

TEAMSTUDIO® PROFILER

LotusScriptパフォーマンス・モニタ

- ④ LotusScriptパフォーマンスの問題をライン単位で把握
- ④ 試行錯誤のトラブルシューティングの非効率性を除去
- ④ アプリケーション障害が本番環境に影響するリスクを低減
- ④ 他のアプリケーションにおける問題のあるコードの再利用を防止
- ④ コード内のタイムスタンプ独自コーディング不要



アプリケーションのパフォーマンスはタイミングがすべてです。パフォーマンスは速ければ速いほど良いのですが、アプリケーションに障害が発生したりエージェントが予想通り実行されていない時に、問題を簡単に検出することはできません。Teamstudio Profilerはスクリプトの一行ごとにどのように実行されたかを一目で確認でき効果的にボトルネックをピンポイントで把握できます。

ストップウォッチは不要ストップウォッチは不要

管理者や開発者にとって、次のようなシナリオがよく発生します。簡単なアクションに時間がかかっている。エンドユーザーからのアプリケーションパフォーマンスに関するクレーム。スケジュールエージェントの実行が遅い、あるいは実行されていない。あるいは、複数のテクノロジーを組み合わせで実行しているアプリケーション内の原因として指摘されている。

このような場合どうすればいいでしょうか？問題を正確に突き止める唯一のツブを実行する時間を調べることです。しかしスクリプト・ライブラリは膨大でわかりません。さらに、独自のパフォーマンス測定用コードを追加することでアプ違いが生じるリスクがあります。そこでもっとよい方法をご紹介します。

LotusScriptパフォーマンスを検証する優れた方法

TEAMSTUDIO PROFILERはLOTUSSCRIPTのコードの複雑さに関

わらずスクリプトの実行時のパフォーマンスを正確に検証できるツールです。Profilerを使用することでアプリケーションのボトルネックを簡単に発見することができます。シンプルな棒グラフでパフォーマンスデータを表示しているのでスクリプトの遅延原因をすばやく検出することができます。もうスクリプトにTime ステートメントを組み込みながら実行時間を計測したり、ストップウォッチを使って速度を計測する必要はありません。Profilerの包括的なレポート機能を使用しながらコードをテストできるので今後は欠陥のあるスクリプトに悩むことはありません。

開放	コール数	時間+子関	時間	総実行対上	平均時間	時間+子関	要求
INITIALIZE	1	0.007	0.007	96.05%	0.007	0.007	ニュースレタ
TERMINATE	1	0.000	0.000	3.95%	0.000	0.000	ニュースレタ

#	ソース	#	%	時間
1	Sub Initialize			
2	On Error Goto ErrorCleanup			
3				
4	Set s = New NotesSession	1	0.50%	0.000
5	Set db = s.CurrentDatabase	1	0.40%	0.000
6	Set view = db.GetView("Profiles")	1	0.34%	0.000
7	Set profile = view.GetFirstDocument	1	96.71%	0.007
8	If profile Is Nothing Then Exit Sub	1	0.65%	0.000
9	Failed = False			
10				
11				
12	FormName = profile Form			
13	If FormName(0) <> "Interest Profile" Then Set profile = view.GetNextDocumentL...			
14				
15	Do Until profile Is Nothing			
16	FormName = profile Form			

PROFILERの結果表示

Profilerが開発やテストのフェーズで使用されるようになっていけば、パフォーマンス低下の原因となるスクリプトを探してひとつひとつレビューする必要がなくなると同時に、本番環境で発生する前に問題を取り除き堅牢かつパフォーマンスの良いアプリケーションを常に作成できる可能性が大幅に高まります。