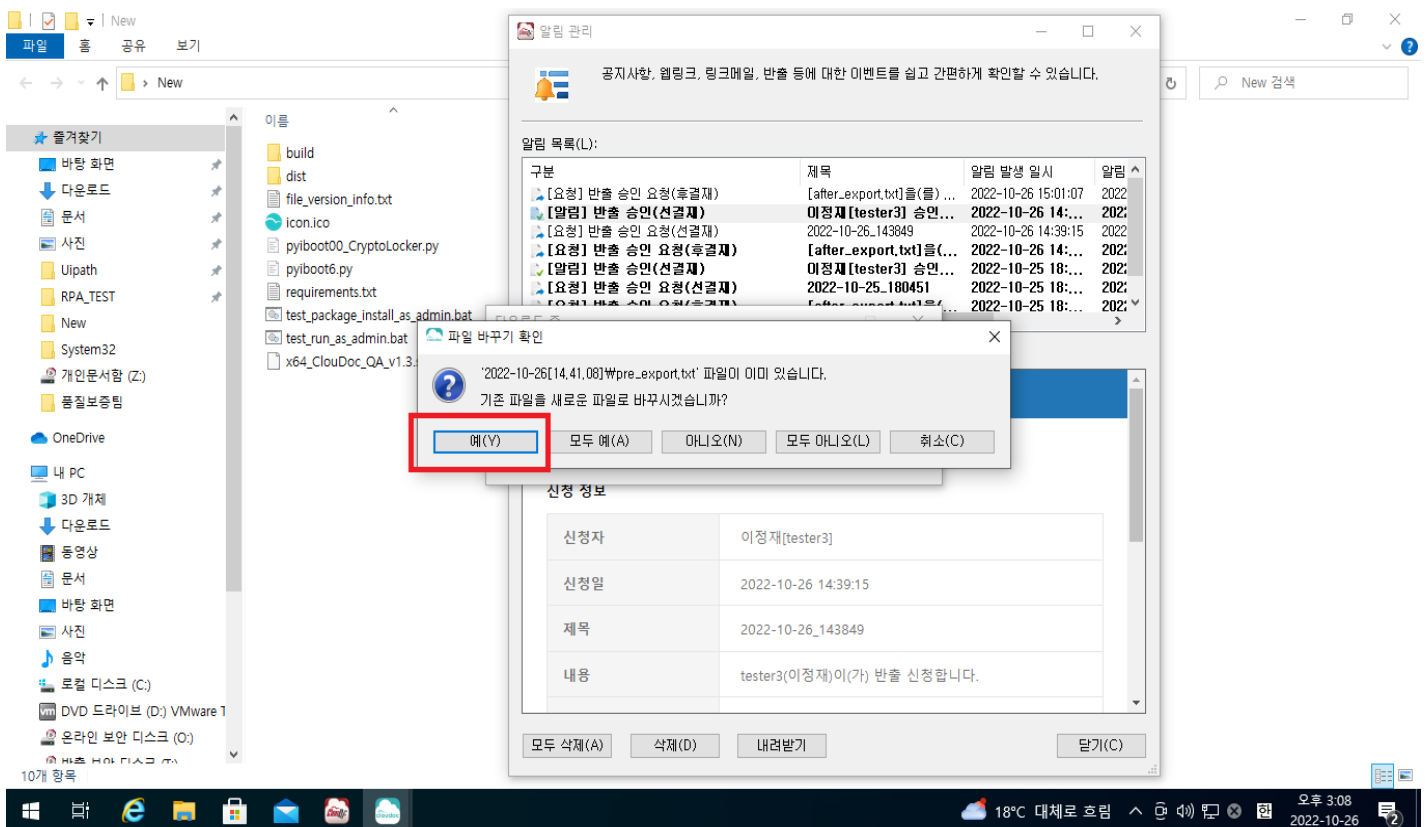


# QA\_autotest Update v1.3

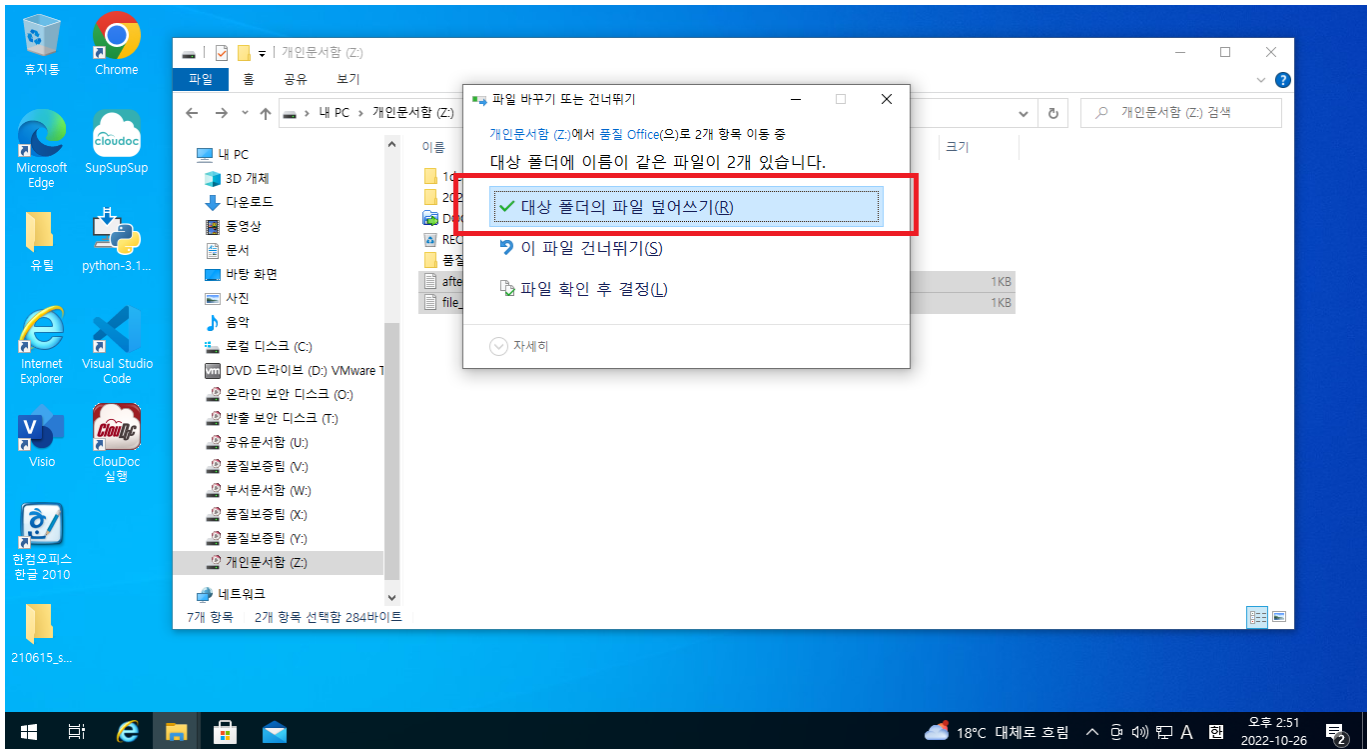
2022. 11. 02 / 작성자 : 이상윤

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

A. 문서보안등급(dsl) 켜져 있을 때



B. 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)
⑥

```

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

#### A. 변경 후

```

B. if overwrite==True:
C.     try:
D.         window = uiautomation.WindowControl(searchDepth=2, Name='파일 바꾸기
확인')
E.         window.ButtonControl(Name="예(Y)").Click()
F.         win32gui.PostMessage(win2._hWnd, win32con.WM_CLOSE, 0, 0)
G.     except:
H.         pyautogui.hotkey('alt', 'r', interval=0.5)
I.
J.         #window.ButtonControl(Name="대상 폴더의 파일 덮어쓰기(R)").Click()
K.         #win32gui.PostMessage(win2._hWnd, win32con.WM_CLOSE, 0, 0)
L.

```

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



## 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 내려 받기 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 문서함 이 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 :

- A. 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')
```

- B. 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')
```

- C. 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)
```

- D. 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 된 Python Script 배포

### 1. 개요

: 매주 금요일 마다 작성하는 품질보증팀 결함 보고 메일을 RPA 자동화로 대체  
 기획이나 의사 결정이 필요 없는 단순 반복 업무를 RPA 를 이용해 자동으로 처리. 단순 반복 업무 비  
 중 감소에 따른 직원 만족도 제고 및 늘어난 인력 리소스는 고부가가치 업무로 전환  
 품질보증팀에서 도입한 RPA 프로세스 및 방법을 공유하여 다른 팀에서도 단순 반복 업무를 RPA 로 전  
 환할 수 있도록 도움

