ENTERPRISE ARCHITECT

Série de Guides d'Utilisateur

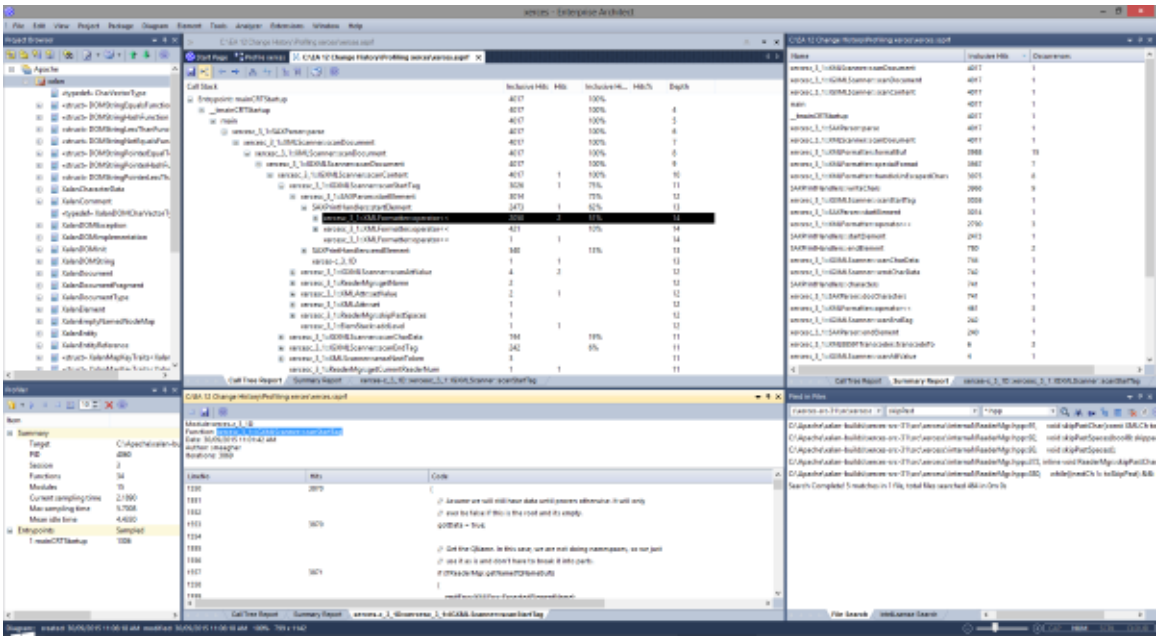
Profilage

Author: Sparx Systems
Date: 7/11/2024
Version: 17.0

CRÉÉ AVEC ENTERPRISE ARCHITECT

Table des Matières

Profilage

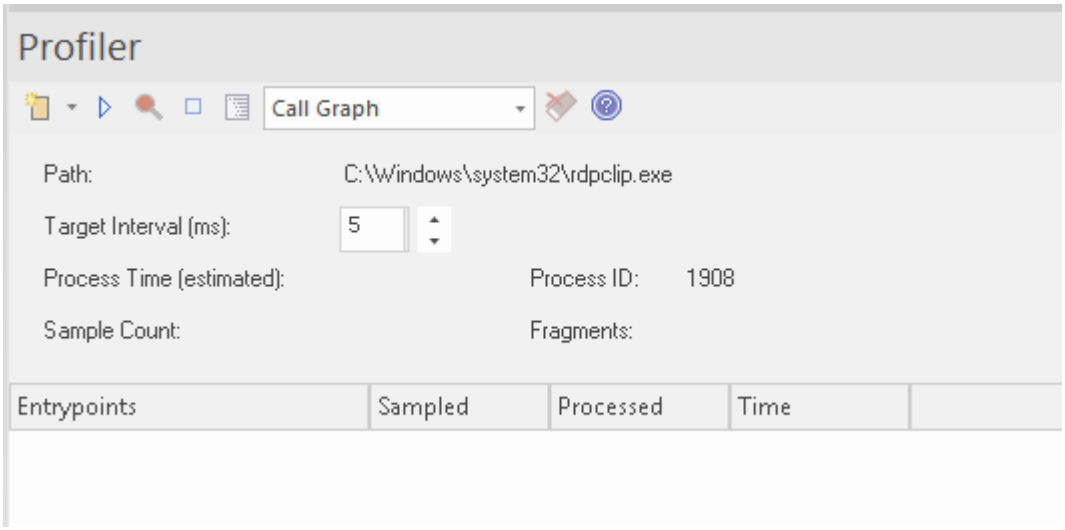


Au cours de la vie des applications logicielles, il n'est pas rare d'examiner des tâches d'application qui s'exécutent plus lentement que prévu. Vous pouvez également simplement vouloir savoir ce qui se passe lorsque vous *appuyez sur ce bouton* ! Vous pouvez résoudre ce problème assez rapidement dans Enterprise Architect en utilisant son Profiler. Les résultats peuvent généralement être produits en quelques secondes et vous pourrez rapidement voir les actions qui consomment l'application et les fonctions impliquées. Dans l' Analyseur d'Exécution , la fonctionnalité utilise deux stratégies distinctes : *l'échantillonnage de processus* et *le hooking de processus.* Dans l'une, des échantillons sont prélevés à intervalles réguliers pour identifier motifs gourmands en ressources CPU, tandis que dans l'autre, le processus est accroché pour enregistrer les demandes faites sur la mémoire. Les données sont analysées pour produire un Graphique d'Appel pondéré. Les comportements sont généralement identifiables comme des nœuds racines (points d'entrée) dans le graphique, ou des branches proches de ces points. Tous les rapports peuvent être examinés à la demande. Ils peuvent être enregistrés dans un fichier au sein du modèle, à la fois comme éléments d'artefact et comme messages Bibliothèque d'Équipe .

Accéder

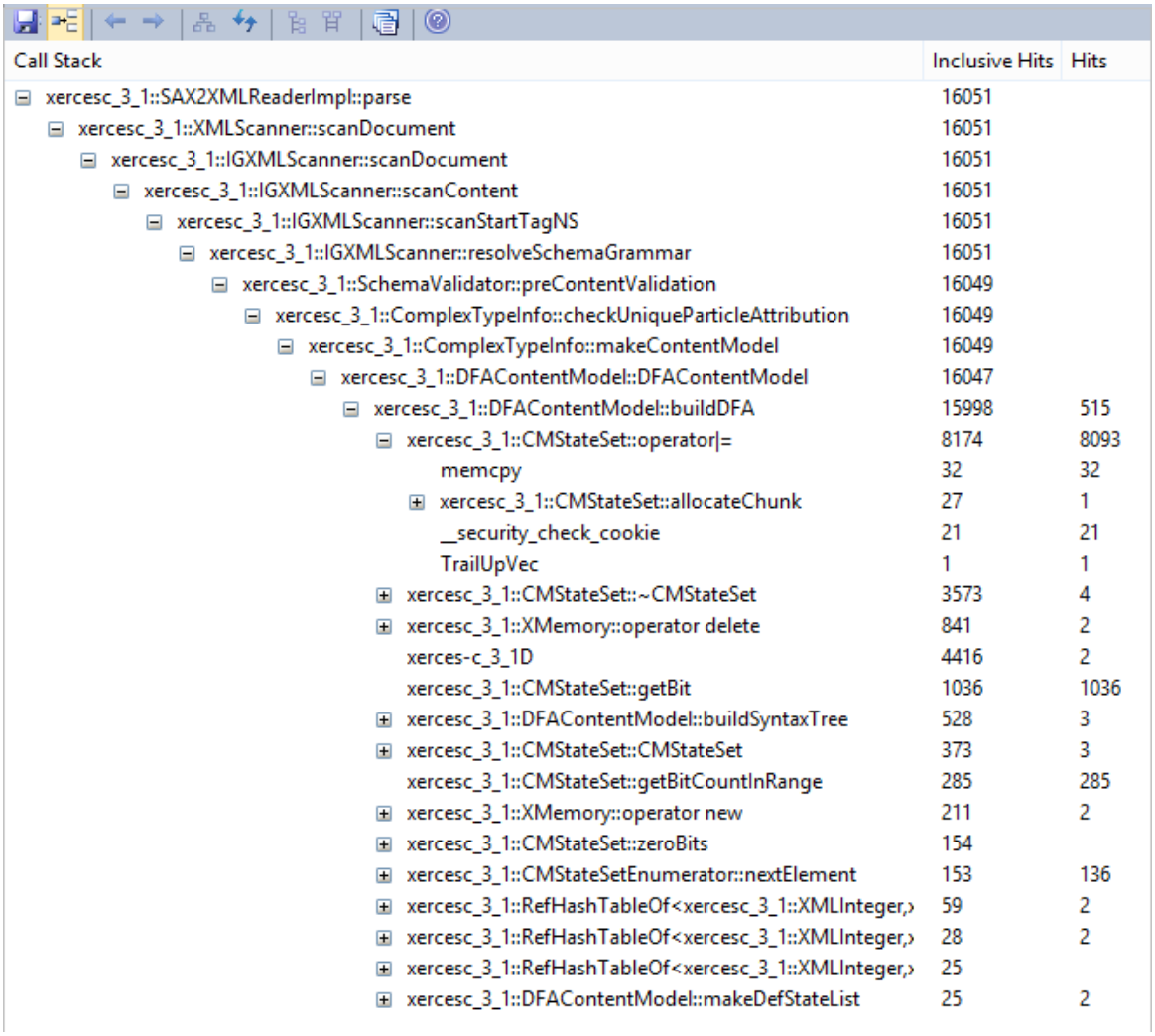
Ruban	Exécuter > Outils > Profiler
Autre	Barre d'outils Analyseur d'Exécution : Analyseur Windows | Profileur

Échantillonnage d'appel



Le profileur est contrôlé à l'aide des boutons de sa barre d'outils. Ici, vous pouvez attacher le profileur à un processus existant (ou à une JVM) ou lancer l'application pour le script Analyzer actif. La fenêtre du profileur affiche les détails du processus cible au fur et à mesure de son profilage. Ces détails fournissent des informations en retour, vous permettant de voir le nombre d'échantillons prélevés. Vous disposez également d'options pour suspendre et reprendre la capture, effacer les données capturées et générer des rapports. Vous pouvez accéder à la fonctionnalité de création de rapports en mettant la capture en pause - la fonctionnalité de création de rapports est désactivée pendant que la capture de données est en cours.

