

# Travailler dans le support

## Répondre aux demandes du 3<sup>ème</sup> niveau



### Objectifs ICT

4.	Préparer et répondre aux demandes du 3 <sup>ème</sup> niveau d'assistance et introduire dans le processus du support
----	--

Au terme de ce chapitre, je suis capable de ...

Gestion des incidents : le rapport d'erreur	
<input type="checkbox"/> oui <input type="checkbox"/> non	Connaître les exigences pour formuler correctement un problème technique et pour la transmission vers le lieu compétent pour le traitement
<input type="checkbox"/> oui <input type="checkbox"/> non	Connaître le processus de support permettant de transmettre les réponses vers le bon destinataire

**Table des matières**

Informations sur le chapitre .....	1
Durée .....	1
Références .....	1
C'est quoi un rapport d'erreur ? .....	2
Collecte de données .....	2
Pourquoi un rapport d'erreur ? .....	5
Rapport de bug dans l'assistance de premier niveau.....	6
Chez Microsoft : .....	6
Dans le monde du logiciel libre : .....	6
En résumé : .....	7
Les rapports de bugs « à l'autre bout » .....	8
Gestion des rapports de bug.....	8
État.....	8
Sévérité .....	9
Catégorie et doublon.....	9
Recherche.....	9

## Informations sur le chapitre

### Durée



#### Durée prévue :

- 3 périodes, avec les exercices

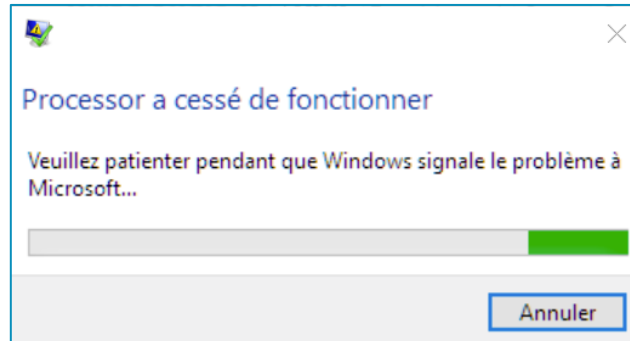
### Références



- <http://www.chiark.greenend.org.uk/~sgtatham/bugs-fr.html>
- <https://bugs.launchpad.net/>
- <http://www.debian.org/Bugs/index.fr.html>
- <http://support.microsoft.com/search/>
- <https://www.lesnumeriques.com/informatique/vie-privee-comment-contrer-indiscretions-windows-10-a2345.html>
- <https://forums.cnetfrance.fr/topic/1261567-windows-10--signaler-un-bug-a-microsoft-via-windows-feedback-ou-insider-hub/>

## C'est quoi un rapport d'erreur ?

Vous avez déjà tous été confrontés au moins une fois à ce genre de fenêtre :

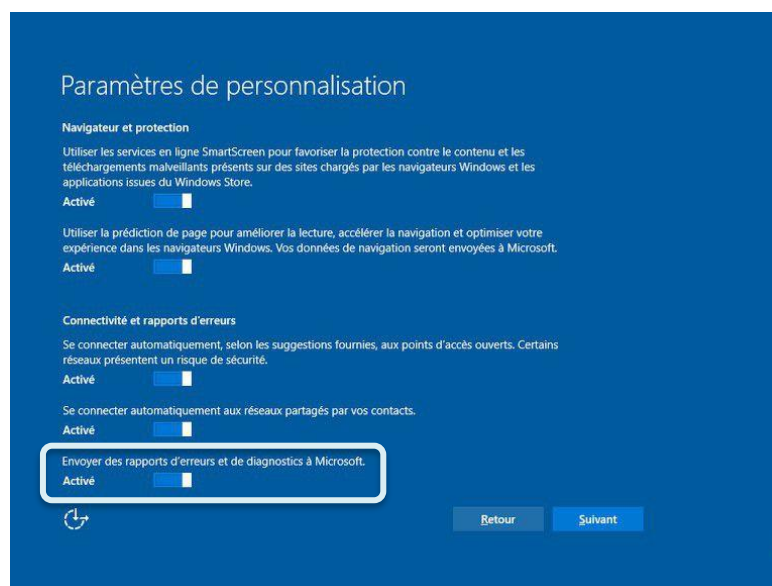


C'est la fenêtre que Windows vous affiche en cas de "plantage" d'une application, derrière se cache l'envoi d'un rapport d'erreurs (signalement du problème).

