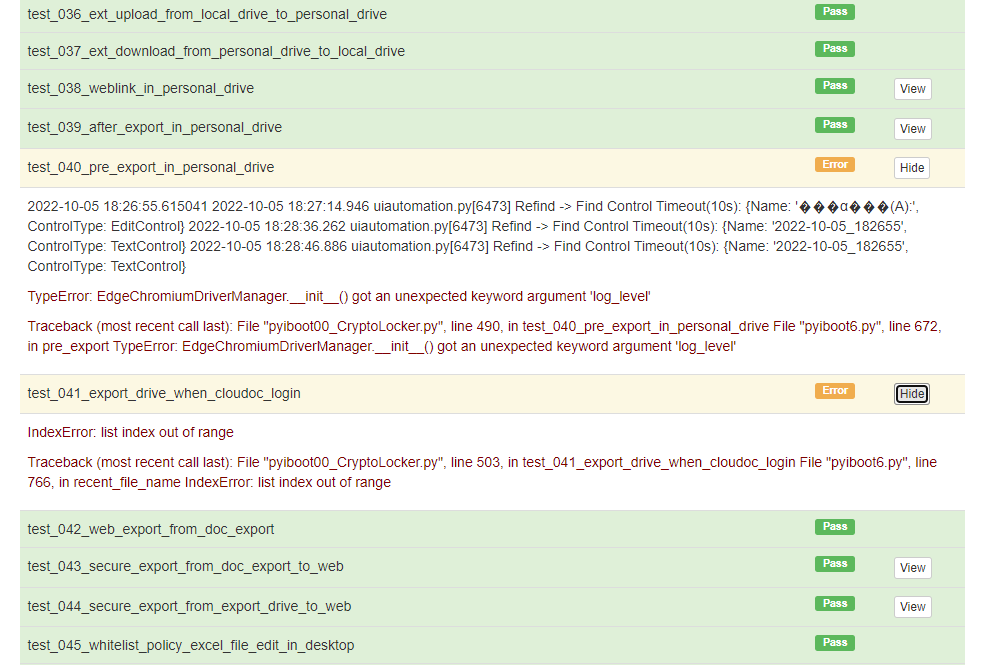
변경 전 코드는 위와 같으며 dsl off (default) 상태에서의 overwrite 대화상자를 처리하지 못함

1. **변경 후**
2. if overwrite==True:
3. try:
4. window = uiautomation.WindowControl(searchDepth=2, Name='파일 바꾸기 확인')
5. window.ButtonControl(Name="예(Y)").Click()
6. win32gui.PostMessage(win2.\_hWnd, win32con.WM\_CLOSE, 0, 0)
7. except:
8. pyautogui.hotkey('alt', 'r', interval=0.5)
10. #window.ButtonControl(Name="대상 폴더의 파일 덮어쓰기(R)").Click()
11. #win32gui.PostMessage(win2.\_hWnd, win32con.WM\_CLOSE, 0, 0)

Try – except 구문을 활용하여 dsl on 일 때는 그대로 ‘예’ 버튼을 dsl off 일 때는 대상 폴더의 파일 덮어쓰기 (R) 버튼 클릭하도록 hotkey() 설정. - 소스 출처: pyiboot6

**Step 2. 선결재 보안 반출 & 반출 보안 디스크 테스트 장애 개선**





**1. 개요**

: 클라우독 선결재 보안 반출 문서 반출 후 반출 보안 디스크에 내려받기 한 파일에 대한 비교(cmp)함수의 동작과 접근 권한이 충돌함. 테스트의 목적에 부합하면서 반출 보안 디스크의 기능을 그대로 유지하는 방향으로 소스코드 업데이트.