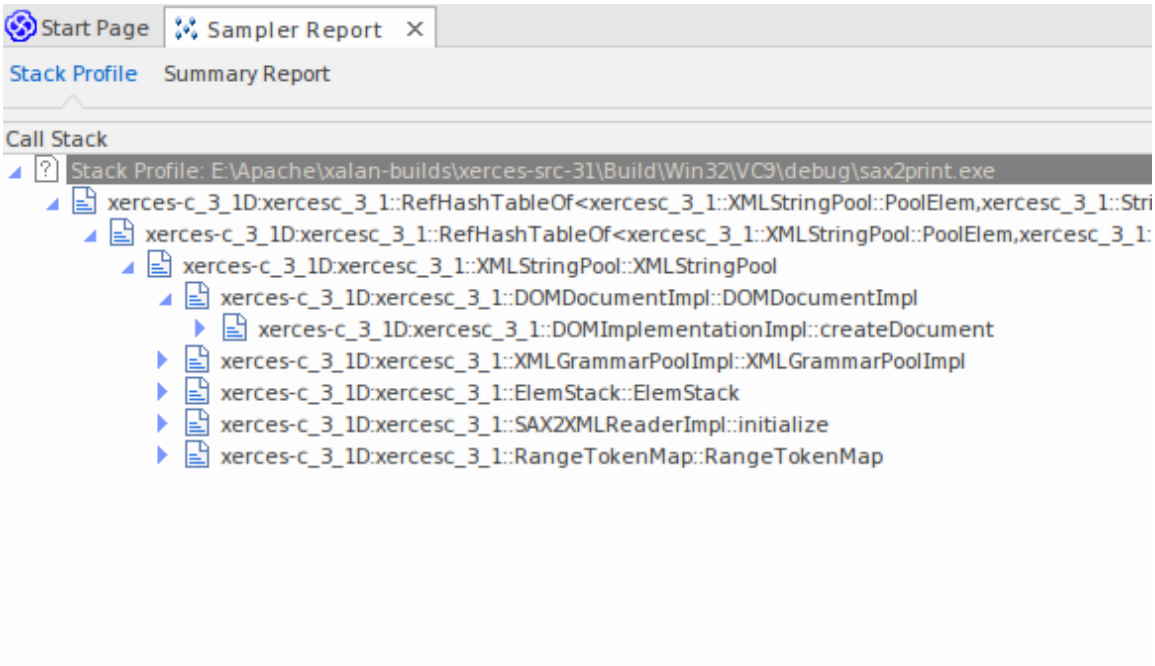
Graphique d'Appel pondéré



Call Stack	Inclusive Hits	Hits
xercesc_3_1::SAX2XMLReaderImpl::parse	16051	
xercesc_3_1::XMLScanner::scanDocument	16051	
xercesc_3_1::IGXMLScanner::scanDocument	16051	
xercesc_3_1::IGXMLScanner::scanContent	16051	
xercesc_3_1::IGXMLScanner::scanStartTagNS	16051	
xercesc_3_1::IGXMLScanner::resolveSchemaGrammar	16051	
xercesc_3_1::SchemaValidator::preContentValidation	16049	
xercesc_3_1::ComplexTypeInfo::checkUniqueParticleAttribution	16049	
xercesc_3_1::ComplexTypeInfo::makeContentModel	16049	
xercesc_3_1::DFAContentModel::DFAContentModel	16047	
xercesc_3_1::DFAContentModel::buildDFA	15998	515
xercesc_3_1::CMStateSet::operator|=	8174	8093
memcpy	32	32
xercesc_3_1::CMStateSet::allocateChunk	27	1
__security_check_cookie	21	21
TrailUpVec	1	1
xercesc_3_1::CMStateSet::~CMStateSet	3573	4
xercesc_3_1::XMemory::operator delete	841	2
xerces-c_3_1D	4416	2
xercesc_3_1::CMStateSet::getBit	1036	1036
xercesc_3_1::DFAContentModel::buildSyntaxTree	528	3
xercesc_3_1::CMStateSet::CMStateSet	373	3
xercesc_3_1::CMStateSet::getBitCountInRange	285	285
xercesc_3_1::XMemory::operator new	211	2
xercesc_3_1::CMStateSet::zeroBits	154	
xercesc_3_1::CMStateSetEnumerator::nextElement	153	136
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	59	2
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	28	2
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	25	
xercesc_3_1::DFAContentModel::makeDefStateList	25	2

Ce rapport détaillé présente l'ensemble unique de piles d'appels/comportements sous forme Graphique d'Appel pondéré. Le poids de chaque branche est représenté par un nombre de hits, qui correspond au nombre total de hits de cette branche plus tous les branches à partir de ce point. En suivant la piste de hits, vous pouvez rapidement identifier les zones de code qui ont le plus occupé le programme pendant la période de capture.

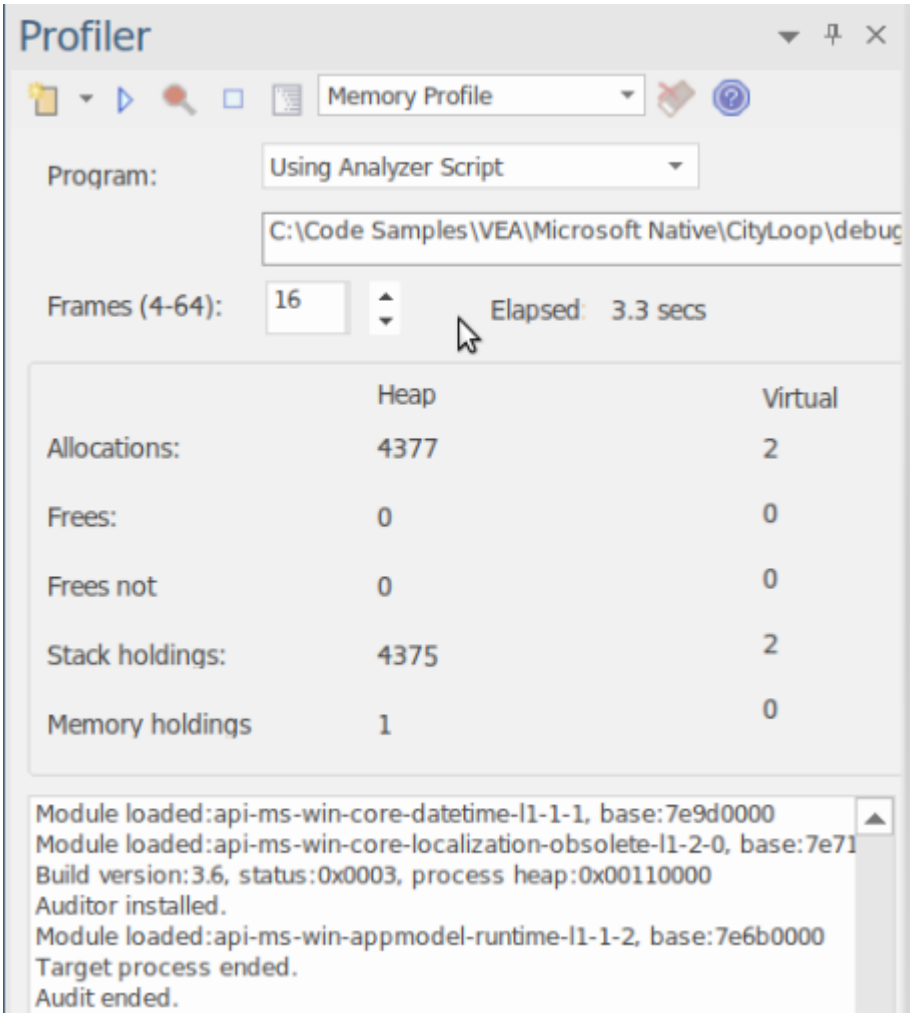
Profil de Pile



Les profils de pile permettent de découvrir les différentes manières (piles) et le nombre de façons dont une fonction particulière est invoquée pendant l'exécution du programme. Contrairement aux autres modes de profilage, ce profil est activé via l'utilisation d'un point de profil, qui est un type spécial de marqueur de point d'arrêt. Le marqueur est défini dans le code source comme tout autre point d'arrêt. Lorsque le point d'arrêt est rencontré par le programme, la pile est capturée. Lorsque vous produisez ultérieurement le rapport, les piles sont analysées et un graphique d'appels pondéré est produit. Le graphique montre les piles uniques impliquées dans cette fonction pendant la période d'exécution du profileur. La colonne « Hit Count » indique le nombre de fois que la même pile s'est produite.

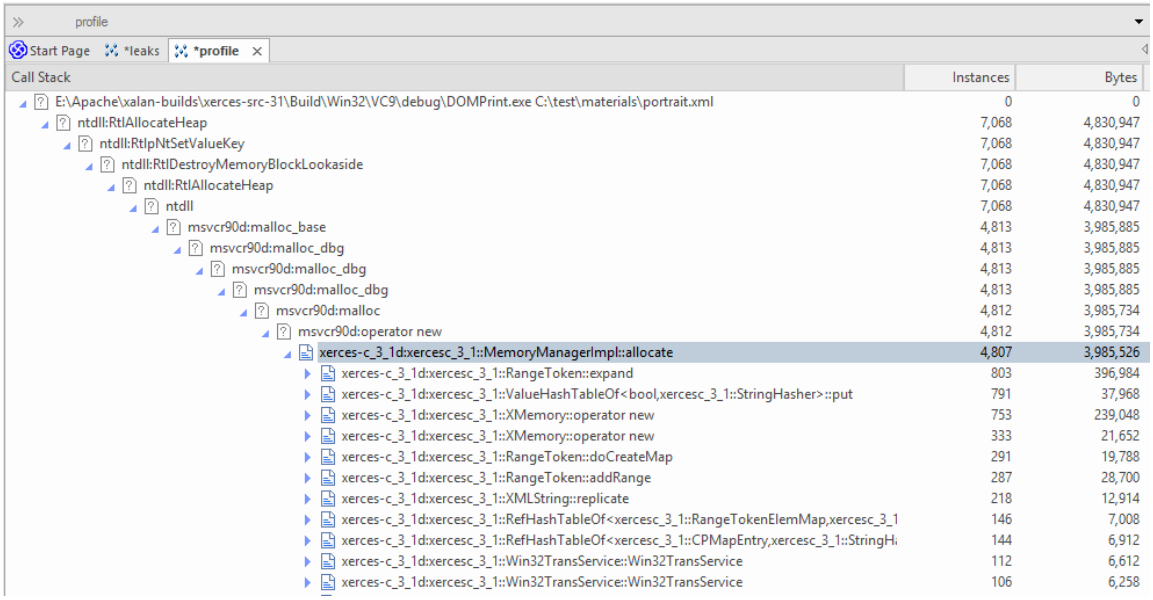
```
106
107 template <class TVal, class THasher>
108 void RefHashTableOf<TVal, THasher>::initialize(const XMLSize_t modulus)
109 {
110     if (modulus == 0)
111         ThrowXMLwithMemMgr(IllegalArgumentException, XMLExcepts::HshTbl_ZeroMo
112
113     // Allocate the bucket list and zero them
114     fBucketList = (RefHashTableBucketElem<TVal>**) fMemoryManager->allocate
115     (
116         fHashModulus * sizeof(RefHashTableBucketElem<TVal>*)
117     );
118     for (XMLSize_t index = 0; index < fHashModulus; index++)
119         fBucketList[index] = 0;
120 }
121
```

Profils de Mémoire



Le profil de mémoire suit les allocations, en ignorant le moment où la mémoire est libérée. Il utilise ces informations pour évaluer les demandes de mémoire du code en cours d'exécution, non pas en termes de quantité de mémoire, mais de fréquence des demandes. Le chiffre *Allocations* correspond au nombre total d'allocations de mémoire demandées. Le nombre de *traces de pile* correspond au nombre de traces de pile prises à ces moments-là, et le chiffre *Heap Holding* correspond à la quantité totale de mémoire obtenue par ces appels. Note que le profilage peut être activé et désactivé à la demande. Il n'est pas non plus nécessaire de reconstruire votre programme pour le faire fonctionner car aucun lien n'est impliqué.

Graphique de Mémoire



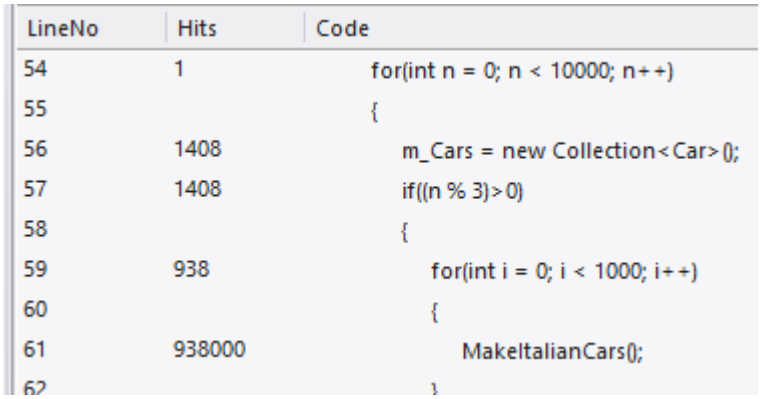
Cet exemple est un rapport produit à partir du profilage d'un programme de démonstration du projet Xerces à partir d'Apache. Le programme parcourt le Modèle Object de document (DOM) pour un fichier XML fourni.

Rapport Sommaire de Fonction

Name	Inclusive Hits
profiler/Example.Run	156
profiler/Example.main	156
java/io/FileOutputStream.write	154
java/io/PrintStream.println	154
profiler/Example.Print	154
profiler/Example.MakeItalianCars	2
profiler/Example.NewCar	2

Ce rapport récapitulatif répertorie les fonctions et uniquement celles exécutées pendant la période d'échantillonnage. Les fonctions sont répertoriées par nombre total d'appels, une fonction qui apparaît deux fois dans des piles d'appels distinctes apparaissant avant une fonction qui n'apparaît qu'une seule fois.

Rapport de Ligne de Fonction



LineNo	Hits	Code
54	1	for(int n = 0; n < 10000; n++)
55		{
56	1408	m_Cars = new Collection<Car>();
57	1408	if((n % 3)>0)
58		{
59	938	for(int i = 0; i < 1000; i++)
60		{
61	938000	MakeItalianCars();
62		}

Ce rapport détaillé montre le code source d'une fonction ligne par ligne, affichant à côté le nombre total de fois que chacune a été exécutée. Nous avons découvert du code à l'aide de ce rapport, qui exposait des instructions case dans du code qui ne semblaient jamais être exécutées.

Support

Le profileur est pris en charge pour les programmes écrits en C, C++, Visual Basic, Java et les langages Microsoft .NET . Le profilage de la mémoire est actuellement disponible pour les programmes C et C++ natifs.

