

엑셀 파일 열기 전에 하는 7가지 구조 점검

XLSX/XLSM 사전감사 실전 가이드 + 무의존성 CLI 키트

구조 신호를 빠르게 수집하고, 사람이 검증할 질문을 만드는 읽기 전용 사전
점검 워크플로

v1.0 · 2026

이 책이 해결하는 문제

누군가 만든 .xlsx 또는 .xlsm 파일을 넘겨받았을 때 바로 열고 수식부터 읽기 시작하면 검토 범위가 쉽게 무너진다. 시트가 몇 개인지, 숨겨진 시트가 있는지, 외부 파일 연결이 남아 있는지, 매크로가 포함됐는지, 계산 설정이 수동인지조차 모른 채 눈에 보이는 표만 따라가게 되기 때문이다.

이 책은 수식이 맞다고 판정하는 책이 아니다. 통합문서의 Open XML 패키지를 읽기 전용으로 조사해 일곱 가지 구조 신호를 모으고, 그 결과를 바탕으로 다음 검토 질문을 정하는 방법을 설명한다. 함께 제공되는 CLI 도구는 Microsoft Excel을 실행하거나 파일을 수정하지 않는다.

이 사전점검은 오류 부재 증명, 회계 감사 의견, 보안 인증 또는 비즈니스 로직 검증을 대신하지 않는다. 결과는 검토 범위를 정하기 위한 인벤토리다.

독자

- 타인이 만든 Excel 파일을 정기적으로 넘겨받는 운영·재무·데이터 실무자
- 터미널에서 Python 명령 한 줄을 실행할 수 있는 분석가
- 파일을 수정하기 전에 구조와 위험 신호를 기록해야 하는 검토자
- .xlsx 내부 형식과 자동 점검 원리를 배우려는 개발자

얻는 것

- 원본 보존부터 결과 기록까지 반복 가능한 사전점검 순서
- 일곱 가지 구조 신호의 의미와 오해하기 쉬운 한계
- JSON 결과를 검토 질문으로 바꾸는 판단 규칙
- 자동 점검 뒤 사람이 반드시 해야 하는 독립 검증 체크리스트

목차

1. 사전점검의 경계 2. 통합문서를 받았을 때의 위협 모델 3. XLSX/XLSM은 ZIP 패키지다 4. 원본 보존과 실행 워크플로 5. 신호 1 - 시트 이름과 숨김 상태 6. 신호 2 - 시트별 수식 수 7. 신호 3 - 캐시된 오류 값 8. 신호 4 - 병합 셀 범위 9. 신호 5 - 외부 링크 구성요소 10. 신호 6 - VBA 매크로 존재 11. 신호 7 - 계산 설정 12. JSON 결과를 읽는 순서 13. strict 모드와 자동화 14. 신호별 대응 플레이북 15. 사람이 하는 독립 재계산 16. 세 가지 실무 사례 17. 팀 표준운영절차 18. 한계와 오판 방지 19. 최종 체크리스트

처음 사용하는 독자는 1장, 4장, 12장, 18장을 먼저 읽고 샘플 결과를 확인한 뒤 나머지 신호를 참고해도 된다.

1. 사전점검의 경계

사전점검(preflight)은 비행 전 점검처럼 본 검토에 들어가기 전에 조사 범위를 확정하는 단계다. 엔진이 앞으로 고장 나지 않는다고 보증하는 일이 아니라, 출발 전에 관찰할 수 있는 상태를 일정한 순서로 기록하는 일에 가깝다.

Excel 파일 검토에는 최소 세 층이 있다.

- 구조 층: 시트, 수식 노드, 병합 범위, 외부 링크, 매크로, 계산 설정처럼 파일 패키지에서 직접 읽을 수 있는 사실
- 계산 층: 수식이 현재 입력값으로 다시 계산됐을 때 어떤 값을 내는지
- 의미 층: 그 수식과 값이 업무 규칙, 계약, 회계 정책 또는 분석 목적에 맞는지

제공 도구가 다루는 범위는 첫 번째 구조 층과, 파일에 이미 저장된 캐시 오류 값 일부다. 수식을 실행하지 않으므로 계산 층을 검증하지 않고, 업무 맥락을 모르므로 의미 층을 판단하지 않는다.

좋은 결과의 정의

좋은 사전점검 결과는 “문제가 없다”가 아니다. 다음 질문에 답할 수 있는 결과다.