**2.** **불편 사항 (변경 전)**

def test\_040\_pre\_export\_in\_personal\_drive(self):

        f = open(find\_drive('개인문서함')+'pre\_export.txt', 'w')

        f.write(now)

        f.close()

        pre\_export(web\_version, addr, id, pw, find\_drive('개인문서함'), 'pre\_export.txt')

#반출신청 -> 승인 요청(웹뷰알람) -> 자가 승인(웹로그인) -> 내려받기 //크로미움 동작

exp\_folder\_name = recent\_file\_name(find\_drive('개인문서함')+"DOC\_EXPORT",0)

exist\_doc\_export = os.path.exists(find\_drive('개인문서함')+"DOC\_EXPORT\\"+exp\_folder\_name+'\\pre\_export.txt') #Return True



compare\_export\_drive = cmp(find\_drive('개인문서함')+'pre\_export.txt', find\_drive('반출 보안 디스크')+exp\_folder\_name+'\\pre\_export.txt')

#cmp -> Return True x permission denied ,비교 동작시 접근 권한 충돌



self.assertTrue(exist\_doc\_export and compare\_export\_drive) #(True and True)

        os.remove(find\_drive('개인문서함')+'pre\_export.txt') # 개인문서함 문서 삭제



    def test\_041\_export\_drive\_when\_cloudoc\_login(self):

        export\_drive = find\_drive('반출 보안 디스크')



        read\_permission = False

        write\_permission = False

        delete\_permission = False

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'r')

            f.close()

            read\_permission = True

        except PermissionError:

            pass

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'w')

            f.close()

            write\_permission = True

        except PermissionError:

            pass

        try:

            os.remove(export\_drive+recent\_file\_name(export\_drive, 0)+'\\desktop.ini')

            delete\_permission = True

        except PermissionError:

            pass

        self.assertTrue(export\_drive!=None and read\_permission and write\_permission==False and delete\_permission)

- 소스 출처: pyiboot6

**불편 사항 (변경 후)**

def test\_040\_pre\_export\_in\_personal\_drive(self):

        f = open(find\_drive('개인문서함')+'pre\_export.txt', 'w')

        f.write(now)

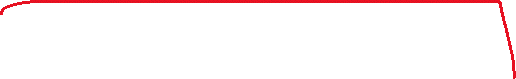
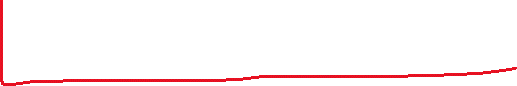
        f.close()

        pre\_export(web\_version, addr, id, pw, find\_drive('개인문서함'), 'pre\_export.txt')

exp\_folder\_name = recent\_file\_name(find\_drive('개인문서함')+"DOC\_EXPORT", 0)

exist\_doc\_export = os.path.exists(find\_drive('개인문서함')+"DOC\_EXPORT\\"+exp\_folder\_name+'\\pre\_export.txt') # exists 확인은 가능 , 권한 충돌 x

compare\_export\_drive = os.path.exists(find\_drive('반출 보안디스크')+exp\_folder\_name+'\\pre\_export.txt')



# cmp -> os. Path 로 변경 , exists 확인은 가능 , 권한 충돌 x

#compare\_export\_drive = cmp(find\_drive('개인문서함')+'pre\_export.txt', find\_drive('반출 보안 디스크')+exp\_folder\_name+'\\pre\_export.txt') <- 변경 전 코드

self.assertTrue(exist\_doc\_export and compare\_export\_drive)



        os.remove(find\_drive('개인문서함')+'pre\_export.txt')

    def test\_041\_export\_drive\_when\_cloudoc\_login(self):

        export\_drive = find\_drive('반출 보안 디스크')

        read\_permission = True #True 로 변경

        write\_permission = True

        delete\_permission =True

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'r')

            f.close()

            read\_permission = False

        except PermissionError:

            pass

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'w')

            f.close()

            write\_permission = False

        except PermissionError:

            pass

        try:

            os.remove(export\_drive+recent\_file\_name(export\_drive, 0)+'\\desktop.ini')

            delete\_permission = False

        except PermissionError:

            pass

        self.assertTrue(export\_drive!=None and read\_permission and write\_permission and delete\_permission) #True -> Return True

**3. 해결 방안**

테스트의 목적 :

1. 선 결재 보안반출로 반출 신청 후 승인된 문서가 개인 문서함의 DOC\_EXPORT 에 잘 저장이 되는가?
2. 반출 승인 후 신청한 문서를 반출 보안 디스크로 내려받기 가능한가?
3. 내려 받기 후 개인 문서함의 DOC\_EXPORT 에 저장된 파일과 반출 보안 디스크로 내려받기 한 파일이 동일한가? (최종 비교)

반출 보안 디스크의 기능 :

1. 문서 반출 승인받은 파일은 해당 문서함 DOC\_EXPORT 에 생성된다. 이 파일들을 로컬 영역이 아닌, ClouDoc 서버랑 통신이 안되는 곳에서 열람 가능하도록 사용하는 문서함이 반출 보안 디스크이다.
2. DOC\_EXPORT 속 파일 -> 반출 보안 디스크
3. ClouDoc 로그아웃 후, 노트북을 들고 출장 등 사유로 먼 곳으로 떠난다.
4. 반출 보안 디스크만 마운트 되어 있으면 반출된 파일이 열람 가능하나 로컬 영역이 아니기때문에 보안은 유지.