Notes

- Le profileur est disponible dans l'édition Professional Enterprise Architect et au-dessus
- Le Profiler peut également être utilisé sous WINE (Linux et Mac) pour le profilage d'applications Windows standard déployées dans un environnement WINE

Exigences du système

À l'aide du Profiler, vous pouvez analyser les applications créées pour ces plateformes :

- MicrosoftTM Native (C++, C, Visual Basic)
- Microsoft .NET (prenant en charge un mélange de code géré et non géré)
- Java

Applications natives Microsoft

Pour les applications C, C++ ou Visual Basic, le profileur nécessite que les applications soient compilées avec le compilateur Microsoft ™ Native et que pour chaque application ou module concerné, un fichier PDB soit disponible. Le profileur peut échantillonner les configurations de débogage et de publication d'une application, à condition que le fichier PDB de chaque exécutable existe et soit à jour.

Applications Microsoft .NET

Pour les applications Microsoft .NET , le Profiler nécessite que le framework Microsoft .NET approprié soit installé et que pour chaque application ou module à analyser, un fichier PDB soit disponible.

Java

Pour Java, le profileur nécessite que le JDK approprié d'Oracle soit installé.

Les classes d'intérêt doivent également avoir été compilées avec des informations de débogage. Par exemple : « java -g *.java »

- Une nouvelle instance de la machine virtuelle d'application est lancée à partir d' Enterprise Architect - aucune autre action n'est requise
- La machine virtuelle d'application existante est attachée à partir d' Enterprise Architect - la Virtual Machine Java cible doit avoir été lancée avec l'agent de profilage Enterprise Architect

Voici des exemples de lignes de commande pour créer une machine virtuelle Java avec un agent JVMTI spécifique :

1. java.exe -cp "%classpath%;.\ " -agentpath:"C:\Program Files (x86)\ Sparx Systems \EA\vea\x86\ssamplerlib32" monapplication
2. java.exe -cp "%classpath%;.\ " -agentpath:"C:\Program Files (x86)\ Sparx Systems \EA\vea\x64\ssamplerlib64" monapplication

(Reportez-vous à la documentation du JDK pour plus de détails sur l'option de démarrage de la machine virtuelle -agentpath.)

Démarrage

Le Profiler peut être utilisé pour étudier les problèmes de performances, en vous proposant quatre outils distincts parmi lesquels choisir, à savoir :

- Graphique d'Appel
- Profil de Pile
- Profil de mémoire
- Fuites de Mémoire

Vous sélectionnez ces outils dans la barre d'outils du Profiler.

Accéder

Ruban	Exécuter > Outils > Profiler

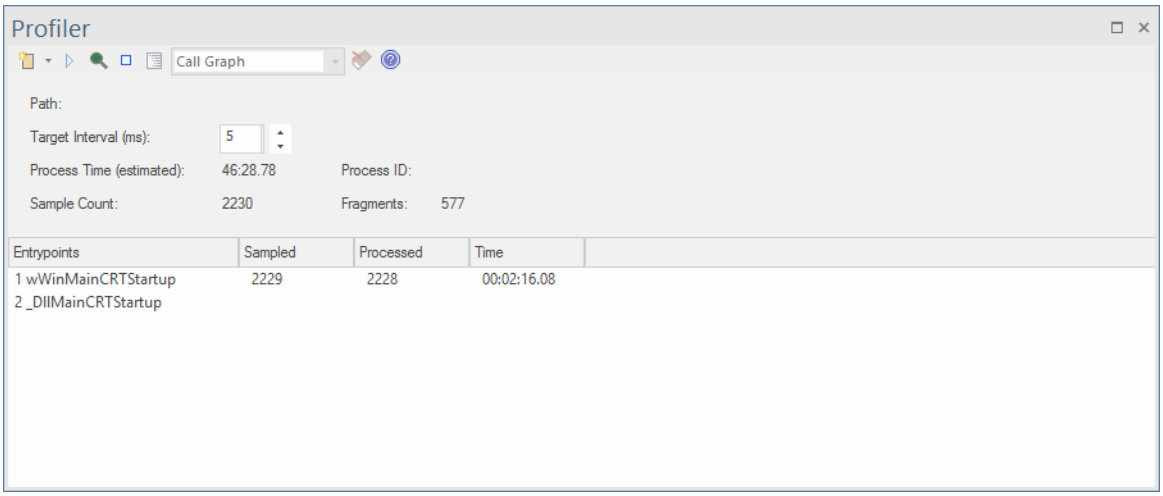
Outils

Outil	Description
Graphique d'Appel	Analyse les performances en prenant des échantillons au cours d'une activité dans un programme. Chaque échantillon représente une pile. Les échantillons sont prélevés à des intervalles contrôlés à l'aide de la barre d'outils. Dans ce scénario, les mauvaises performances sont évaluées en fonction des motifs de comportement qui se répètent le plus pendant la période d'échantillonnage. Ce chiffre est utilisé pour pondérer le Graphique d'Appel produit.
Profil de mémoire	Analyse les performances en exploitant les allocations de mémoire effectuées par un programme. Dans ce scénario, les performances médiocres sont évaluées en fonction des activités qui sollicitent le plus de mémoire. Ce chiffre est utilisé pour pondérer le Graphique d'Appel produit.
Profil de Pile	Le Stack Profiler vous permet de définir un marqueur dans votre code source afin que chaque fois que l'exécution atteint ce marqueur, une trace de pile complète soit capturée. Au fur et à mesure que l'application continue de s'exécuter et que la position marquée est accessible à partir de divers emplacements dans l'exécutable en cours d'exécution, une image très détaillée et utile est créée, montrant les points chauds et les scénarios d'utilisation pour un point particulier du code. Le rapport Profil de Pile , comme le rapport Profil de mémoire, s'affiche dans l'ordre « pile inversée ». Cela signifie que la racine du rapport est toujours un nœud unique (dans ce cas, le marqueur) et que l'arbre se déploie ensuite pour afficher tous les différents emplacements à partir desquels la position marquée a été accédée.
Fuites de Mémoire	Analyse les fuites de mémoire en récupérant les opérations de mémoire effectuées par un programme. Le résultat est un Graphique d'Appel présentant les piles d'appels qui alloué de la mémoire pour lesquelles aucune opération de libération n'a été détectée.

Boutons de la barre d'outils

Bouton	Action
	Affiche un menu d'options pour gérer votre session de profilage.
	Lance l'application configurée à profiler. Par défaut, il s'agit de l'application configurée dans le script Analyzer actif.
	Indique l'état de l'échantillonneur. Lorsqu'il est vert, l'échantillonnage est activé ; lorsqu'il est rouge, l'échantillonnage est désactivé.
	Arrête le processus du profileur ; si des échantillons ont été collectés, les boutons Rapport et Supprimer les données sont actifs.
	Génère un rapport à partir de la collecte de données actuelle.
	Affiche l'outil de profilage utilisé, qui détermine les champs affichés dans la fenêtre du profileur. Cliquez sur la flèche déroulante et sélectionnez un autre outil, ce qui modifie les champs de la fenêtre.
	Supprime les données collectées. Vous êtes invité à confirmer la suppression.
	Affiche la rubrique d'aide de cette fenêtre.

Graphique d'Appel



- Découvrez rapidement ce que fait un programme à tout moment
- Identifiez facilement les problèmes de performances
- Soyez surpris de la rapidité avec laquelle vous pouvez réaliser des améliorations
- Voyez vos améliorations au travail et ayez-en la preuve
- Support des plateformes C/C++, .NET et Java

Usage

L'option « Graphique d'Appel » est généralement utilisée dans les situations où une activité s'exécute plus lentement que prévu, mais elle peut également être utilisée simplement pour mieux comprendre les motifs de comportement en jeu au cours d'une activité.

Opération

Le Profiler fonctionne en prenant des échantillons (ou piles d'appels) à intervalles réguliers sur une période donnée ; l'intervalle est défini à l'aide de la barre d'outils du Profiler. Vous pouvez utiliser le Profiler pour exécuter un programme particulier ou vous pouvez vous connecter à un processus existant. La capture du Profiler est contrôlée et vous pouvez suspendre et reprendre la capture à tout moment. Vous pouvez également choisir de lancer la capture immédiatement au démarrage du Profiler. Si nécessaire, vous pouvez supprimer les échantillons capturés et recommencer au cours de la même session. Si vous ne pouvez pas continuer avec la même session, le redémarrage du Profiler est rapide et facile.

Note que le champ « Temps de processus (estimé) » affiche une estimation de la durée d'exécution du processus en cours de profilage, en tenant compte des interruptions du processus par le profileur lors de la collecte des échantillons.

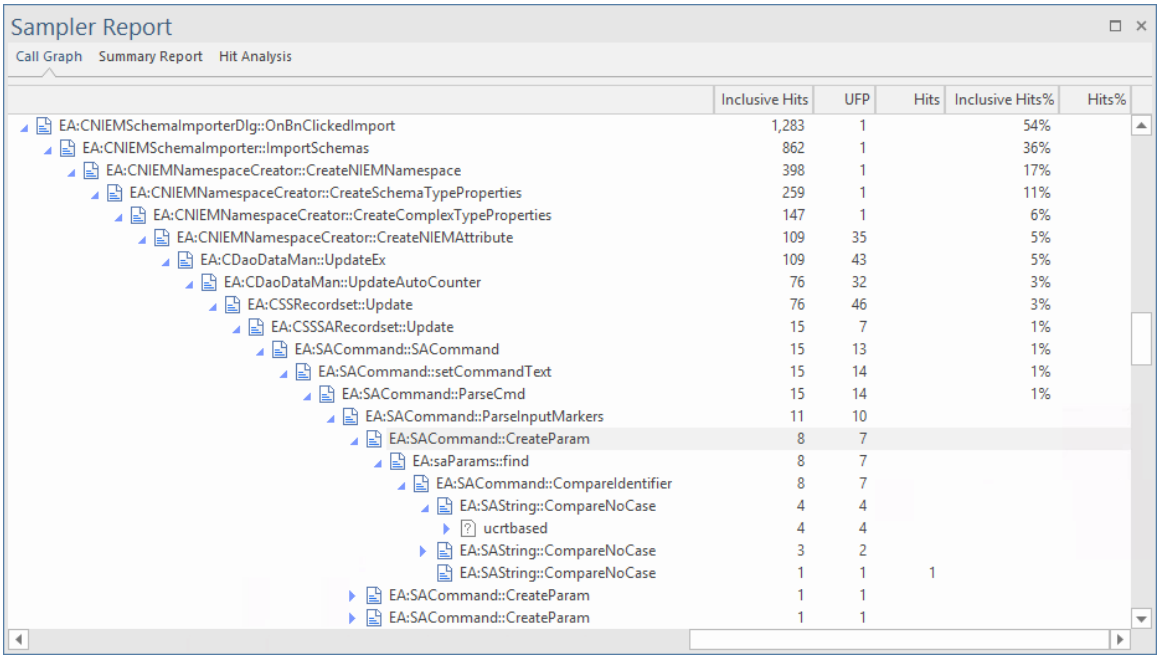
Résultats

Les résultats peuvent être produits à tout moment au cours de la session. Cependant, la capture doit être désactivée pour que le bouton Rapport devienne actif. C'est à vous de décider combien de temps vous laissez le Profiler exécuter . Vous pouvez savoir quand une activité est terminée, ou cela peut être apparent pour d'autres raisons. La raison pour laquelle vous êtes ici peut être qu'une activité ne se termine pas du tout.

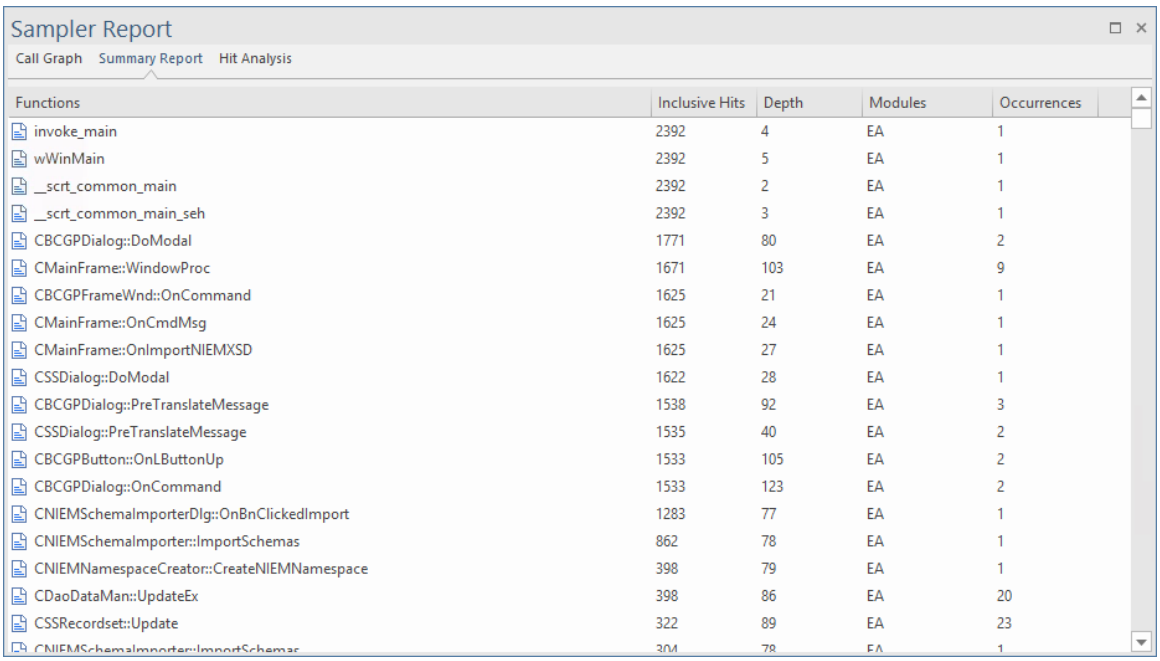
Le bouton Rapport sera activé soit en mettant la capture en pause, soit en arrêtant complètement le Profiler.

