

LatencyCheck - فارسی

LatencyCheck

شرح مرحله‌ای عملکرد

جریان کلی نصب، اجرا، بررسی، trace، تحلیل، نتایج، به‌روزرسانی و حذف نصب

هدف این سند این است که مرحله‌به‌مرحله توضیح دهد LatencyCheck از نصب تا استفاده روزانه و حذف نصب چگونه کار می‌کند. این توضیح عمومی است و به build مشخصی وابسته نیست.

اصل پایه ارزیابی: برنامه فعالیت سیستم را «مجازات» نمی‌کند. بررسی می‌کند که آیا آن فعالیت بر پاسخ عملکردی، latency، queue/backlog، یافته‌های DPC/ISR و علائم عملی اثر دارد یا نه.

1. نمای کلی عملکرد

مرحله	آنچه رخ می‌دهد
1	کاربر برنامه را با setup نصب می‌کند که مجوزها، فایل‌ها، shortcuts و اجزای کمکی را آماده می‌کند.
2	برنامه به عنوان GUI اصلی اجرا می‌شود و بررسی می‌کند که نیازمندی‌های لازم برای latency diagnostics وجود داشته باشند.
3	کاربر نوع بررسی/trace مورد نظر را انتخاب می‌کند.
4	برنامه داده‌ها را جمع‌آوری می‌کند، پیشرفت را نشان می‌دهد و ETL یا گزارش‌ها را ایجاد می‌کند.
5	داده‌ها تحلیل می‌شوند و به نتایج قابل فهم تبدیل می‌شوند.
6	ارزیابی، یافته‌ها را بسته به اثر بر پاسخ، به OK، activity، warning یا alert طبقه‌بندی می‌کند.
7	نتایج فعلی در Results/Summary نمایش داده می‌شود و نتایج اخیر در History نگه داشته می‌شود.
8	کاربر می‌تواند update را بررسی کند، زبان را تغییر دهد یا برنامه را حذف نصب کند.

2. نصب و آماده‌سازی

مرحله	توضیح
2.1	کاربر setup برنامه را اجرا می‌کند.
2.2	setup مجوزهای administrator/UAC را در موارد لازم بررسی می‌کند.
2.3	زبان و گزینه‌های پایه نصب انتخاب می‌شوند.
2.4	نیازمندی‌هایی مانند runtime، ابزارهای trace و مجوزهای سیستم بررسی می‌شوند.
2.5	اگر باقی‌مانده‌های نصب قبلی وجود داشته باشد، پیش از نصب پاک‌سازی انجام می‌شود.
2.6	فایل‌های اجرایی، آیکون‌ها، scripts کمکی و resources لازم در پوشه نصب کپی می‌شوند.
2.7	Start Menu/Desktop shortcuts، App Paths، uninstall entries و در صورت نیاز elevated scheduled task ساخته می‌شوند.
2.8	پس از پایان نصب، برنامه می‌تواند فوراً یا از shortcut باز شود.

3. اجرای برنامه

مرحله	توضیح
3.1	کاربر LatencyCheck را از Desktop shortcut، Start Menu، یا pinned taskbar icon باز می‌کند.
3.2	برنامه بررسی می‌کند که برای diagnostics با مجوزهای درست اجرا شده باشد.
3.3	اگر اجرای elevated لازم باشد، با مکانیزم پیش‌بینی شده relaunch انجام می‌شود تا GUI درست باز شود.
3.4	پنجره اصلی بارگذاری می‌شود و تب‌ها، دکمه‌ها، paths و تنظیمات زبان آماده می‌شوند.
3.5	برنامه وضعیت فعلی را نشان می‌دهد: نیازمندی‌ها، ابزارهای موجود، آخرین گزارش‌ها و اقدامات آماده.

4. بررسی نیازمندی‌ها و وضعیت سیستم

پیش از شروع trace واقعی، برنامه بررسی می‌کند که سیستم بتواند داده‌های قابل اعتماد تولید کند. این بررسی از نتیجه‌گیری غلط ناشی از محیط ناقص جلوگیری می‌کند.

