

프로젝트 sort의 다음 작업을 이어서 진행한다.

작업 기준 ZIP

최신 기준 파일:

```
sort_local_v283_chrome_session_download_ws_move.zip
```

새 채팅에 이 ZIP을 첨부하고 작업한다.

현재 기준 버전:

```
ZIP: v283  
manifest version: 2.3.8.2830  
화면 버전: v283
```

이전 채팅에서는 v284 작업 시작 계획만 출력했으며, **v284 실제 소스 수정은 시작하지 않았다.**

이번 단일 목표

v284: 단일 Instagram 탭 + Chrome 실제 다운로드 경로 처리

기존의 새 탭 생성·OWNER 인계·이전 탭 종료 구조를 완전히 제거한다.

자동 실행은 처음 선택된 Instagram 탭 하나를 계속 사용한다.

```
현재 Instagram tab_id 고정  
→ 같은 탭에서 다음 계정 URL로 이동  
→ Content/Main READY 확인  
→ 다음 계정 수집
```

1. 폐기할 탭 분리 구조

다음 로직은 미련 없이 제거한다.

```
새 Instagram 탭 생성  
bootstrap 탭 생성  
generation 증가  
HANDOFF_READY  
OWNER_COMMITTED
```

RETIRED
OLD_TAB_CLOSED
pending handoff
handoff token
이전 탭 종료
OWNER 인계 복구
최근 탭 인계 이력
다중 OWNER 정리

관련 상태값·메시지·타이머·저장 키·진단 UI·테스트도 함께 제거한다.

사용자가 직접 열어 둔 다른 Instagram 탭은 건드리지 않는다.

현재 자동 실행이 시작된 탭 하나만 `tab_id`를 고정해서 사용한다.

계정 이동은 새 탭 생성이 아니라 현재 탭 URL 변경으로 처리한다.

2. unload 관련 판단

WS8771의 Python 직접 다운로드 방식이 Chrome 수동 다운로드와 다른 세션으로 요청하면서 문제가 발생한 것으로 판단한다.

v283에서 Chrome 세션 다운로드로 변경한 결과:

3/3 계정 순회 완료
파일 18/18 완료
자동종료 완료
탭 OWNER 인계도 완료

실제 화면 결과:

전체 실행: 게시물 9개 · MP4 8개 · JPG 8개
파일 18/18 · 100%
상태: Chrome session download · WS8771 move complete

3번째 계정은 게시물이 0개였으며 정상 자동종료했다.

따라서 탭 생성·종료로 unload를 피하는 구조는 더 이상 사용하지 않는다.

기존 CDP unload 추적·차단 코드가 단일 탭 이동에 필요하지 않은지 실제 소스를 확인한다.

불필요한 탭 종료용 unload guard는 제거한다.

단, 다른 기능에서 실제로 사용하는 CDP 기능까지 임의로 제거하지 않는다.

3. 다운로드 방식

v283은 Chrome 다운로드를 사용하지만 `YELLOW_TEMP` 임시 폴더를 강제로 지정했다.

v284에서는 특정 임시 폴더를 지정하지 않는다.

Chrome이 실제로 저장한 위치를 `downloadId` 로 확인한다.

처리 순서:

```
chrome.downloads.download()  
→ downloadId 확보  
→ chrome.downloads.onChanged에서 state=complete 확인  
→ chrome.downloads.search({ id: downloadId })  
→ DownloadItem.filename으로 실제 절대 경로 확인  
→ 실제 완료 파일 경로를 WS8771에 전달  
→ WS8771이 기존 최종 폴더로 이동  
→ 최종 파일 존재·크기 확인  
→ 완료 집계
```

폴더를 감시하거나 가장 최근 파일을 추측하지 않는다.

각 다운로드 요청은 반드시 `downloadId` 기준으로 연결한다.

저장할 작업 정보:

```
downloadId  
account  
shortcode  
asset: mp4 또는 jpg  
원래 파일명  
post_url  
media_url  
tab_id  
run_seq  
최종 저장 예정 경로
```

여러 다운로드가 동시에 실행돼도 MP4와 JPG가 뒤바뀌지 않아야 한다.