Les résultats sont affichés dans une vue Rapport . Le rapport s'ouvre avec trois onglets initialement visibles : le Graphique d'Appel , le Rapport de synthèse (Résumé des fonctions) et les onglets Analyse Hit . Les rapports peuvent être enregistrés dans un fichier, stockés dans le modèle sous forme d'Artefacts ou publiés dans la Bibliothèque d'Équipe .

L'onglet Graphique d'Appel



L'onglet Rapport récapitulatif



L'onglet Analyse Hit

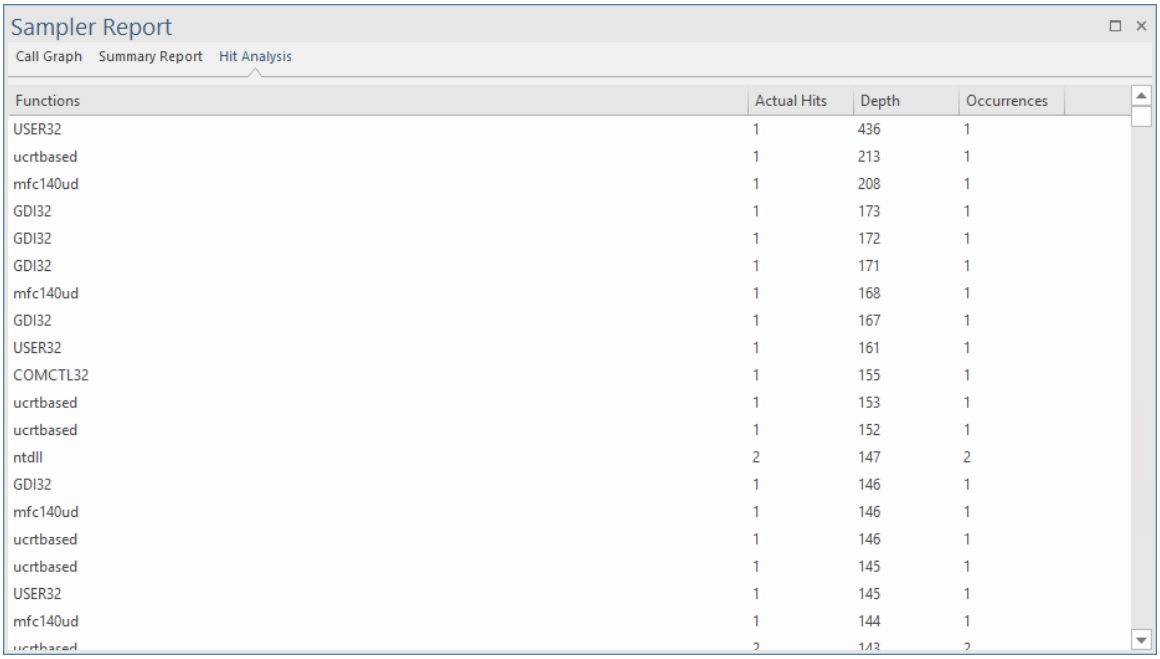
L'onglet « Analyse Hit » affiche un certain nombre de colonnes :

- Fonction : le nom de la fonction (ou du module s'il n'y a pas de symbole pour le module)
- Hits : le nombre d'échantillons prélevés, dans lesquels la fonction était en cours d'exécution.
- Profondeur : le numéro de l'image ou la profondeur de la pile à laquelle le coup a eu lieu.

- Occurrences : le nombre de fois où la fonction a été touchée à cette profondeur de pile particulière

Le nombre de hits sur une fonction particulière est agrégé en fonction de la profondeur de la trame de pile lors de l'échantillonnage.

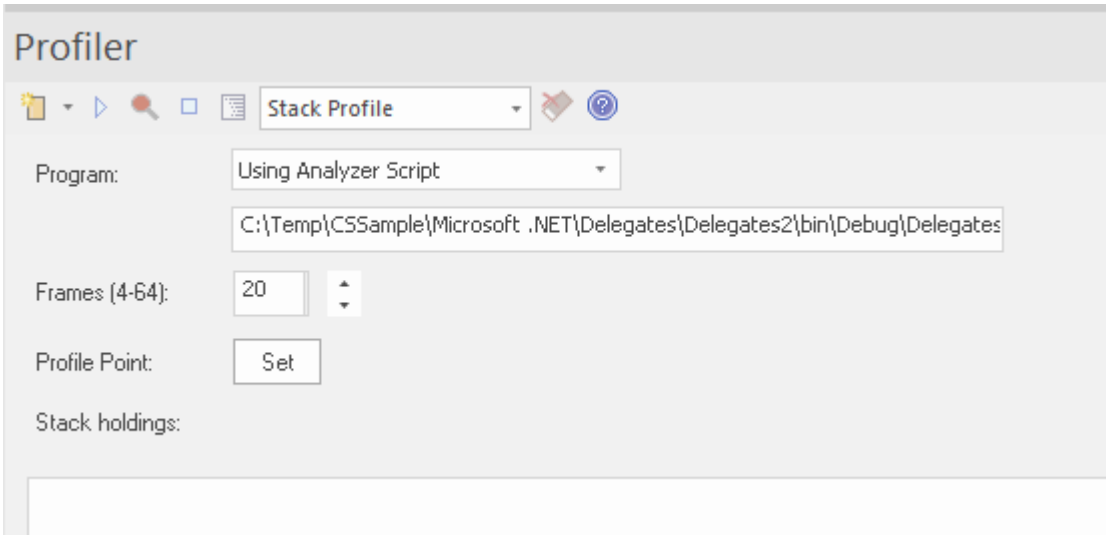
Si le nom de la fonction n'est pas disponible, par exemple les DLL système Windows telles que User32 ou les DLL sans informations de débogage, le nom du module est affiché à la place.



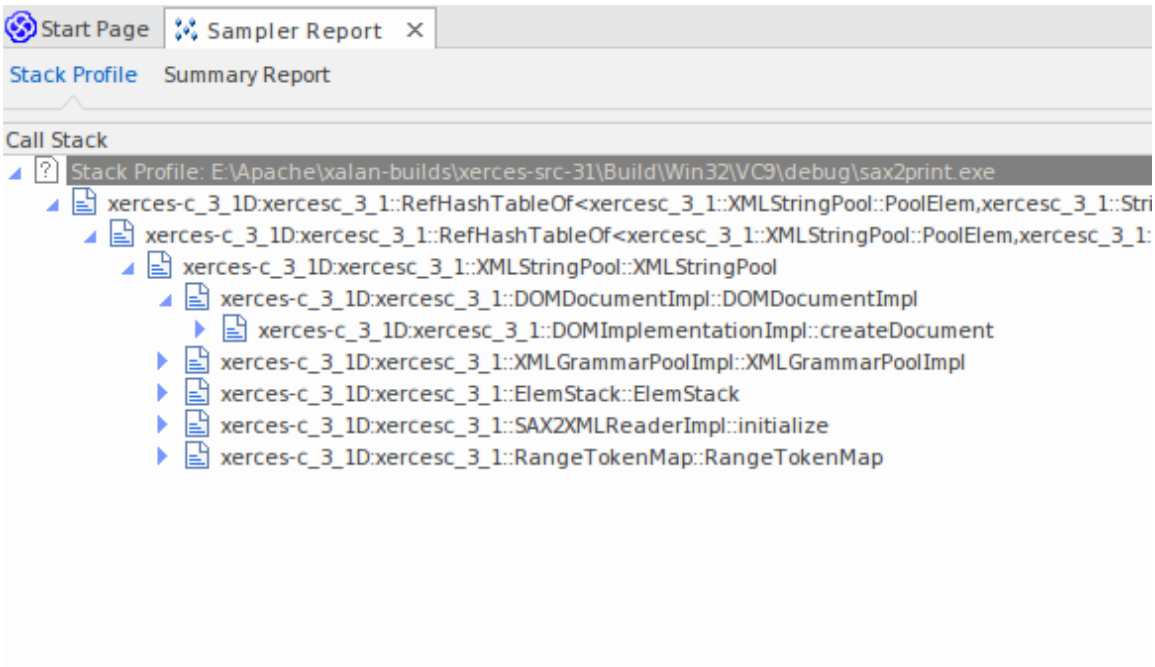
Functions	Actual Hits	Depth	Occurrences
USER32	1	436	1
ucrtbased	1	213	1
mfc140ud	1	208	1
GDI32	1	173	1
GDI32	1	172	1
GDI32	1	171	1
mfc140ud	1	168	1
GDI32	1	167	1
USER32	1	161	1
COMCTL32	1	155	1
ucrtbased	1	153	1
ucrtbased	1	152	1
ntdll	2	147	2
GDI32	1	146	1
mfc140ud	1	146	1
ucrtbased	1	146	1
ucrtbased	1	145	1
USER32	1	145	1
mfc140ud	1	144	1
ucrtbased	2	143	2

Profil de Pile

Le Stack Profiler vous permet de définir un marqueur dans votre code source afin que chaque fois que l'exécution atteint ce marqueur, une trace de pile complète soit capturée. Au fur et à mesure que l'application continue de s'exécuter et que la position marquée est accessible à partir de divers emplacements dans l'exécutable en cours d'exécution, une image très détaillée et utile est créée, montrant les points chauds et les scénarios d'utilisation pour un point particulier du code.



Le rapport Profil de Pile , comme le rapport Profil de mémoire, s'affiche dans l'ordre « pile inversée ». Cela signifie que la racine du rapport est toujours un nœud unique (dans ce cas, le marqueur) et que l'arbre se déploie ensuite pour afficher tous les différents emplacements à partir desquels la position marquée a été accédée.



Usage

Utilisez le mode Profil de Pile pour générer un rapport qui montre les différentes manières dont une fonction peut être invoquée pendant l'exécution d'un programme. Déterminez les parties du modèle qui s'appuient sur cette fonction et leur fréquence.

Opération

```
106
107 template <class TVal, class THasher>
108 void RefHashTableOf<TVal, THasher>::initialize(const XMLSize_t modulus)
109 {
110     if (modulus == 0)
111         ThrowXMLwithMemMgr(IllegalArgumentException, XMLExcepts::HshTbl_ZeroMo
112
113     // Allocate the bucket list and zero them
114     fBucketList = (RefHashTableBucketElem<TVal>**) fMemoryManager->allocate
115     (
116         fHashModulus * sizeof(RefHashTableBucketElem<TVal>*)
117     );
118     for (XMLSize_t index = 0; index < fHashModulus; index++)
119         fBucketList[index] = 0;
120 }
121
```

Les modes de profilage sont sélectionnés à l'aide de la barre d'outils du contrôle Profiler. Si un point de profilage est déjà créé, il est affiché. Le point de profilage est le point auquel les traces de pile sont capturées. Vous pouvez définir le point de profilage à l'aide du bouton Définir sur le contrôle lui-même, une fois le mode sélectionné. Après avoir choisi le point de profilage, générez le projet pour vous assurer que tout est à jour, puis démarrez le profileur. Le nombre de holdings de pile uniques détectés est visible pendant l' exécuter .