- 내가 검토해야 할 시트와 숨겨진 범위는 어디인가?
- 오류 캐시가 있는 셀은 어디에서 시작해 추적해야 하는가?
- 외부 데이터나 코드 실행 가능성을 별도 검토해야 하는가?
- 계산 설정 때문에 화면 값이 최신이 아닐 가능성이 있는가?
- 사람이 독립적으로 다시 계산할 핵심 합계는 무엇인가?

결과가 깨끗해도 논리가 틀릴 수 있다. 결과가 시끄러워도 실제 결함이 아닐 수 있다. 사전점검의 산출물은 판결이 아니라 우선순위다.

2. 통합문서를 받았을 때의 위협 모델

위협 모델은 공격자만을 뜻하지 않는다. 파일이 잘못될 수 있는 현실적인 원인을 먼저 적는 작업이다. Excel 파일에서는 다음 원인이 반복된다.

우발적 변경

행 삽입, 시트 복사, 범위 확장 누락, 수동 계산 설정, 잘못된 붙여넣기처럼 악의가 없는 편집이 결과를 바꾼다. 구조 점검은 이런 변경의 흔적을 직접 증명하지 못하지만 검토할 범위를 좁혀 준다.

숨겨진 의존성

보이는 표가 다른 시트, 다른 통합문서, 이름 정의, 쿼리 또는 매크로에 의존할 수 있다. 외부 링크와 숨김 시트, 매크로 존재는 이런 의존성을 조사해야 한다는 신호다.

오래된 캐시

XLSX는 수식과 함께 마지막 계산 결과를 저장할 수 있다. 생성 도구가 수식을 다시 계산하지 않았거나 계산 모드가 수동이면 화면에서 본 값이 최신 입력과 일치하지 않을 수 있다.

검토 범위 착시

시트 탭 몇 개만 보고 “전체를 봤다”고 생각하기 쉽다. 매우 숨김(veryHidden) 시트나 외부 링크 부품이 있으면 화면에 보이는 범위와 파일 구조가 다르다.

실행 가능 콘텐츠

.xlsm의 VBA는 업무 자동화일 수도 있고 위험한 코드일 수도 있다. 제공 도구는 매크로 존재만 알리고 코드를 실행하거나 안전성을 판정하지 않는다.

사전점검을 시작하기 전에 “누가 만들었는가, 어떤 시스템에서 나왔는가, 마지막으로 언제 정상 계산됐는가, 어떤 결정에 쓰이는가”를 기록하면 구조 신호를 훨씬 정확히 해석할 수 있다.

3. XLSX/XLSM은 ZIP 패키지다

.xlsx와 .xlsm은 단일 바이너리 덩어리가 아니라 여러 XML과 관계 파일을 묶은 ZIP 기반 Open XML 패키지다. 확장자를 .zip으로 바꾸지 않아도 Python의 zipfile 모듈로 내부 항목을 읽을 수 있다.

대표적인 구성요소는 다음과 같다.

- xl/workbook.xml: 시트 목록과 계산 설정
- xl/_rels/workbook.xml.rels: 시트 ID와 실제 XML 부품의 관계
- xl/worksheets/sheetN.xml: 셀, 수식, 병합 범위
- xl/externalLinks/: 다른 통합문서와의 연결 정보
- xl/vbaProject.bin: VBA 프로젝트가 있을 때 포함되는 바이너리

제공 도구는 이 패키지를 읽기 전용으로 열고 XML 네임스페이스를 따라 필요한 항목만 쉰다. Excel 애플리케이션을 자동화하지 않으므로 화면 렌더링, 수식 엔진, Power Query 실행, VBA 실행은 발생하지 않는다.

왜 이 방식이 유용한가

- 파일을 열어 계산이나 링크 갱신을 유발하기 전에 구조를 볼 수 있다.
- Microsoft Excel이 없는 환경에서도 일관된 JSON을 얻는다.
- 외부 Python 패키지를 설치하지 않아도 된다.
- 결과를 CI 로그나 감사 작업표에 보관하기 쉽다.

무엇을 볼 수 없는가

- 수식의 새 계산 결과
- 조건부 서식이 화면에 표현되는 방식
- 차트가 의도대로 보이는지
- VBA 코드가 안전한지
- Power Query가 실제로 어떤 네트워크 요청을 하는지

패키지 구조를 읽는 것과 통합문서를 “이해하는 것”은 다르다. 이 차이를 잊지 않는 것이 가장 중요한 안전장치다.