소스 코드 변경 :

* 1. 변경 전 cmp 비교 동작 -> 서버랑 통신 중인 테스트 환경에서 반출 보안 디스크로 접근 권한 충돌

compare\_export\_drive = cmp(find\_drive('개인문서함')+'pre\_export.txt', find\_drive('반출 보안 디스크')+exp\_folder\_name+'\\pre\_export.txt')

* 1. 반출 보안 디스크의 기능과 로컬에서의 접근 제한은 유지하면서 cmp 가 아니 os.path 로 파일 존재 여부 확인

compare\_export\_drive = os.path.exists(find\_drive('반출 보안디스크')+exp\_folder\_name+'\\pre\_export.txt')

* 1. 반출 보안 디스크 내 권한 검사 (변경 전)

 def test\_041\_export\_drive\_when\_cloudoc\_login(self):

        export\_drive = find\_drive('반출 보안 디스크')

        read\_permission = False

        write\_permission = False

        delete\_permission = False

False 처리 후 False 가 맞는지 검사 방식

 try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'r')

            f.close()

            read\_permission = True

        except PermissionError:

            pass

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'w')

            f.close()

            write\_permission = True

        except PermissionError:

            pass

        try:

            os.remove(export\_drive+recent\_file\_name(export\_drive, 0)+'\\desktop.ini')

            delete\_permission = True

        except PermissionError:

            pass

        self.assertTrue(export\_drive!=None and read\_permission and write\_permission==False and delete\_permission)

* 1. 반출 보안 디스크 내 권한 검사 (변경 후)

def test\_041\_export\_drive\_when\_cloudoc\_login(self):

export\_drive = find\_drive('반출 보안 디스크')

read\_permission = True

write\_permission = True

delete\_permission = True

True 처리 후 True 가 맞는지 검사 방식

 try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'r')

            f.close()

            read\_permission = False

        except PermissionError:

            pass

        try:

            f = open(export\_drive+recent\_file\_name(export\_drive, 0)+'\\pre\_export.txt', 'w')

            f.close()

            write\_permission = False

        except PermissionError:

            pass

        try:

            os.remove(export\_drive+recent\_file\_name(export\_drive, 0)+'\\desktop.ini')

            delete\_permission = False

        except PermissionError:

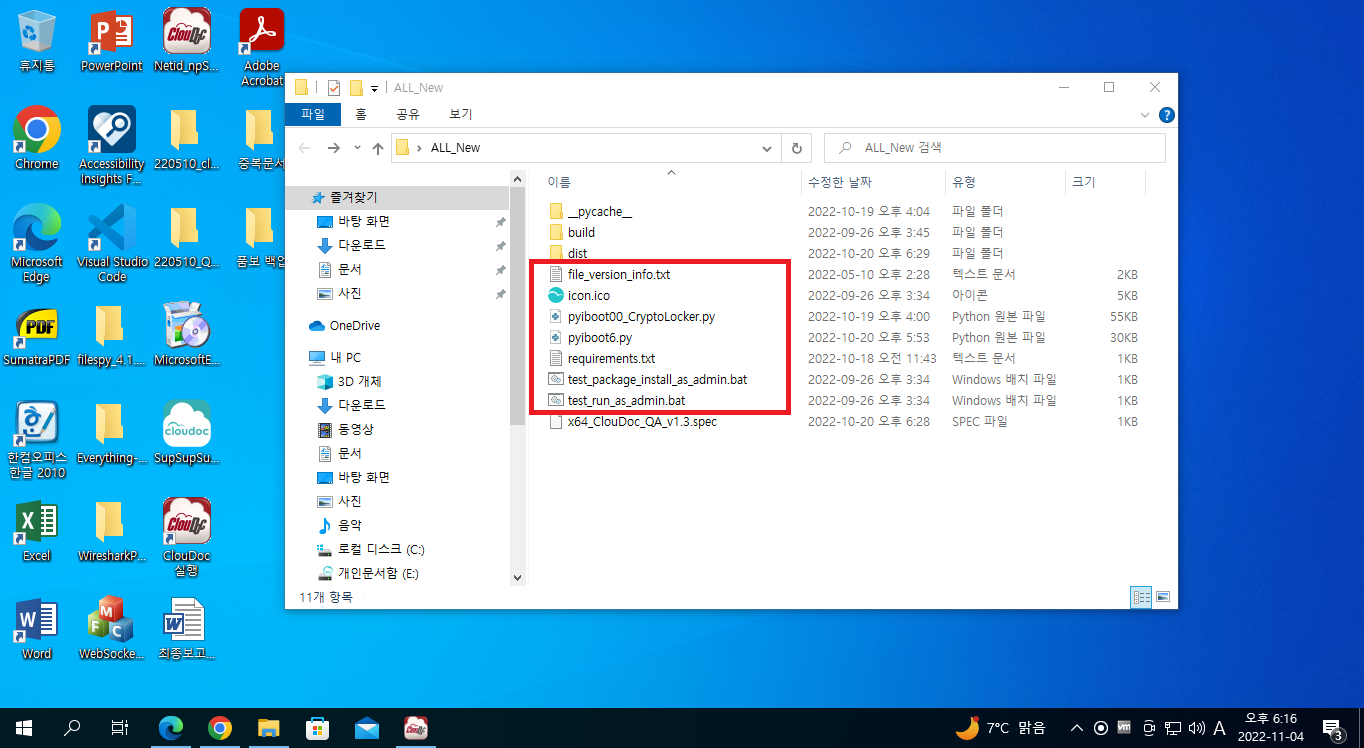
            pass

        self.assertTrue(export\_drive!=None and read\_permission and write\_permission and delete\_permission) #True -> True 권한 체크