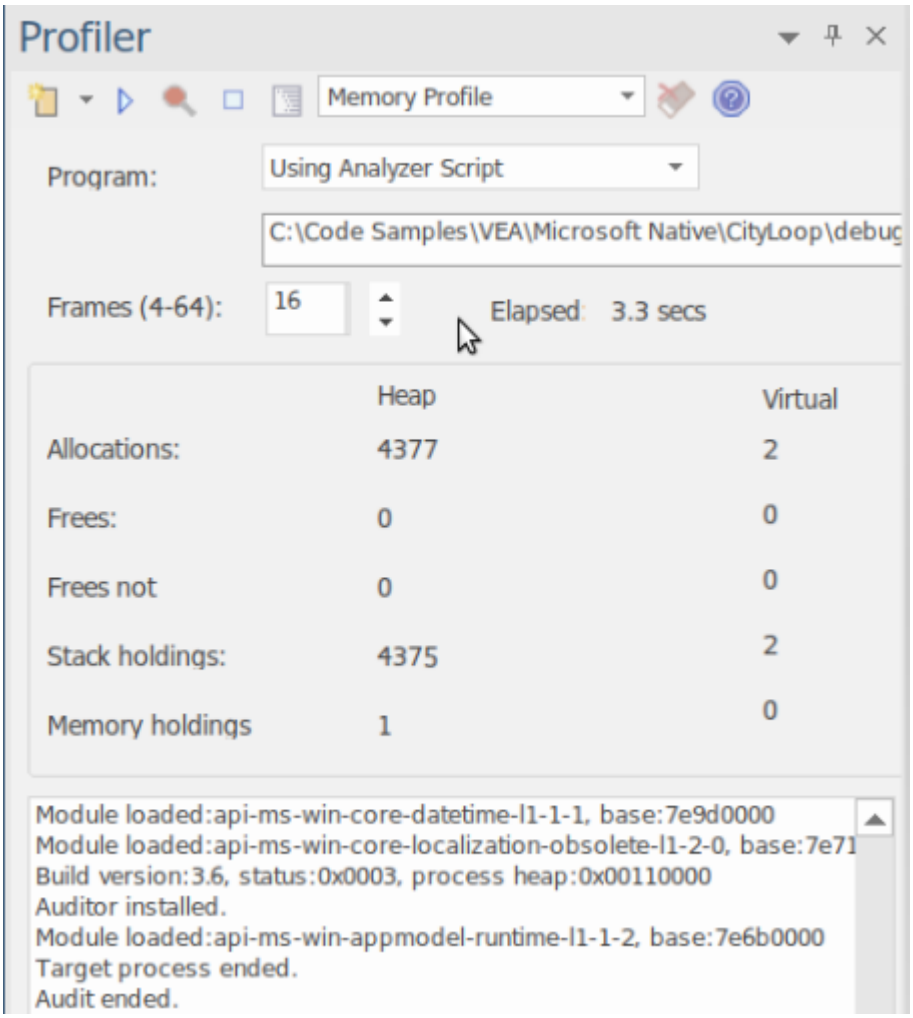
Résultats

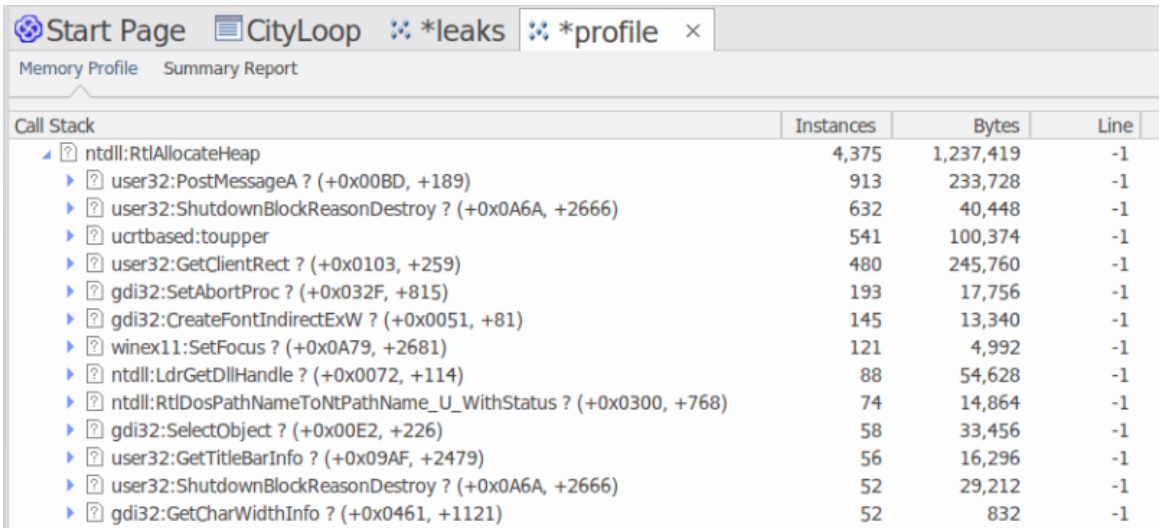
Un résultat peut être généré en cliquant sur le bouton Rapport dans la barre d'outils du contrôle du profileur. Ce bouton est activé lorsque :

- La capture est désactivée (à l'aide du bouton Pause) ou
- Le profileur est arrêté (en utilisant le bouton Stop)

Les résultats produits sont affichés sous forme de graphique d'appels pondérés, où les lignes du graphique représentent une pile unique et sont pondérées pour afficher en premier les piles à fréquence plus élevée. Le rapport peut ensuite être enregistré, soit dans un fichier, soit dans le modèle, à l'aide du menu contextuel du rapport lui-même.

Profil de mémoire





- Évaluez rapidement les performances des activités qui vous intéressent
- Rien n'influence davantage une discussion que les preuves
- Récompensez vos efforts en travaillant dans les domaines qui feront la différence

- Surprenez-vous en proposant des optimisations dont vous ignoriez peut-être l'existence

Usage

Le profil de mémoire peut être utilisé pour révéler les performances des activités en termes de consommation de mémoire. En utilisant ce mode, un utilisateur pourrait s'intéresser à la fréquence des demandes de mémoire au cours d'une tâche. Il serait moins intéressé par la quantité réelle consommée. Une activité bien gérée peut effectuer relativement peu d'appels pour allouer des ressources, mais allouer suffisamment de mémoire pour effectuer son travail efficacement. D'autres activités peuvent effectuer plusieurs milliers de demandes, ce qui les rend généralement moins efficaces. Ce mode est utile pour détecter ces scénarios.

Opération

Le profil de mémoire fonctionne en accrochant le processus en question, de sorte que le programme doit être lancé à l'aide de l'outil dans Enterprise Architect . Contrairement à l'option Graphique d'Appel , vous ne pouvez pas vous rattacher à un processus existant. Lorsque le programme est démarré, les mécanismes d'accrochage suivent l'allocation de mémoire ; ces informations sont collectées et rassemblées dans Enterprise Architect . Vous pouvez facilement surveiller le nombre d'allocations effectuées. De plus, le processus est contrôlé ; c'est-à-dire que les crochets de mémoire peuvent être activés et désactivés à la demande. Si vous avez mal synchronisé une action, vous pouvez suspendre la capture, supprimer les données et reprendre la capture facilement.

Résultats

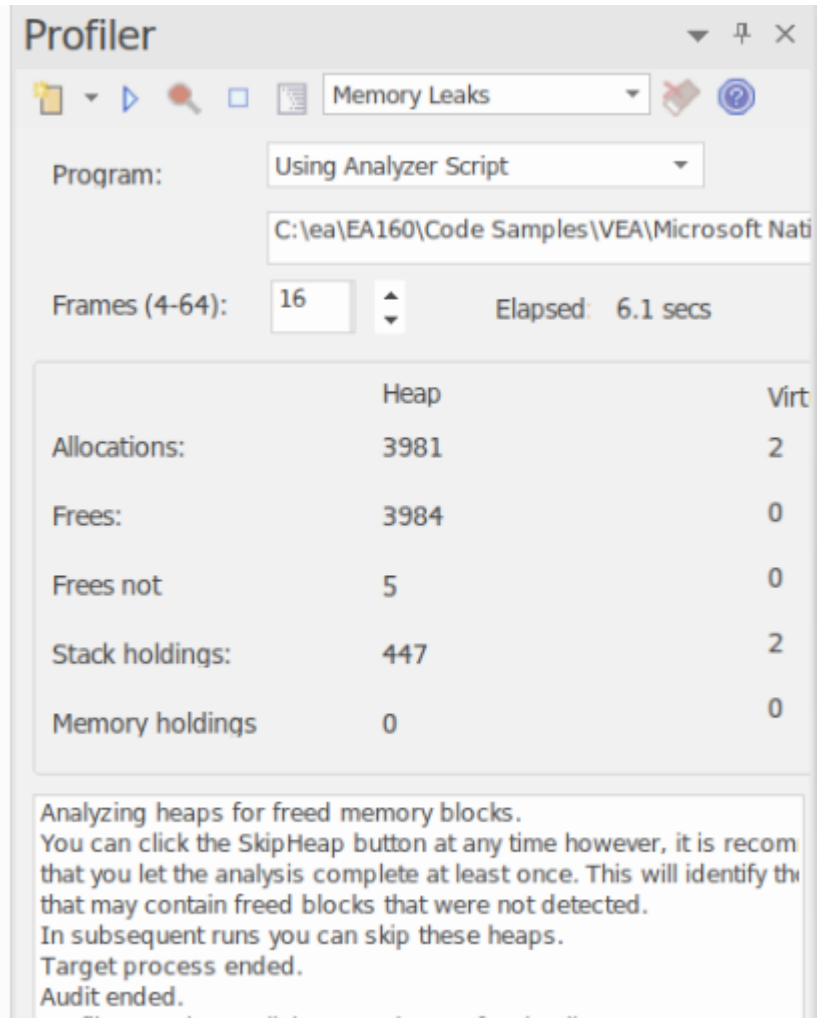
Les résultats peuvent être produits à tout moment au cours de la session. Cependant, la capture doit être désactivée pour que le bouton Rapport devienne actif. C'est vous qui décidez combien de temps vous laissez le Profiler exécuter . Vous activez le bouton Rapport en mettant en pause la capture ou en arrêtant complètement le Profiler.

Les résultats sont affichés dans une vue Rapport . Le rapport s'ouvre initialement avec deux onglets visibles : un seul Graphique d'Appel pondéré et un résumé de fonction. Le Graphique d'Appel représente toutes les piles d'appels qui ont conduit à des allocations de mémoire, qui sont agrégées et pondérées en fonction de la fréquence du motif .

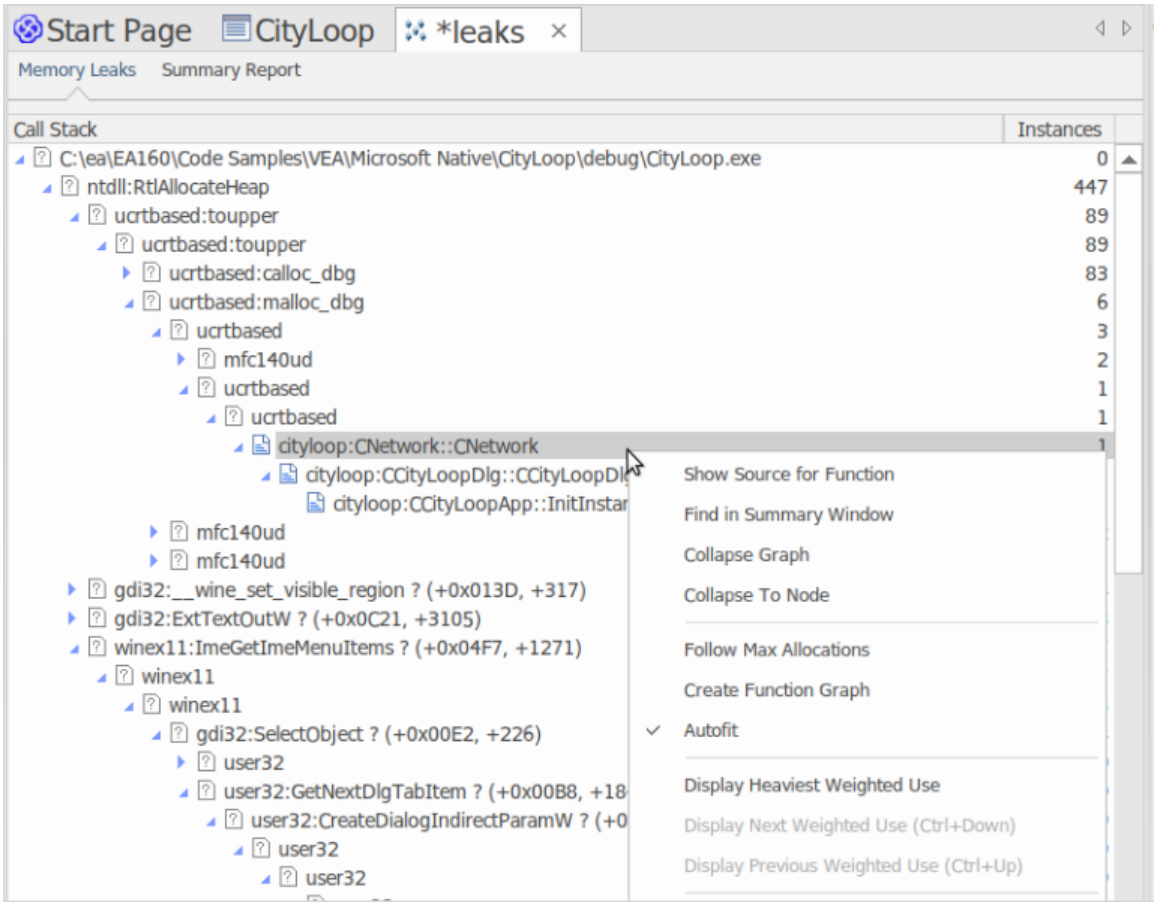
Exigences

Pour obtenir les meilleurs résultats, l'image et ses modules doivent être créés avec les informations de débogage incluses et sans optimisations. Tout module doté de l'optimisation Frame Pointer Omission (FPO) est susceptible de produire des résultats trompeurs.