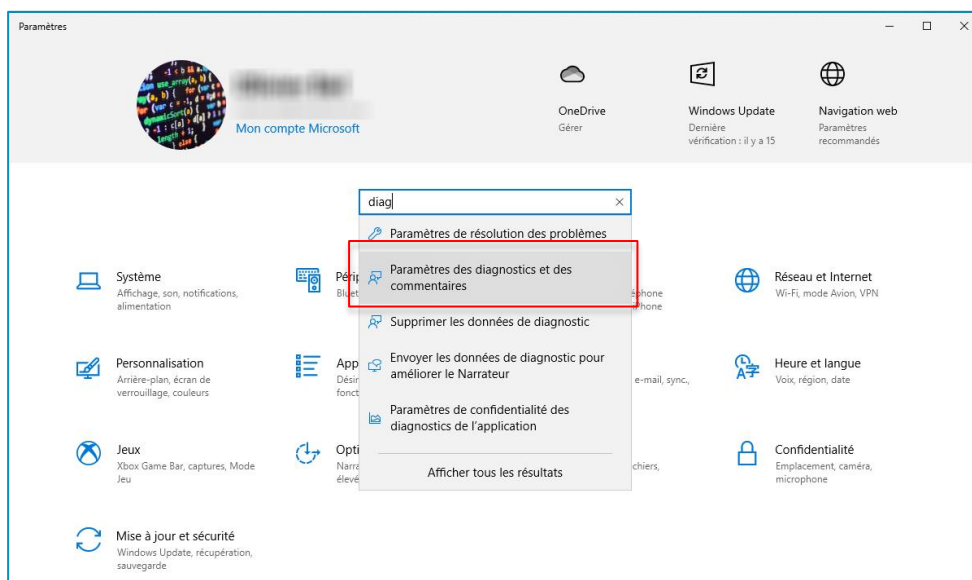
### Collecte de données

Windows 10 a introduit de nombreuses nouveautés, mais aussi une certaine tendance à la curiosité envers votre vie privée. Microsoft n'hésite plus à récupérer beaucoup d'informations présentes sur votre ordinateur afin de faciliter la compréhension de son assistant Cortana ou bien encore pour proposer un ciblage publicitaire performant. Windows ne se cache pas vraiment sur le sujet, tous ces aspects étant détaillés dans la déclaration de confidentialité. Dans les faits, un identifiant publicitaire unique est attribué à chaque compte Microsoft. Comme l'indique la firme de Redmond, cet identifiant est susceptible d'être utilisé par les développeurs d'applications, mais aussi par les "réseaux publicitaires" afin de dresser un profil et d'améliorer la pertinence des publicités.

Il ne faut surtout pas se précipiter lors de l'installation du nouveau système de Microsoft. Lors d'une installation de Windows, une partie qui nous concerne plus particulièrement est l'envoi de rapports d'erreurs et de diagnostics.