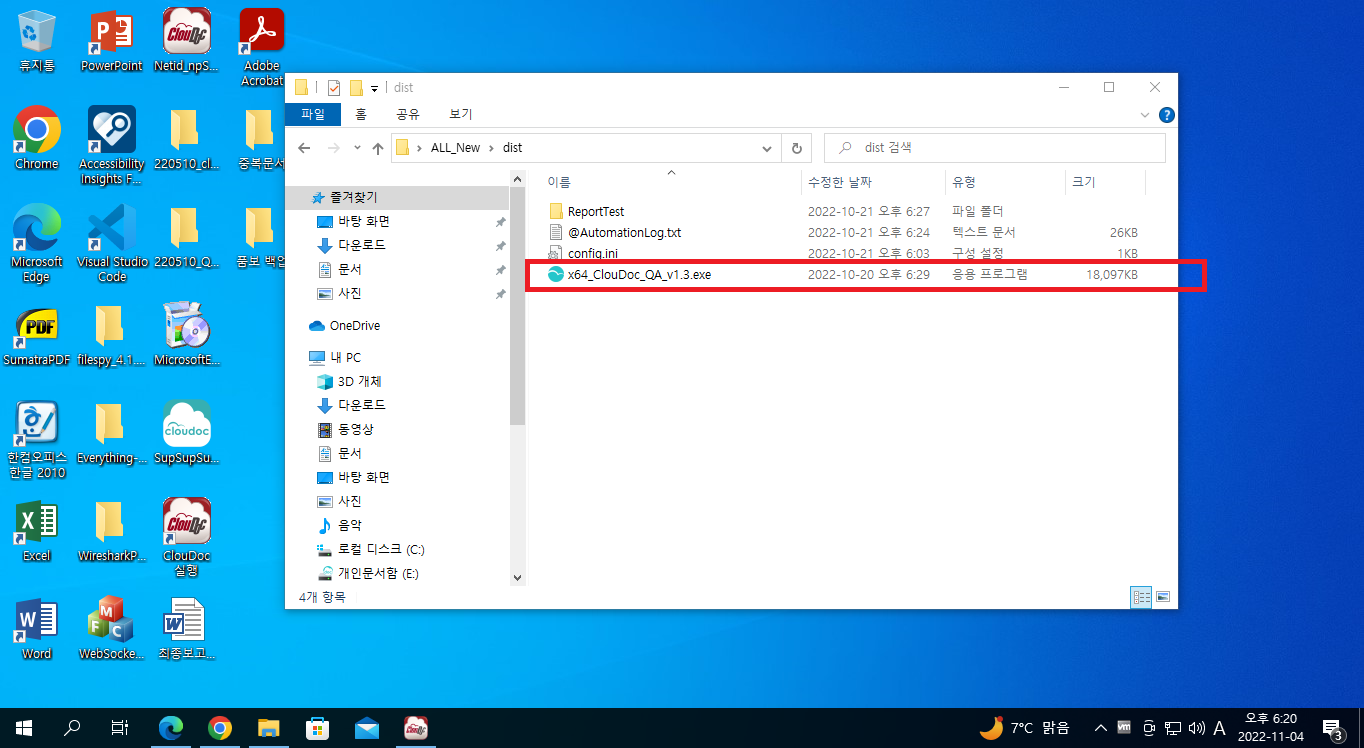
**Step 3. 업데이트 후 Python Script 배포**

**1. 개요**

: 수정 및 업데이트 한 Python 스크립트를 test\_package\_install\_as\_admin.bat[[1]](#endnote-1) 파일을 이용해 exe 파일로 배포한다. 테스트를 원하는 VM 윈도우에 폴더를 생성하고 install 포함 배치파일 2개, file\_version\_info , requirements , icon , pyiboot00\_CryptoLocker , pyiboot6를 생성한 폴더에 다운로드 한다.



배치파일을 통해 exe 파일 생성에 성공하면 dist 폴더가 생기고 그 안에 exe 파일이 생성된다.



1. **Pyinstaller EXE 파일 배포 명령어 (64bit ,32bit)**
2. pyinstaller --noconfirm --onefile --console --icon "icon.ico" --name "x64\_ClouDoc\_QA\_v1.3" --clean --key "TeamSQA" --version-file "file\_version\_info.txt"
3. pyinstaller --noconfirm --onefile --console --icon "icon.ico" --name "x86\_ClouDoc\_QA\_v1.3" --clean --key "TeamSQA" --version-file "file\_version\_info.txt" "pyiboot00\_CryptoLocker.py"

**Redmine 페이지에서 결함 목록을 CSV 형식으로 다운로드, 데이터 처리**

1. “UiPath.Core.Activities.OpenBrowser” 로 Redmine 페이지에서 결함 목록 CSV 파일 다운로드
2. “UiPath.CSV.Activities.ReadCsvFile” 로 다운로드한 데이터 파일을 읽고 주차 별 결함 개수 계산
3. A~B 스텝을 서버, 클라이언트, 모바일 결함 페이지에서 각각 수행

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **보고 메일 본문 작성**
2. CSV 파일에서 처리한 주차 별 취합된 결함 개수 데이터를 미리 준비된 워드 양식에 입력
3. “UiPath.Excel.Activities” 로 CSV 파일을 엑셀 표 형식으로 변환 후 준비된 워드 양식에 삽입
4. **아웃룩 메일 작성**
5. “UiPath.Mail.Activities.Business.OutlookApplicationCard” 로 로컬 아웃룩 계정 연결
6. 수신인, 참조, 제목 및 작성한 본문 문서 연결
7. 상황에 따라 초안으로 저장하여 보관하거나 작성한 메일을 바로 전송

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **Uipath Orchestrator 연결 (반복 작업 스케줄 등록)**
2. “<https://cloud.uipath.com/>”에서 Add machine, Add robot을 클릭하여 머신, 로봇 생성
3. Uipath Assistant 환경설정 이동 후 생성한 머신 키를 통해 PC와 연결
4. Uipath Studio에서 작성한 자동화 스크립트를 My Workspace에 게시 (Publish)
5. 게시된 프로세스로 트리거 생성 (매주 금요일, 오전 9시 반복)

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**3. 공유 경로( Y: 100번 서버 부서문서함)**

1. RPA 설계 파일: Y:\8. QA\7. Uipath\_RPA\Bug\_Report
2. RPA 설계 문서: Y:\8. QA\7. Uipath\_RPA\Bug\_Report\참고문서

1. Y:\8. QA\0. 인수인계\cloudoc\_test\_code 경로에 존재하는 배치파일 [↑](#endnote-ref-1)