### 2. RPA 자동화 설계 방법 (품질보증팀 주간 결함 보고 메일 보내기)

#### ① Redmine 페이지에서 결함 목록을 CSV 형식으로 다운로드, 데이터 처리

A. "UiPath.Core.Activities.OpenBrowser" 로 Redmine 페이지에서 결함 목록 CSV 파일 다운로드



- B. "UiPath.CSV.Activities.ReadCsvFile" 로 다운로드한 데이터 파일을 읽고 주차 별 결함 개수 계산
- C. A~B 스텝을 서버, 클라이언트, 모바일 결함 페이지에서 각각 수행

```

if startday > Now.AddDays(-1*(7+(Now.DayOfWeek-DayOfWeek.Sunday)) mod 7)
    this_week = this_week+1
else startday > Now.AddDays(-1*(7+(Now.DayOfWeek-DayOfWeek.Sunday)) mod 7).AddDays(-7)
    one_weeks_ago = one_weeks_ago+1
else startday > Now.AddDays(-1*(7+(Now.DayOfWeek-DayOfWeek.Sunday)) mod 7).AddDays(-14)
    two_weeks_ago = two_weeks_ago+1
else startday > Now.AddDays(-1*(7+(Now.DayOfWeek-DayOfWeek.Sunday)) mod 7).AddDays(-21)
    three_weeks_ago = three_weeks_ago+1

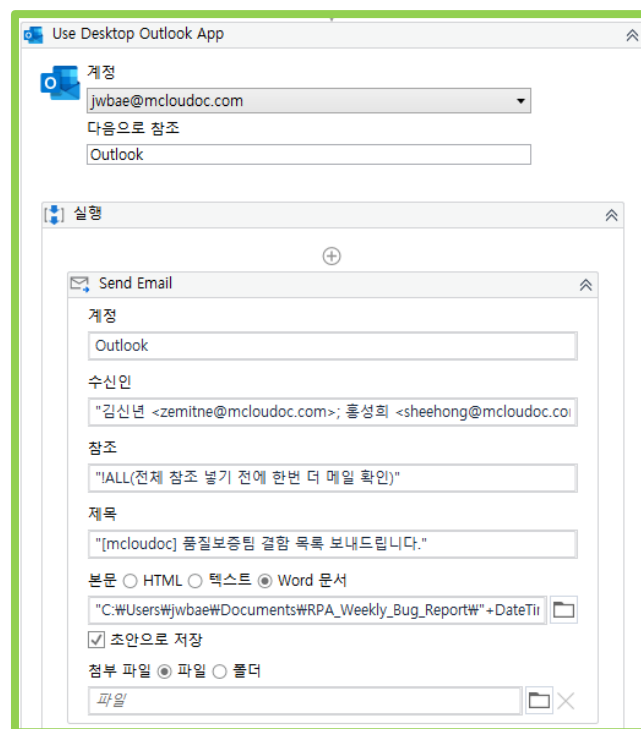
```

## ② 보고 메일 본문 작성

- A. CSV 파일에서 처리한 주차 별 취합된 결함 개수 데이터를 미리 준비된 워드 양식에 입력
- B. "UiPath.Excel.Activities" 로 CSV 파일을 엑셀 표 형식으로 변환 후 준비된 워드 양식에 삽입

## ③ 아웃룩 메일 작성

- A. "UiPath.Mail.Activities.Business.OutlookApplicationCard" 로 로컬 아웃룩 계정 연결
- B. 수신인, 참조, 제목 및 작성한 본문 문서 연결
- C. 상황에 따라 초안으로 저장하여 보관하거나 작성한 메일을 바로 전송



#### ④ UiPath Orchestrator 연결 (반복 작업 스케줄 등록)

- "<https://cloud.uipath.com/>"에서 Add machine, Add robot 을 클릭하여 머신, 로봇 생성
- UiPath Assistant 환경설정 이동 후 생성한 머신 키를 통해 PC 와 연결
- UiPath Studio 에서 작성한 자동화 스크립트를 My Workspace 에 게시 (Publish)
- 게시된 프로세스로 트리거 생성 (매주 금요일, 오전 9 시 반복)

프로세스	호스트 이름	작업 형식	런타임 라이선스	상태	우선 순위	시작됨	종료됨	소스	액션
QA_BUG_REPORT	DESKTOP-9JEHSMT	원격 개인	개발	성공	일반	3시간 전	3시간 전	수동	①
QA_BUG_REPORT	DESKTOP-9JEHSMT	원격 개인	개발	오류 발생	일반	3시간 전	3시간 전	수동	①
QA_BUG_REPORT	DESKTOP-9JEHSMT	원격 개인	개발	오류 발생	일반	3시간 전	3시간 전	수동	①
QA_TEAM	DESKTOP-9JEHSMT	원격 개인	개발	성공	일반	3시간 전	3시간 전	수동	①
QA_TEAM	DESKTOP-9JEHSMT	원격 개인	개발	성공	일반	3시간 전	3시간 전	수동	①
QA_TEAM	DESKTOP-9JEHSMT	원격 개인	개발	중지됨	일반	4시간 전	4시간 전	수동	①
QA_TEAM	DESKTOP-9JEHSMT	원격 개인	개발	중지됨	일반	4시간 전	4시간 전	수동	①

### 3. 공유 경로( Y: 100 번 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