بررسی	هدف
مجوزها	تأیید اینکه مجوزهای لازم برای جمع‌آوری trace و دسترسی به system data وجود دارد.
WPT / Xperf	بررسی اینکه ابزارهای لازم برای trace و تحلیل DPC/ISR در دسترس هستند.
Runtime	تأیید اینکه NET/Desktop runtime لازم وجود دارد یا در صورت نبود، امکان نصب دستی دارد.
پوشه‌ها	وجود پوشه‌های درست برای Traces, Results, History و فایل‌های کمکی.
نتایج قبلی	بارگذاری یا پاک‌سازی نمایش‌های فعلی تا نتیجه قدیمی با نتیجه جدید اشتباه نشود.

5. اجرای trace / اندازه‌گیری

مرحله	توضیح
5.1	کاربر عمل trace را از GUI انتخاب می‌کند.
5.2	برنامه پوشه ذخیره را آماده می‌کند و نام ETL/گزارش را با timestamp تعیین می‌کند.
5.3	جمع‌آوری داده شروع می‌شود و پیشرفت نمایش داده می‌شود تا کاربر بداند فرآیند در حال اجراست.
5.4	در طول اندازه‌گیری، فقط به دلیل وجود activity ارزیابی زود هنگام انجام نمی‌شود.
5.5	اندازه‌گیری پس از پایان زمان/عمل یا با توقف کاربر متوقف می‌شود.
5.6	ETL و گزارش‌های مربوط در پوشه‌های پیش‌بینی شده نوشته می‌شوند.
5.7	برنامه از جمع‌آوری داده به تحلیل منتقل می‌شود.

6. تحلیل داده‌ها

تحلیل، داده‌های خام trace را به نتیجه‌های قابل فهم تبدیل می‌کند. نکته مهم فقط فعال بودن یک driver نیست، بلکه این است که آیا فعالیت با latency واقعی یا پاسخ غیرعملکردی ارتباط دارد یا نه.

مرحله تحلیل	چه چیزی بررسی می‌شود
DPC / ISR	زمان‌ها، drivers/modules، تجمع‌ها، peak ها و نقاط احتمالی latency.

مرحله تحلیل	چه چیزی بررسی می شود
ETL trace	ارتباط فعالیت با رفتار زمانی و نقاط مهم.
Drivers / modules	شناسایی modules که بار سیستماتیک یا غیرعادی نشان می دهند.
Focus modules	جداسازی مهم ترین یافته ها تا کاربر در اطلاعات نامربوط گم نشود.
نتیجه	تبدیل اندازه گیری های فنی به نشانه عملی OK، توجه یا مشکل.

7. منطق ارزیابی

برنامه از ارزیابی عملکردی استفاده می کند. یعنی نشانه فقط بر اساس وجود حرکت در سیستم تعیین نمی شود، بلکه بر اساس این است که آیا آن حرکت مشکل پاسخ ایجاد می کند یا نه.

نشانه	معنی	محل نمایش
OK	اندازه گیری پاک / تأخیر عملکردی تأیید نمی شود.	Check, Comparison, Recommendations و .reports
OK - فعالیت فعال	فعالیت وجود دارد، اما اثر مستند بر پاسخ عملکردی ندارد.	Check و Recommendations.
هشدار - PEAK	مقدار فعلی به OK برگشته، اما بالاترین/حداکثر مقدار در طول اندازه گیری از حد عبور کرده است.	وضعیت در Check، وضعیت قدیم/جدید در Comparison و علت Recommendations.
هشدار - ارتباط	هشدار از ترکیب شاخص هایی مانند CPU queue، DPC/IRQ، disk latency یا network backlog به دست می آید.	Recommendations و خلاصه ها.
هشدار فعال	مقدار فعلی هنوز بالاتر از حد است؛ فشار اکنون وجود دارد.	Check, Comparison, Recommendations و .reports
ALERT فعال	فشار شدید یا تأیید شده پاسخ در پنجره زمانی وجود دارد.	Check, IRQ/ISR, Comparison و Recommendations .reports

اثر در Recommendations و Comparison

- تب Recommendations از همان طبقه بندی در وضعیت کلی، معنی، پیشنهاد، ارتباط با Check و علت استفاده می کند.
- تب Comparison از آن برای وضعیت اندازه گیری قدیمی و جدید استفاده می کند. فیلدهای عددی اختلاف، عددی باقی می ماند.
- این تغییر فقط تفسیر متن را تحت تأثیر قرار می دهد. حدود، رنگها، counters، منطق Worker logic، trace یا flow نصب تغییر نمی کند.