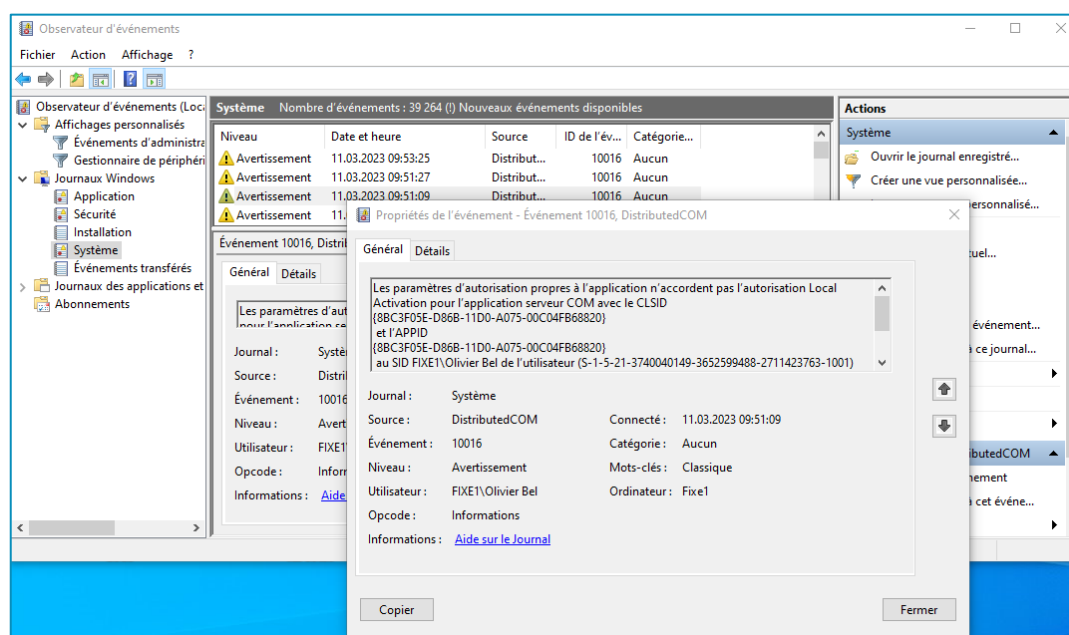


Ou par la suite, *Démarrer, paramètres* puis:



C'est donc un recueil d'informations sur votre machine, votre programme (celui qui a planté) ainsi que sur l'état de votre machine au moment du plantage.

Ouvrons l'Observateur d'événements :



Pour ne pas faire d'erreurs dans ses choix et optimiser la correction de bugs lors du développement de Windows 10, Microsoft a mis en place le programme *Insider* qui permet à tous de devenir des testeurs. Ainsi chaque utilisateur a la possibilité de s'exprimer après une simple inscription au programme.

Dans Windows 10 ce retour d'expérience, qui peut prendre la forme d'une suggestion ou d'un signalement de problème, se fait par l'application "*Hub de commentaires*". Elle vous permet aussi de mettre une note de satisfaction relative à la fonction que vous avez utilisé.



Hub de commentaires  
Application

## Pourquoi un rapport d'erreur ?

Lorsque vous achetez un logiciel et que vous êtes confronté à un dysfonctionnement de ce dernier, vous avez plusieurs manières d'agir :

Dans un monde non-professionnel :

- La méthode la plus courante dans un monde non-professionnel c'est de fermer l'application et de la relancer en espérant que ça n'arrive plus. Souvent cela fonctionne et on oublie très vite ce désagrément.
- Par contre, si le problème persiste, malgré les multiples fermetures et réouvertures du programme, vous aurez recours à de l'assistance de premier niveau soit en téléphonant à une hotline (ou à un ami) ou alors, si votre travail n'est pas urgent ou si vous n'avez pas d'ami informaticien, vous pourrez envoyer un rapport de bug.
- Mais encore, si vous téléphonez à une hotline et que malgré toute la bonne volonté du technicien au bout du fil, vous ne parvenez pas à résoudre le problème, vous serez alors amené à envoyer le rapport d'erreurs.