Fuites de Mémoire



Le contrôle Profiler, affichant le nombre d'allocations de mémoire et le nombre d'opérations qui libèrent de la mémoire.



Un programme bien mené.

La détection des fuites de mémoire est une voie bien connue. Bien que de nombreuses autres options intéressantes soient disponibles, nous pensons que notre approche présente des avantages majeurs, tels que :

- Aucun changement n'a été apporté à la version existante du projet
- Aucun fichier d'en-tête requis par le code du projet
- Aucune dépendance d'exécution à prendre en compte
- Aucune configuration système à prendre en compte

Usage

Une personne peut utiliser ce mode pour suivre les fuites de mémoire dans une application ou dans une activité au sein de l'application. Une fuite de mémoire du point de vue du profileur est un appel réussi effectué vers une fonction d'allocation de mémoire qui renvoie une adresse mémoire pour laquelle aucun appel correspondant n'est effectué pour libérer cette adresse.

Opération

La détection des fuites de mémoire fonctionne par hooking. Les routines de mémoire du processus sont hookées pour suivre le moment où la mémoire est à la fois alloué et libérée. Les piles d'appels sont capturées au moment de l'allocation et ces informations sont rassemblées dans Enterprise Architect pour produire un rapport sous la forme d'un Graphique d'Appel . La capture est contrôlée ; c'est-à-dire que les mécanismes de hooking peuvent être activés ou désactivés à la demande.

Selon le type de programme et sa consommation de mémoire, vous pouvez employer une stratégie appropriée. Pour les petits programmes, vous pouvez suivre le programme du début à la fin. Pour les programmes à fenêtres plus volumineux,

il serait probablement préférable d'activer la capture avant et après une tâche spécifique pour éviter de suivre trop de données.

Résultats

Les résultats peuvent être produits à tout moment au cours de la session. Cependant, la capture doit être désactivée pour que le bouton Rapport devienne actif. C'est vous qui décidez combien de temps vous laissez le Profiler exécuter . Vous activez le bouton Rapport en mettant en pause la capture ou en arrêtant complètement le Profiler.

Les résultats sont affichés dans une vue Rapport . Le rapport s'ouvre initialement avec deux onglets visibles : un seul Graphique d'Appel pondéré et un résumé de fonction. Le Graphique d'Appel décrit toutes les piles d'appels qui ont conduit à des allocations de mémoire et sont agrégées et pondérées en fonction de la fréquence du motif .

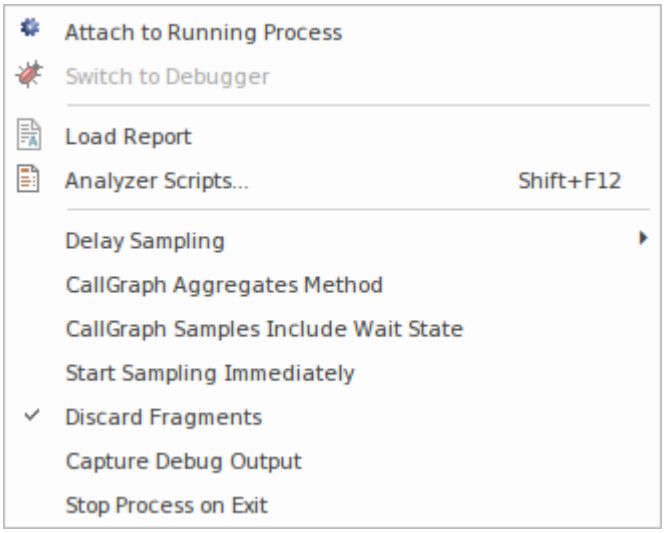
Rapports peuvent contenir une quantité variable de « bruit ». Pour vous concentrer sur un domaine qui vous préoccupe particulièrement, recherchez une fonction que vous connaissez dans le rapport récapitulatif et utilisez-la pour accéder directement à la ligne du graphique où elle est présentée.

Exigences

Pour obtenir les meilleurs résultats, l'image et ses modules doivent être créés avec les informations de débogage incluses et sans optimisations. Tout module doté de l'optimisation Frame Pointer Omission (FPO) est susceptible de produire des résultats trompeurs.

Options de Réglage

La première icône de la barre d'outils de la fenêtre Profiler affiche une liste d'options que vous pouvez définir pour personnaliser votre session de profilage.



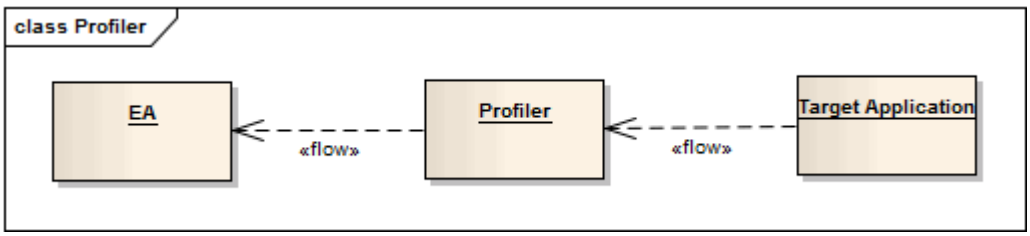
Options

Option	Description
Attacher au processus en cours d'exécution	Sélectionnez cette option pour afficher la dialogue « Attacher au processus », à partir de laquelle vous choisissez un processus actif à profiler.
Passer à Débogueur	Sélectionnez cette option pour passer des opérations de profilage à débogage. Le Débogueur dispose d'une option de menu déroulant équivalente que vous pouvez utiliser pour passer du débogage au profilage.
Rapport de charge	Sélectionnez cette option pour charger un rapport précédemment enregistré à partir du système de fichiers.
Scripts d'Analyseur	Sélectionnez cette option pour ouvrir la fenêtre Analyzer Script, qui est le référentiel modèle pour la configuration des builds, le débogage et toutes les autres options Analyseur d'Exécution Visuelle .
Échantillonnage différé	Sélectionnez cette option pour définir un délai entre le moment où vous cliquez sur l'option « Démarrer le profilage » et le début effectif du profilage. Le délai peut être de 3, 5 ou 10 secondes. Sélectionnez « Aucun » pour annuler tout délai défini.
Méthode d'agrégation CallGraph	Lorsque cette option est sélectionnée, les instances des séquences de pile identiques sont agrégées par méthode. En d'autres termes, les numéros de ligne/instructions au sein d'une méthode sont ignorés, de sorte que deux piles seront comptées comme une seule lorsqu'elles ne diffèrent que par le numéro de ligne dans leur trame finale.
Les exemples de CallGraph incluent State d'attente	Lorsque cette option est sélectionnée, le profileur échantillonne tous les threads, y compris ceux en état d'attente. Lorsqu'elle n'est pas sélectionnée, le profileur échantillonne uniquement les threads qui ont accumulé du temps CPU depuis

	l'expiration du dernier intervalle.
Démarrer l'échantillonnage immédiatement	Sélectionnez cette option pour déclencheur la collecte de données immédiatement au lancement. Vous utiliserez généralement cette option pour profiler un processus au démarrage.
Jeter les fragments	Lorsque les piles ne peuvent pas être rapprochées du point d'entrée d'un thread, elles sont appelées fragments. Le nombre de fragments rencontrés lors de l'échantillonnage est affiché dans la fenêtre Résumé de l'échantillonneur. Vous pouvez définir cette option pour collecter ou supprimer des fragments ; lorsque l'option Supprimer les fragments est définie sur : • Sélectionné, les fragments n'apparaissent pas dans les rapports, bien que le nombre rencontré soit toujours mis à jour • Désélectionné, une collection spéciale nommée « fragments » est créée dans le graphique d'appel pour les héberger et garantir que les données ne sont pas mélangées aux échantillons complets
Capturer la sortie Déboguer	(S'applique à l'échantillonnage de processus). Lorsque cette option est sélectionnée, la sortie normalement visible pendant le débogage est capturée et affichée dans la fenêtre Déboguer . Note que seules les versions de débogage émettront généralement une sortie de débogage.
Arrêter le processus à la sortie	Cette option détermine le comportement de fin du profileur. Lorsque l'option est sélectionnée, le processus cible se termine lorsque le profileur est arrêté.

Démarrer et Arrêter le Profileur

Le profilage est un processus en deux étapes de collecte de données et de création de rapports. Dans Enterprise Architect la collecte de données présente l'avantage d'être une tâche d'arrière-plan : vous êtes donc libre de faire d'autres choses pendant l'exécution. Les informations renvoyées à Enterprise Architect sont stockées jusqu'à ce que vous génériez un rapport. Pour afficher un rapport, la capture doit être désactivée. Une fois le rapport généré, vous pouvez reprendre la capture en cliquant sur un bouton. Si, pour une raison quelconque, vous décidez de supprimer vos données et de recommencer, vous pouvez le faire facilement et sans avoir à arrêter et redémarrer le programme.



Accéder

Ruban	Exécuter > Outils > Profiler > Ouvrir le profileur
Autre	Barre d'outils Analyseur d'Exécution : Analyseur Windows | Profileur

Actes

Action	Détail
Barre d'outils	
Sélection de stratégie	Sélectionnez la stratégie de profilage parmi les options disponibles dans la barre d'outils.
Démarrer le Profileur	Cliquez sur le bouton Exécuter dans la fenêtre du Profiler
Arrêter le Profileur	Le processus se termine si : • Vous cliquez sur le bouton Stop • L'application cible se termine, ou • Vous fermez le modèle actuel Si vous arrêtez le Profiler et que le processus est toujours en cours d'exécution, vous pouvez rapidement vous y connecter à nouveau.
Pause et Reprise de Capture	Vous pouvez mettre en pause et reprendre la capture à tout moment pendant une session. Lorsque la capture est activée, des échantillons sont collectés à partir de la cible. Lorsqu'il est en pause, le profileur entre et reste dans un état d'attente jusqu'à ce que

	la capture soit activée, que le profileur soit arrêté ou que l'application se termine.
Générer Rapports	Le bouton Rapport est désactivé pendant la capture mais est disponible lorsque la capture est désactivée.
Mode déroulant	Cliquez sur le menu déroulant et sélectionnez le mode de Profilage - Graphique d'Appel , Profil de Pile , Memory Profile ou Fuites de Mémoire .
Effacer la collecte de données	Vous pouvez effacer tous les échantillons de données collectés et reprendre à tout moment. Suspendez d'abord la capture en cliquant sur le bouton Pause. Le bouton Annuler, comme le bouton Rapport , est activé chaque fois que la capture est désactivée. En cliquant sur le bouton Annuler, il vous sera demandé de confirmer l'opération. Cette action ne peut pas être annulée.

Rapports de Ligne de Fonction

Après avoir exécuter le profileur sur une application en cours d'exécution et généré un rapport d'échantillon, vous pouvez analyser plus en détail l'activité d'une fonction spécifique répertoriée dans le rapport en générant un rapport de ligne de fonction à partir de cet élément. Un rapport de ligne de fonction indique le nombre de fois que chaque ligne de la fonction a été exécutée. Vous produisez un rapport de ligne de fonction à la fois, sur n'importe quelle méthode du rapport d'échantillon qui possède un fichier source valide. Le rapport de ligne de fonction est particulièrement utile pour les fonctions qui exécutent des boucles contenant des ramifications conditionnelles ; la couverture peut fournir une image des parties de code les plus et les moins fréquemment exécutées au sein d'une seule méthode.

Le rapport de ligne que vous générez est enregistré lorsque vous enregistrez le rapport Sampler. Le corps de la fonction est également enregistré avec le rapport de ligne de fonction pour conserver l'état de la fonction à ce moment-là.

Cette facilité ne s'applique pas aux rapports de profil mémoire.