8. نمایش نتایج

مرحله	توضیح
8.1	پس از تحلیل، نتیجه اصلی در تب Results/Summary نمایش داده می شود.
8.2	رنگها و متن وضعیت برای نشان دادن اثر واقعی بر پاسخ استفاده می شوند.
8.3	نتیجه نهایی از ترکیب latency، DPC/ISR، queue/backlog، disk/network و علائم عملی می آید.
8.4	اگر activity وجود داشته باشد اما اثر عملکردی تأیید نشود، نتیجه نباید به اشتباه مشکل اعلام شود.
8.5	گزارش های جزئی برای بررسی عمیق تر در دسترس می مانند.
8.6	کاربر می تواند از نتیجه به Recommendations، Compare یا reports برود.

9. History و مقایسه

مرحله	توضیح
9.1	نتایج اخیر در History ذخیره می‌شوند تا کاربر بتواند روند را دنبال کند.
9.2	Comparison اندازه‌گیری فعلی را با اندازه‌گیری قدیمی‌تر مقایسه می‌کند.
9.3	اختلاف‌های عددی به صورت عددی باقی می‌مانند و متن وضعیت برای فهم عملی استفاده می‌شود.
9.4	مقایسه driver/trace فقط وقتی معتبر است که trace سازگار در هر دو طرف وجود داشته باشد.
9.5	هدف History این است که بعد از driver update, Windows update, BIOS بتوان اثر تغییر را دید.

10. به‌روزرسانی و دانلود

مرحله	توضیح
10.1	کاربر می‌تواند از داخل برنامه یا سایت عمومی، نسخه جدید را بررسی کند.
10.2	metadata نسخه، SHA256, release notes, و download link باید با build جدید هماهنگ باشند.
10.3	اگر نسخه جدید وجود داشته باشد، جریان update به کاربر نمایش داده می‌شود.
10.4	دانلود/به‌روزرسانی با جریان کنترل‌شده انجام می‌شود و کاربر در صورت نیاز پیشرفت را می‌بیند.
10.5	پس از به‌روزرسانی، نسخه جدید باید در UI, About, update check و installer metadata درست دیده شود.

11. حذف نصب

مرحله	توضیح
11.1	کاربر حذف نصب را از Windows Apps/Programs, Start Menu, یا uninstall entry مربوط شروع می‌کند.
11.2	uninstaller در صورت نیاز مجوز می‌خواهد.
11.3	اگر برنامه در حال اجرا باشد، آن را می‌بندد.
11.4	registry uninstall entries و scheduled task, shortcuts, App Paths را حذف می‌کند.
11.5	پوشه نصب را به شکل ایمن پاک‌سازی می‌کند.
11.6	در جایی که پیش‌بینی شده، می‌تواند results/history را حفظ یا منتقل کند تا فوراً از بین نرود.

12. جریان کامل مرحله‌به‌مرحله از کاربر تا نتیجه

#	جریان
1	نصب LatencyCheck.
2	باز کردن برنامه از shortcut یا Start Menu.
3	بررسی نیازمندی‌ها و مجوزها.
4	انتخاب زبان و تأیید UI.
5	انتخاب نوع بررسی/trace.

#	جریان
6	شروع اندازه‌گیری با نمایش پیشرفت.
7	جمع‌آوری داده‌های ETL/DPC/ISR.
8	تحلیل drivers/modules/timings.
9	ایجاد گزارش‌ها و خلاصه.
10	نمایش نتیجه با ارزیابی عملکردی.
11	باز کردن reports یا recommendations جزئی.
12	ذخیره نتیجه فعلی و به‌روزرسانی history.
13	بررسی اختیاری update برنامه.
14	حذف نصب اختیاری وقتی دیگر لازم نیست.

13. ایده مرکزی

LatencyCheck مانند راهنمای تشخیص پاسخ عمل می‌کند: جمع‌آوری داده را سازمان‌دهی می‌کند، پیچیدگی ETW/xperf را برای کاربر پنهان می‌کند و نتیجه را به زبان عملی تبدیل می‌کند: آیا سیستم واقعاً مشکل پاسخ دارد یا فقط فعالیت عادی انجام می‌دهد.

نسخه سایت: v256