Dans un monde professionnel :

- Vous ne pouvez pas laisser « traîner » un bug dans une application. Imaginez un technicien utilisant un logiciel informatique permettant la régulation de température d'une centrale nucléaire. Si cette personne voit apparaître un rapport d'erreurs, vous imaginez bien qu'il ne va pas simplement fermer l'application en espérant qu'à la réouverture le message aura disparu ! La première chose que ce technicien fera, après avoir enclenché une alarme, sera d'appeler le responsable du logiciel pour qu'il traite au plus vite le problème. Et le responsable du logiciel recevra comme du pain béni le rapport d'erreurs contenant le plus d'informations possibles.
- Dans un chapitre moins dramatique, si un employé de banque utilisant un logiciel de transfert de fond voit son programme planter en affichant un rapport d'erreurs, pensez-vous qu'il va fermer l'application puis la réouvrir en espérant que le programme ne plantera plus et que le transfert c'est bien déroulé !? Lui aussi prendra immédiatement contact avec le responsable du logiciel qui a son tour recevra comme du pain bénis le rapport d'erreurs.

Dans le monde professionnel comme dans le monde non-professionnel, le rapport d'erreurs est capital lorsqu'il s'agit de déboguer un logiciel.



*Microsoft nous a habitués depuis longtemps à attendre les patches de mises à jour ainsi que les services packs. Mais pour pouvoir créer les patches, encore faut-il être au courant des bugs.*

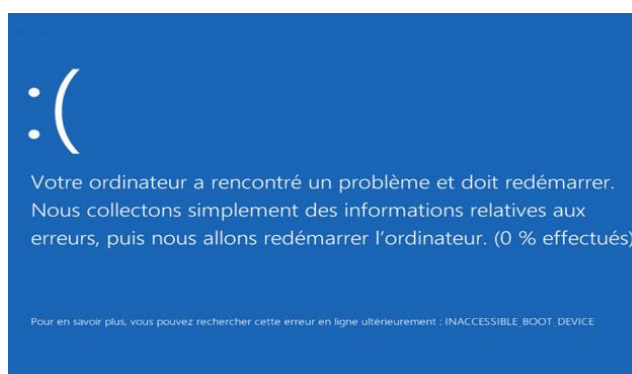
*Et ça, c'est le rôle du rapport d'erreurs !*

## Rapport de bug dans le processus d'assistance

Le rapport d'erreurs, c'est une autre manière de demander de l'aide de premier niveau à une personne ou à une entreprise. C'est une manière de dire : « mon programme ne fonctionne pas, aidez-moi ! ». Cela reste du premier niveau car ça n'implique pas le déplacement d'un technicien. Vous envoyez un email ou alors vous postez votre erreur sur un site dédié.

### Chez Microsoft :

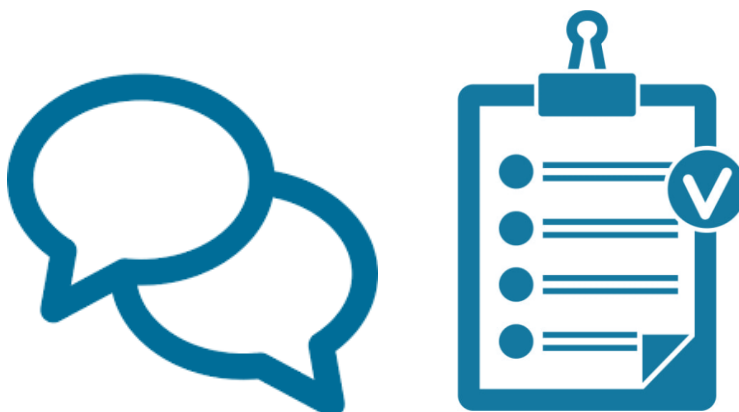
Cette demande se fait automatiquement.



### Dans le monde du logiciel libre :

Le dialogue entre l'utilisateur et le développeur est monnaie courante. Bien sûr, ce dialogue peut s'instaurer pour complimenter le développeur mais la plupart du temps, ce dialogue se crée pour une demande d'aide. Soit l'utilisateur ne comprend pas le fonctionnement du programme soit le programme plante ou encore, l'utilisateur pense que le programme plante ou qu'il ne fait pas ce qu'il devrait. Dans tous les cas, il établit un contact avec le développeur pour obtenir une aide.