4. Chrome 저장 팝업 자동 처리

실제 v283 테스트에서 다운로드마다 Chrome 저장 팝업이 나타났고, 사용자가 직접 Enter를 눌러 18/18을 완료했다.

WS8771 설정에 기존 항목이 있다.

저장 자동처리: 자동

이 설정을 다시 실제 동작에 연결한다.

자동 설정

Chrome 저장 팝업 감지

- WS8771 설정이 자동인지 확인
- 현재 선택된 기본 위치·파일명은 변경하지 않음
- Enter 입력
- Chrome 다운로드 완료 대기
- downloadId로 실제 저장 경로 확인
- WS8771 최종 이동

WS가 임의 경로나 파일명을 입력하면 안 된다.

팝업의 현재 기본 선택값을 그대로 승인한다.

수동 설정

저장 팝업을 건드리지 않음

- 사용자가 직접 위치 선택 및 저장
- 완료 후 downloadId로 실제 경로 확인
- WS8771 후처리

팝업 자동 감지는 기존 WS8771의 창 감지·키 입력 구조를 실제 소스에서 확인하고 재사용한다.

함수명이나 파일 위치는 소스 확인 전 단정하지 않는다.

5. 자동 다운로드와 수동 다운로드 단일화

자동과 수동은 동일한 Chrome 다운로드 함수를 사용해야 한다.

수동 버튼

자동 실행

↓

공통 Chrome 다운로드 함수

↓

downloadId 완료 확인

↓
WS8771 최종 이동

WS8771이 URL을 Python으로 다시 요청하는 경로는 자동 다운로드에서 사용하지 않는다.

구버전 호환 수신 코드가 남아 있다면 실제 호출 여부를 확인하고:

- 호출되지 않지만 호환을 위해 필요한 경우 유지
- 완전히 폐기 가능하면 `ai_read_삭제.md`에 기록 후 제거

실패를 조용한 fallback으로 Python 직접 다운로드로 돌리지 않는다.

6. 완료 판정

Chrome 다운로드 시작 성공만으로 완료 처리하지 않는다.

완료 순서:

```
downloadId 생성
→ Chrome state=complete
→ 실제 filename 조회
→ .crdownload 아님
→ 파일 존재
→ 파일 크기 > 0
→ WS8771 이동 성공
→ 최종 목적지 파일 존재
→ 임시 원본 처리 확인
→ MP4/JPG 완료 집계
```

실패는 성공처럼 집계하지 않는다.

7. 기능 체크 로그

첨부된 `로그활용지침.txt` 기준을 따른다.

반복 상세 로그를 남기지 않고 체크형 로그만 사용한다.

예:

```
[단일탭전환][ST-01][체크완료] 자동 실행 탭 고정 / tab_id=...
[단일탭전환][ST-02][체크완료] 같은 탭 계정 이동 / 계정=gonggan__gyeol
[단일탭전환][ST-03][체크완료] Main READY 확인
```

[Chrome다운로드][CD-01][체크완료] 다운로드 요청 / downloadId=...
[Chrome다운로드][CD-02][체크완료] Chrome 저장 완료 / 실제경로=...
[Chrome다운로드][CD-03][체크완료] WS8771 최종 이동
[Chrome다운로드][CD-04][체크완료] 최종 파일 확인

[저장자동처리][SA-01][체크완료] 저장 팝업 감지
[저장자동처리][SA-02][체크완료] 자동 설정 확인
[저장자동처리][SA-03][체크완료] 기본 선택값 Enter 승인

쿠키·세션·민감값은 로그에 원문으로 남기지 않는다.

8. UI 정리

MAIN의 기존 통합 상태 UI는 유지한다.

현재 v282부터 적용된 형태:

CDP:0 · 탭 N · OWNER tab ...

그러나 OWNER 인계 구조를 제거하므로 표시 내용을 단일 탭 기준으로 단순화한다.

예:

CDP:0 · tab 934967

또는 실제 UI 구조에 맞는 가장 간결한 표현으로 정리한다.

