LatencyCheck

Покроковий опис роботи

Загальний перебіг інсталяції, запуску, перевірки, trace, аналізу, результатів, оновлення та видалення

Цей документ крок за кроком описує, як працює LatencyCheck від інсталяції до щоденного використання та видалення. Опис є загальним і не залежить від конкретної build-версії.

Основний принцип оцінювання: програма не «карає» саму активність системи. Вона перевіряє, чи ця активність впливає на функціональну реакцію, затримку, queue/backlog, знахідки DPC/ISR та практичні симптоми.

1. Загальний огляд роботи

Фаза	Що відбувається
1	Користувач інсталює програму за допомогою setup, який готує права, файли, shortcuts і допоміжні компоненти.
2	Програма запускається як основний GUI і перевіряє, чи є потрібні умови для latency diagnostics.
3	Користувач вибирає перевірку або trace, який потрібно виконати.
4	Програма збирає дані, показує прогрес і створює ETL-файли та/або звіти.
5	Дані аналізуються і перетворюються на зрозумілі результати.
6	Оцінювання класифікує знахідки як OK, активність, warning або alert залежно від впливу на реакцію.
7	Поточні результати відображаються в Results/Summary, а останні зберігаються в History.
8	Користувач може перевірити оновлення, змінити мову або видалити програму.

2. Інсталяція та підготовка

Крок	Опис
2.1	Користувач запускає setup програми.
2.2	Setup перевіряє права administrator/UAC там, де це потрібно.
2.3	Вибирається мова та базові параметри інсталяції.
2.4	Перевіряються вимоги, такі як runtime, інструменти trace і системні права.

2.5	Якщо є залишки попередньої інсталяції, перед інсталяцією виконується очищення.
2.6	До інсталяційної папки копіюються виконувані файли, іконки, допоміжні scripts і потрібні resources.
2.7	Створюються ярлики Start Menu/Desktop, App Paths, записи uninstall і, де потрібно, elevated scheduled task.
2.8	Після завершення програму можна відкрити одразу або через ярлик.

3. Запуск програми

Крок	Опис
3.1	Користувач відкриває LatencyCheck зі Start Menu, Desktop shortcut або pinned taskbar icon.
3.2	Програма перевіряє, чи працює з правильними правами для diagnostics.
3.3	Якщо потрібен elevated запуск, виконується relaunch через передбачений механізм, щоб GUI відкрився правильно.
3.4	Завантажується головне вікно та ініціалізуються вкладки, кнопки, paths і мовні налаштування.
3.5	Програма показує поточний стан: вимоги, доступні інструменти, останні звіти й готові дії.

4. Перевірка вимог і стану системи

Перед початком фактичного trace програма перевіряє, чи система може створити надійні дані. Така перевірка запобігає хибним висновкам через неповне середовище.

Перевірка	Мета
Права	Підтвердити наявність потрібних прав для збору trace і доступу до system data.
WPT / Xperf	З’ясувати, чи доступні інструменти, потрібні для trace і аналізу DPC/ISR.
Runtime	Підтвердити, що потрібний .NET/Desktop runtime існує або може бути встановлений вручну, якщо його немає.
Папки	Забезпечити правильні папки для Traces, Results, History і допоміжних файлів.
Попередні результати	Завантажити або очистити поточні відображення, щоб старий результат не змішувався з новим.

5. Виконання trace / вимірювання

Крок	Опис
5.1	Користувач вибирає дію trace з GUI.
5.2	Програма готує папку збереження та визначає назву ETL/звіту з timestamp.
5.3	Починається збір даних і показується прогрес, щоб користувач бачив, що процес виконується.
5.4	Під час вимірювання не робиться передчасна оцінка лише через те, що є activity.
5.5	Вимірювання зупиняється після завершення часу/дії або коли користувач його припиняє.
5.6	ETL і відповідні reports записуються у визначені папки.
5.7	Програма переходить від збору до аналізу.

6. Аналіз даних

Аналіз перетворює raw trace дані на зрозумілі висновки. Важливо не лише те, який driver був активним, а чи пов'язана активність із реальною затримкою або неправильною реакцією.

Етап	Що перевіряється
DPC / ISR	Часи, drivers/modules, скупчення, піки та можливі точки затримки.
ETL trace	Зв'язок активності з часовою поведінкою та важливими точками.
Drivers / modules	Виявлення modules, які показують систематичне або незвичне навантаження.
Focus modules	Відокремлення найважливіших знахідок, щоб користувач не загубився в несуттєвій інформації.
Висновок	Перетворення технічних вимірювань на практичну ознаку OK, увага або проблема.

7. Логіка оцінювання

Програма використовує функціональне оцінювання. Це означає, що індикація базується не лише на тому, що в системі щось відбувається, а на тому, чи ця активність створює проблему реакції.

Latency Check. Пороги вимірювання не змінюються; уточнюється лише текст, щоб тимчасовий пік не плутався з активним навантаженням.