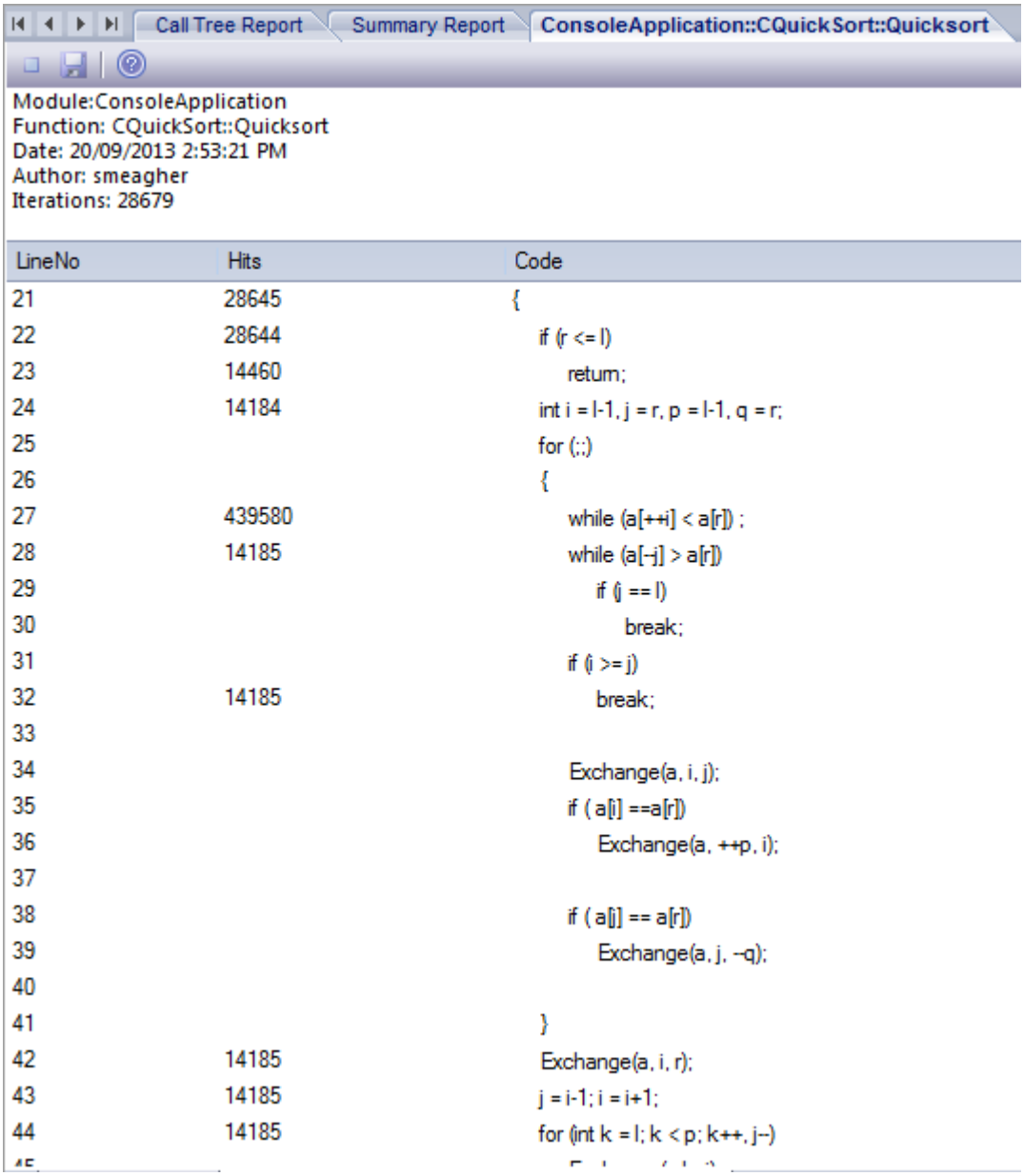
Plateformes prises en charge

Java, Microsoft .NET et code natif Microsoft

Créer un Rapport de Ligne

Dans le rapport Sampler, cliquez-droit sur le nom de la fonction à analyser, et sélectionnez l'option 'Créer Rapport de ligne pour la fonction'.

Une fois que le profileur a lié la méthode, le rapport de ligne de fonction s'ouvre dans la fenêtre Rapport d'échantillon. Le rapport affiche le corps de la fonction, y compris les numéros de ligne et le texte. Au fur et à mesure que chaque ligne est exécutée, une valeur de hit s'accumule sur cette ligne. Un minuteur met à jour le rapport environ une fois par seconde.



Terminer Capture du Rapport de Ligne

Une fois que suffisamment d'informations sont capturées ou que la fonction est terminée, cliquez sur le bouton Arrêter de la barre d'outils du Profiler pour arrêter l'enregistrement de la capture.

Enregistrer Rapports

Utilisez le bouton Enregistrer de la barre d'outils Pile d'Appel pour enregistrer le rapport d'échantillonneur et tous les rapports de ligne de fonction dans un fichier.

Supprimer Rapports de ligne

La fermeture de l'onglet « Rapport de ligne » fermera ce rapport, mais les données du rapport ne seront supprimées que lorsque le rapport sera enregistré.

Générer, Enregistrer et Charger des Rapports de Profil

Rapports peuvent être produits à tout moment au cours d'une session, ou naturellement à la fin d'un programme. Pour activer le bouton Rapport pendant que le programme est en cours d'exécution, vous devez cependant suspendre le profilage en activant le bouton Pause/Reprendre, ou en mettant fin au profileur avec le bouton Arrêter. Vous disposez de plusieurs options pour réviser et partager les résultats :

- Vue le rapport
- Enregistrer le rapport dans le fichier
- Distribuer le rapport comme ressource Bibliothèque d'Équipe
- Joindre le rapport en tant que document à un élément d'artefact
- Synchroniser le modèle en procédant à une rétro-ingénierie du code source ayant participé au profil

Accéder

Ruban	Exécuter > Outils > Profiler > Créer Rapport à partir des données actuelles
Profileur	Depuis la fenêtre du Profiler, cliquez sur l'icône  dans la barre d'outils.

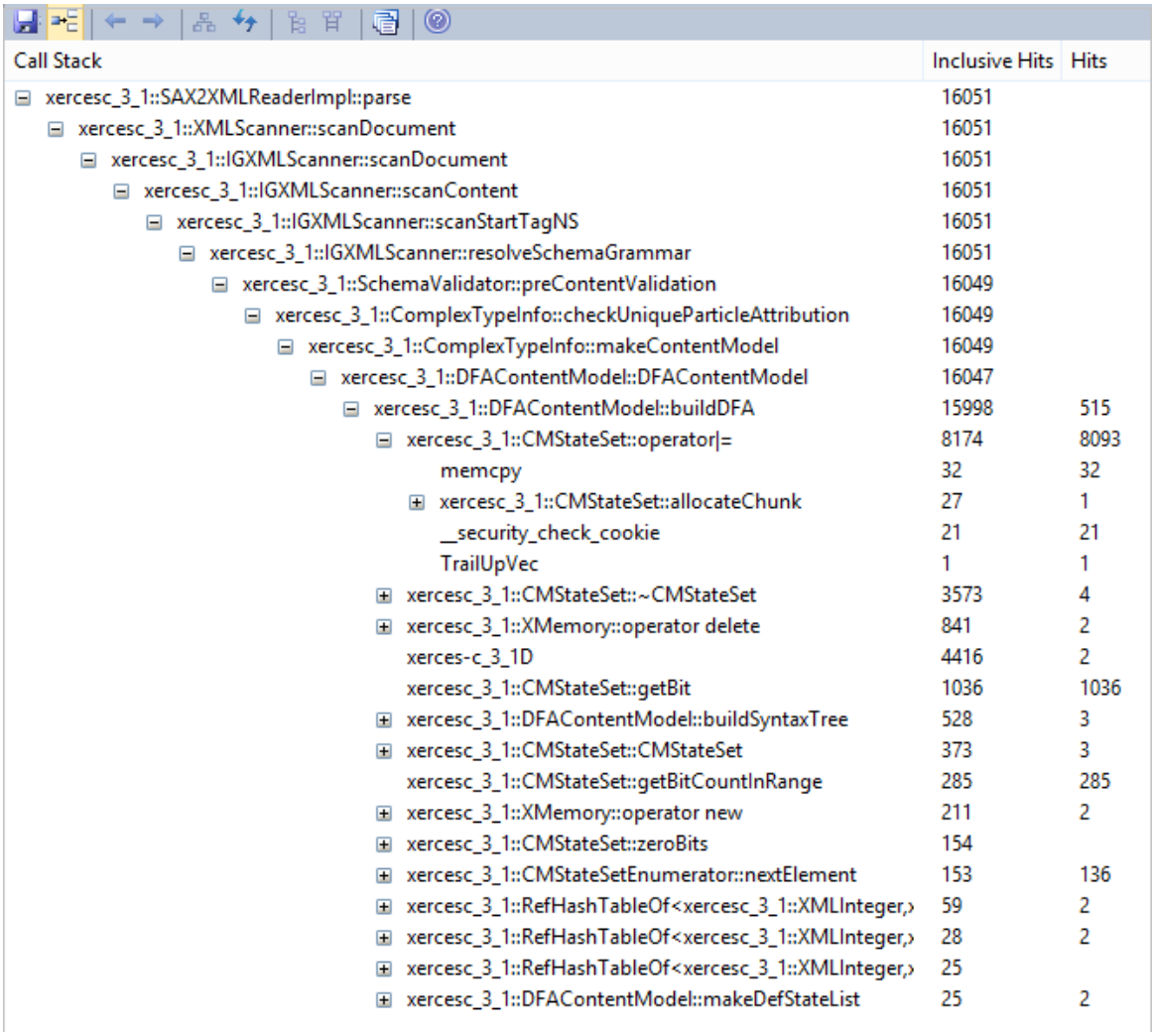
Charger Rapport à partir du fichier

L'option est disponible dans le menu déroulant de la fenêtre du profileur

Générer Rapport

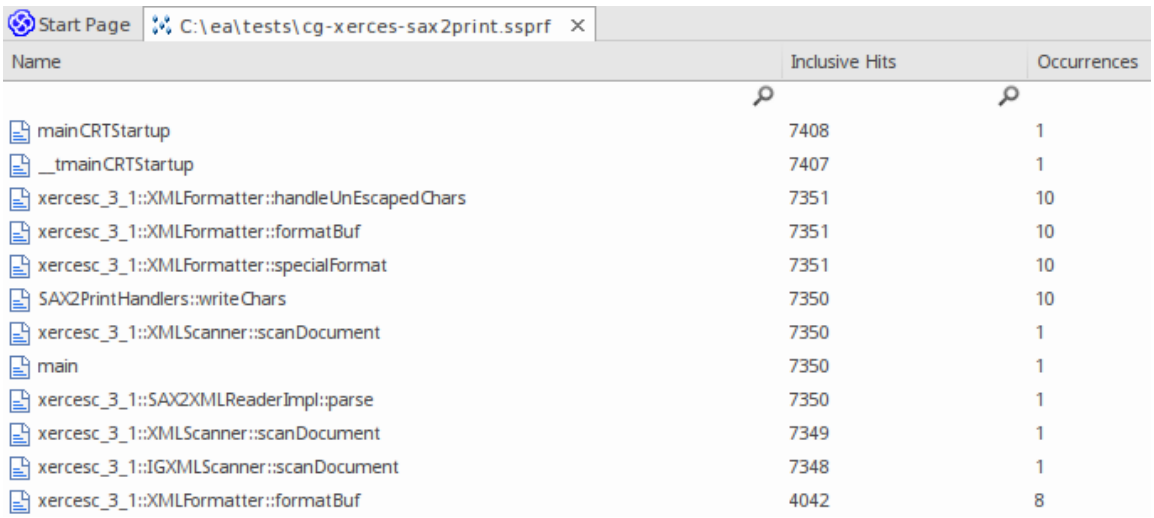
Depuis la fenêtre du Profiler, cliquez sur l'icône  dans la barre d'outils.