다음은 제거한다.

generation
handoff 상태
pending token
RETIRED 탭
최근 탭 인계 이력
OWNER_COMMITTED
HANDOFF_READY
OLD_TAB_CLOSED

현재 존재하는 다음 기능은 유지한다.

상태 표시 클릭 전체 진단 복사
텍스트 드래그 선택

_ 축소·복원
□ 상세 패널
상세 패널 수동 새로그침
SIDE 수집계정 목록의 1부터 버튼

9. 문서 정리

ai_read.md 는 현재 기준으로 압축 정리한다.

ai_read_삭제.md 에 다음을 명확히 기록한다.

탭 분리 방식 폐기
새 탭 생성 방식 폐기
OWNER handoff 폐기
generation 폐기
pending token 폐기
RETIRED/OLD_TAB_CLOSED 로직 폐기
YELLOW_TEMP 강제 저장 방식 폐기
WS8771 Python 미디어 재요청 방식 폐기

다시 되살리면 안 되는 이유도 기록한다.

복잡한 OWNER 인계와 이전 탭 종료가 자동실행 중단 원인이 됨
Chrome 세션 다운로드로 3/3 및 18/18 실제 성공 확인
단일 탭 구조가 더 단순하고 안정적인

v283 실제 성공 결과는 ai_read_완료.md 에 테스트 로그 기준으로 기록한다.

3계정 순회 완료
18/18 파일 완료
자동종료 완료
Chrome session download
WS8771 move complete
저장 팝업은 사용자가 수동 Enter

저장 팝업 자동 Enter는 v284 실제 Windows 로그 전까지 [검증대기] 다.

10. 테스트

반드시 test_* 파일 기준으로 검증한다.

필수 테스트:

단일 tab_id 유지
계정 이동 시 새 탭 생성 없음
계정 이동 시 이전 탭 종료 없음
handoff 메시지 호출 없음
generation/token 저장 없음
다른 사용자 Instagram 탭 영향 없음

Chrome downloadId 작업 매칭
MP4/JPG 동시 다운로드 매칭
실제 DownloadItem.filename 전달
YELLOW_TEMP 강제 경로 없음
WS8771 실제 파일 이동
파일 존재·크기 확인
자동/수동 공통 다운로드 함수

저장 자동처리=자동일 때 Enter 처리
저장 자동처리=수동일 때 키 입력 없음
팝업 감지 실패 시 체크실패

기존 완료 기능 회귀:

JS syntax
Python compile
manifest/version
SIDE 1부터 버튼
MAIN 통합 진단 UI
WS8771 최종 파일 이동
JSON/HTML 갱신
MEDIA_SAVED
자동 집계와 자동종료

실제 Windows 테스트 로그가 없는 항목은 [검증대기] 로 둔다.

11. 버전

다음 버전은:

ZIP: v284
manifest version: 2.3.8.2840
화면 버전: v284

로 맞춘다.

잘못된 낮은 버전을 기준으로 되돌리지 않는다.

12. 작업 시작 순서

사용자가 이 프롬프트와 v283 ZIP을 올린 뒤 정확히 **코딩 시작** 이라고 했으므로 바로 다음 순서로 진행한다.

1. 최신 v283 ZIP 해제
2. ai_read.md 확인
3. ai_read_패치.md 확인
4. ai_read_완료.md 확인
5. ai_read_삭제.md 확인
6. test_list.md 확인
7. 실제 source 구조 확인
8. 완료 예정 작업 목록을 대화창에 먼저 출력
9. v284 실제 수정
10. 테스트
11. 문서·버전 반영
12. 최종 ZIP 하나 생성

한 번의 코딩 작업은 위 단일 목표만 처리한다.

Analyze8780 한국시간 표시 등 다른 독립 작업은 포함하지 않는다.

최종 산출물은 ZIP 하나만 제공한다.

최종 답변 순서:

1. 수정 핵심 요약
2. 테스트 결과
3. 주의사항 또는 남은 문제
4. 마지막 줄 다운로드 링크