Стан	Значення	Де відображається
OK	Чистий вимір / операційна затримка не підтверджена.	Перевірка, Порівняння, Рекомендації та звіти.
OK - АКТИВНА ДІЯЛЬНІСТЬ	Активність є, але без задокументованого впливу на функціональну реакцію.	Перевірка і Рекомендації.

ПОПЕРЕДЖЕННЯ - ПІК	Поточне значення повернулося до OK, але найвище/пікове значення перевищило межу під час вимірювання.	Стан у Перевірці, старий/новий стан у Порівнянні та причина рекомендації.
ПОПЕРЕДЖЕННЯ - КОРЕЛЯЦІЯ	Попередження виникає з комбінації показників, таких як черга CPU, DPC/IRQ, затримка диска або мережевий backlog.	Рекомендації та підсумкові висновки.
АКТИВНЕ ПОПЕРЕДЖЕННЯ	Поточне значення все ще вище межі. Навантаження є зараз.	Перевірка, Порівняння, Рекомендації та звіти.
АКТИВНА ТРИВОГА	Сильне або підтверджене навантаження на відгук у цьому часовому вікні.	Перевірка, Порівняння, Рекомендації та звіти.

Вплив на Порівняння і Рекомендації

- Вкладка Рекомендації використовує ту саму класифікацію в загальному стані, значенні, пропозиції, кореляції з Перевіркою та причині.
- Вкладка Порівняння використовує її для стану старого і нового вимірювання. Числові поля різниці залишаються числовими.
- Зміна стосується лише тексту інтерпретації. Вона не змінює пороги, кольори, лічильники, trace-логіку, Worker або потік встановлення.

8. Відображення результатів

Область	Роль
Summary	Показує основні висновки поточного вимірювання у зведеній формі.
Results	Зберігає поточний результат останнього активного аналізу.
History	Зберігає недавні вимірювання, щоб можна було порівняти їх із попередніми результатами.
Reports	Відкриває або показує створені файли звітів.
Recommendations	Подає практичні вказівки, коли є знахідка, що потребує дії.
Drivers / IRQ / ISR	Показує спеціалізовані знахідки для drivers, interrupts і modules.

9. Зміна мови та оновлення UI

Крок	Опис
9.1	Користувач змінює мову в UI.
9.2	Програма застосовує переклади до основних вікон і вкладок.
9.3	Вікна або допоміжні відображення зі старим текстом закриваються/очищуються, щоб не залишалися змішані мови.
9.4	Користувач знову відкриває окремі результати/інструменти новою мовою.

10. Оновлення програми

Крок	Опис
10.1	Користувач вибирає перевірку нової версії.

10.2	Програма перевіряє джерело оновлення і порівнює локальну версію з доступною.
10.3	Якщо є новіша версія, вона повідомляє користувача і показує можливість завантаження/інсталяції.
10.4	Завантаження/оновлення відбувається контрольованим потоком, і користувач бачить прогрес там, де це потрібно.
10.5	Після оновлення нова версія має правильно відображатися в UI, About, update check та installer metadata.

11. Видалення

Крок	Опис
11.1	Користувач запускає видалення зі Start Menu, Windows Apps/Programs або відповідного uninstall entry.
11.2	Де потрібно, деінсталятор запитує права.
11.3	Закриває програму, якщо вона працює.
11.4	Видаляє scheduled task, shortcuts, App Paths і registry uninstall entries.
11.5	Безпечно очищає інсталяційну папку.
11.6	Там, де передбачено, може зберегти або перенести results/history замість їх негайної втрати.

12. Повний покроковий перебіг від користувача до результату

#	Перебіг
1	Інсталяція LatencyCheck.
2	Відкриття програми через shortcut або Start Menu.
3	Перевірка вимог і прав.
4	Вибір мови та підтвердження UI.
5	Вибір типу перевірки/trace.
6	Запуск вимірювання з індикацією прогресу.
7	Збір ETL/DPC/ISR даних.
8	Аналіз drivers/modules/часів.
9	Створення звітів і зведення.

10	Показ результату з функціональним оцінюванням.
11	Відкриття окремих reports або recommendations.
12	Збереження поточного результату й оновлення history.
13	Необов’язкова перевірка оновлення програми.
14	Необов’язкове видалення, коли програма більше не потрібна.

13. Центральна ідея

LatencyCheck працює як провідник діагностики реакції: організовує збір даних, приховує складність trace-інструментів, показує важливі знахідки та уникає позначення звичайної нормальної активності системи як проблеми.

Мета полягає не просто в тому, щоб знайти, який driver рухався найбільше, а в тому, щоб показати, чи є реальна затримка, яка впливає на використання комп’ютера.

Додатково повний список драйверів DPC/ISR показує drivers/modules із завантаженого DPC/ISR trace: кількість ISR/DPC, час CPU ISR/DPC, загальний час CPU та компанію/продукт, щоб було видно, які драйвери найбільше брали участь у вимірюванні, без зміни логіки оцінювання.