Rapport sur la fréquence des appels



Call Stack	Inclusive Hits	Hits
xercesc_3_1::SAX2XMLReaderImpl::parse	16051	
xercesc_3_1::XMLScanner::scanDocument	16051	
xercesc_3_1::IGXMLScanner::scanDocument	16051	
xercesc_3_1::IGXMLScanner::scanContent	16051	
xercesc_3_1::IGXMLScanner::scanStartTagNS	16051	
xercesc_3_1::IGXMLScanner::resolveSchemaGrammar	16051	
xercesc_3_1::SchemaValidator::preContentValidation	16049	
xercesc_3_1::ComplexTypeInfo::checkUniqueParticleAttribution	16049	
xercesc_3_1::ComplexTypeInfo::makeContentModel	16049	
xercesc_3_1::DFAContentModel::DFAContentModel	16047	
xercesc_3_1::DFAContentModel::buildDFA	15998	515
xercesc_3_1::CMStateSet::operator|=	8174	8093
memcpy	32	32
xercesc_3_1::CMStateSet::allocateChunk	27	1
__security_check_cookie	21	21
TrailUpVec	1	1
xercesc_3_1::CMStateSet::~CMStateSet	3573	4
xercesc_3_1::XMemory::operator delete	841	2
xerces-c_3_1D	4416	2
xercesc_3_1::CMStateSet::getBit	1036	1036
xercesc_3_1::DFAContentModel::buildSyntaxTree	528	3
xercesc_3_1::CMStateSet::CMStateSet	373	3
xercesc_3_1::CMStateSet::getBitCountInRange	285	285
xercesc_3_1::XMemory::operator new	211	2
xercesc_3_1::CMStateSet::zeroBits	154	
xercesc_3_1::CMStateSetEnumerator::nextElement	153	136
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	59	2
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	28	2
xercesc_3_1::RefHashTableOf<xercesc_3_1::XMLInteger,›	25	
xercesc_3_1::DFAContentModel::makeDefStateList	25	2

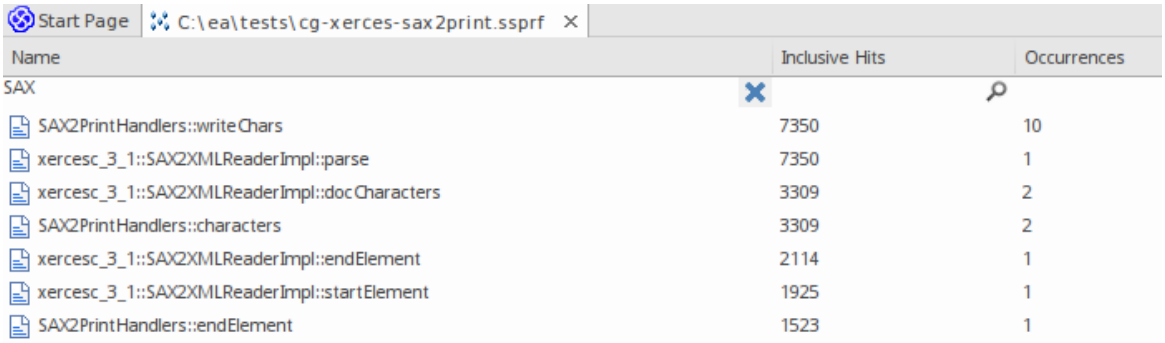
Résumé des fonctions



Start Page | C:\ea\tests\cg-xerces-sax2print.ssprf

Name	Inclusive Hits	Occurrences
mainCRTStartup	7408	1
__tmainCRTStartup	7407	1
xercesc_3_1::XMLFormatter::handleUnEscapedChars	7351	10
xercesc_3_1::XMLFormatter::formatBuf	7351	10
xercesc_3_1::XMLFormatter::specialFormat	7351	10
SAX2PrintHandlers::writeChars	7350	10
xercesc_3_1::XMLScanner::scanDocument	7350	1
main	7350	1
xercesc_3_1::SAX2XMLReaderImpl::parse	7350	1
xercesc_3_1::XMLScanner::scanDocument	7349	1
xercesc_3_1::IGXMLScanner::scanDocument	7348	1
xercesc_3_1::XMLFormatter::formatBuf	4042	8

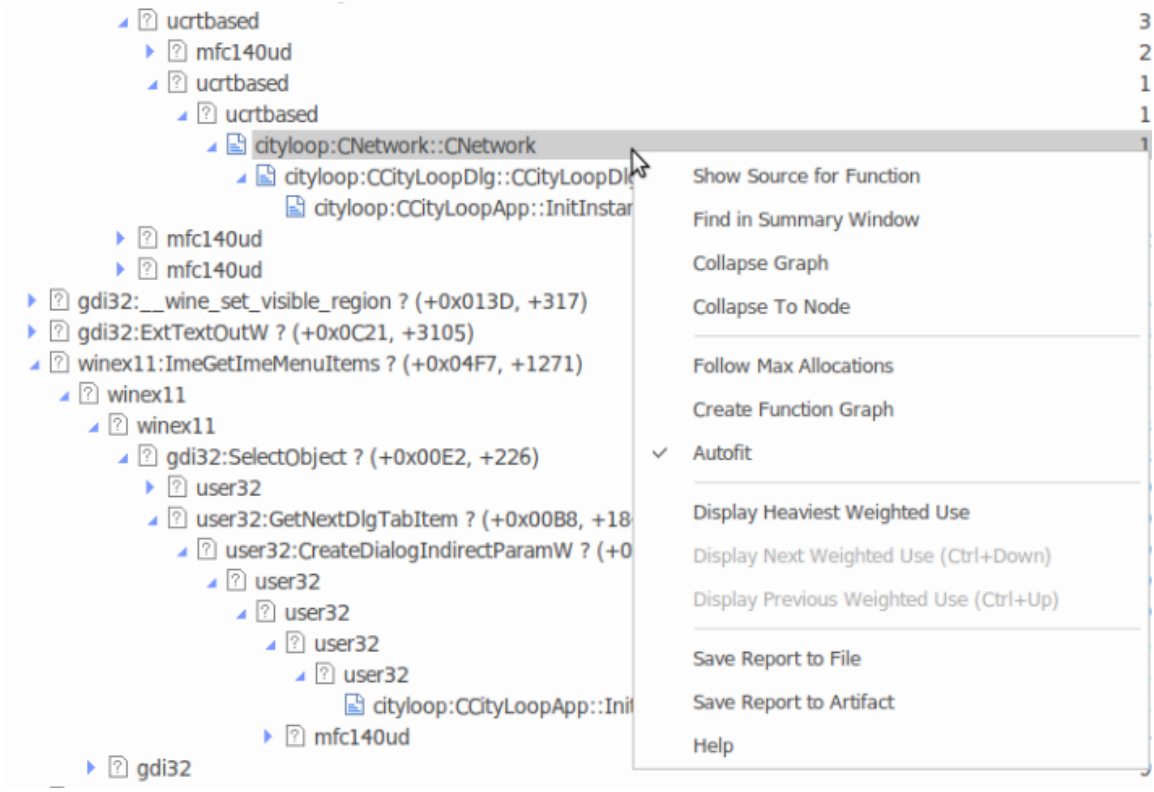
Rapport récapitulatif non filtré répertoriant toutes les fonctions participantes par ordre de succès inclusifs.



Vous pouvez filtrer et réorganiser les informations du rapport, de la même manière que vous le faites pour les résultats d'une recherche Modèle .

Options Rapport

Cliquez-droit sur le rapport pour afficher le menu contextuel.



Note que les options répertoriées dépendent du type de rapport affiché ; le rapport illustré ici est un rapport de profil de mémoire.

Action	Détail
Afficher la source de la fonction	Pour le cadre sélectionné, sélectionnez cette option pour afficher la ligne de code correspondante dans un éditeur de code. Les cadres dont la source est disponible sont identifiables par leur icône.
Rechercher dans la fenêtre de résumé	Sélectionnez cette option pour localiser la fonction dans la fenêtre Résumé.
Graphique de réduction	Sélectionnez cette option pour réduire l'intégralité du graphique, y compris les

	nœuds enfants, visibles ou non.
Réduire au nœud	Sélectionnez cette option pour réduire l'intégralité du graphique, puis le développer et définir le focus sur le nœud sélectionné.
Suivre les allocations maximales	Sélectionnez cette option pour développer une ligne entière dans le graphique.
Créer Rapport de ligne pour la fonction	Sélectionnez cette option pour lancer le Profiler (s'il n'est pas déjà en cours d'exécution), lier immédiatement la fonction sélectionnée et la préparer pour l'enregistrement. Une fois liée, un onglet supplémentaire s'ouvre dans le Rapport Vue actuel. Ce rapport sera mis à jour instantanément, indiquant le nombre de fois que chaque ligne a été exécutée. Bien entendu, le rapport continuera d'enregistrer l'activité dans la fonction même si elle n'est pas visible. Notes : • Dans les programmes fenêtrés, il est généralement nécessaire d'effectuer une action dans l'application pour provoquer l'appel de la fonction. • Cette option ne s'applique pas aux rapports de profil de mémoire
Créer un graphique de fonction	Sélectionnez cette option pour créer un onglet supplémentaire qui affiche la fonction sélectionnée de manière isolée. Pour un profil de fréquence d'appel, cela produit un graphique montrant toutes les lignes qui ont conduit à l'appel de cette fonction (c'est-à-dire les appelants). Pour un profil de mémoire, cela produit un graphique montrant toutes les lignes qui émanent de cette fonction (c'est-à-dire les appelés).
Marquer le cadre initial pour Diagramme Pile d'Appel	Utiliser cette option avant de créer un diagramme Séquence Pile d'Appel pour limiter la longueur de la pile. Lorsque cette option est sélectionnée, le cadre est marqué et son texte est mis en surbrillance. Les cadres situés au-dessus de celui-ci seront alors exclus de tout diagramme Séquence produit. Cette option n'est pas applicable aux rapports de profil de mémoire.
Supprimer la marque	Supprime la marque d'un cadre qui était précédemment marqué comme « Initial ». Cette option n'est pas applicable aux rapports de profil de mémoire.
Créer Diagramme Pile d'Appel	Génère un diagramme Séquence pour une seule pile dans le graphique. Le cadre sélectionné est représenté comme le cadre terminal de la pile. Le cadre initial de la pile est par défaut le nœud racine si aucun cadre « Initial » n'a été marqué. Cette option n'est pas applicable aux rapports de profil de mémoire.
Créer Diagramme Graphique d'Appel pondéré	Génère un diagramme Séquence qui présente une séquence pour chaque branche de pile visible issue de la trame sélectionnée. En développant et en réduisant les nœuds qui vous intéressent, vous pouvez personnaliser le contenu diagramme Séquence à votre guise. Cette option n'est pas applicable aux rapports de profil de mémoire.
Afficher l'utilisation pondérée la plus lourde	Sélectionnez cette option pour afficher la ligne du graphique avec le poids le plus élevé dans laquelle cette fonction apparaît.
Afficher la prochaine utilisation pondérée	Sélectionnez cette option pour accéder à la ligne suivante du graphique où la fonction apparaît. Vous pouvez utiliser la combinaison de touches de raccourci Ctrl+Flèche vers le bas.

Afficher l'utilisation pondérée précédente	Sélectionnez cette option pour accéder à la ligne précédente du graphique où cette fonction apparaît. Vous pouvez également utiliser la combinaison de touches de raccourci Ctrl+Flèche vers le haut.
Importer le code source	Sélectionnez cette option pour importer le code source sélectionné dans le rapport. Cette option n'est pas applicable aux rapports de profil de mémoire.
Ajustement automatique	Lorsque cette option est activée, elle adapte automatiquement les colonnes à la zone d'affichage disponible.
Enregistrer Rapport dans un fichier	Sélectionnez cette option pour afficher la dialogue « Enregistrer sous », vous permettant de choisir où stocker le rapport.
Enregistrer Rapport sur l'artefact	Note : Avant de sélectionner cette option, allez dans la fenêtre Navigateur et sélectionnez le Paquetage ou l'élément sous lequel créer l'élément Artefact. Vous êtes invité à fournir un nom pour le rapport (et l'élément) ; saisissez-le et cliquez sur le bouton OK . L'élément Artefact est créé dans la fenêtre Navigateur , sous le Paquetage ou l'élément sélectionné. Si vous ajoutez l'artefact à un diagramme sous forme de lien simple, lorsque vous double-cliquez sur l'élément, le rapport est rouvert.

Notes

- Si vous ajoutez le rapport du profileur à un élément d'artefact et joignez également un document lié, le rapport du profileur a la priorité et s'affiche lorsque vous double-cliquez sur l'élément ; vous pouvez afficher le document lié à l'aide de l'option de menu contextuel « Modifier le document lié »

Enregistrer Rapport dans Bibliothèque d'Équipe

Vous pouvez enregistrer n'importe quel rapport actuel comme ressource pour une catégorie, un sujet ou un document dans la Bibliothèque d'Équipe . Le rapport peut ensuite être partagé et examiné à tout moment car il est enregistré avec le modèle. Cela vous aide à :

- Conserver un rapport de profilage pour le comparer aux exécutions futures
- Autoriser d'autres personnes à enquêter sur le profil

Accéder

Menu Contexte	Cliquez-droit dans la fenêtre Bibliothèque | Partager la ressource | Rapport Actif Profiler