Dans le cas du programme qui plante, le développeur aimerait pouvoir voir le programme en train de planter. Malheureusement, c'est rarement possible. C'est alors que le rapport d'erreurs prend toute son importance. C'est dans ce rapport que figureront toutes les informations nécessaires au développeur pour pouvoir reproduire au mieux le bug chez lui. Ainsi il pourra se mettre dans les mêmes conditions que vous pour essayer de faire planter le programme chez lui comme il l'a fait chez vous. Et s'il n'y parvient pas, alors, il pourra fouiller dans ce rapport à la recherche du détail qui lui indiquera peut-être la source du problème.





## En résumé :



- *Le premier objectif d'un rapport de bug est de permettre au programmeur de voir le bug de ses propres yeux. Si vous ne pouvez le faire planter devant lui, donnez-lui des instructions détaillées afin qu'il puisse le faire planter par lui-même*
- *Si le premier objectif ne peut être atteint, et donc si le programmeur ne peut voir l'erreur se produire, le second objectif d'un rapport de bug est de décrire ce qui s'est mal passé. Décrivez tout, en détails. Dites ce que vous avez vu, mais dites aussi ce que vous vous attendiez à voir. Recopiez les messages d'erreur, surtout s'ils contiennent des nombres*
- *Quand votre ordinateur fait quelque chose d'inattendu, arrêtez-vous. Ne faites rien tant que vous n'êtes pas calme, et ne faites rien que vous pensez pouvoir être dangereux*
- *Bien entendu, si vous pensez en être capable, diagnostiquez l'erreur par vous-même, mais si vous le faites, vous devriez tout de même lui communiquer les symptômes*
- *Soyez prêt à fournir des informations supplémentaires si le programmeur en a besoin. S'il pouvait s'en passer, il ne vous les demanderait pas. Il ne vous dérange pas pour le plaisir*
- *Écrivez clairement. Dites ce que vous voulez dire, et soyez sûr que ça ne peut pas être mal compris*
- *Et surtout, soyez précis : les programmeurs aiment la précision*

## Les rapports de bugs « à l'autre bout »

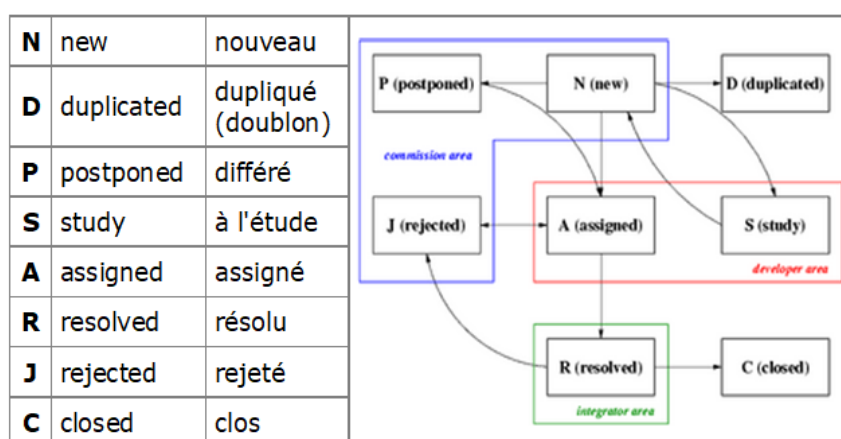
### Gestion des rapports de bug

Tous les bugs doivent être référencés de façon unique dans une base de données sous forme de rapports de bug numérotés. Le rapport de bug est un enregistrement contenant différents champs :

- Un identifiant unique (numéro)
- Un titre
- Une description
- Un état
- Une sévérité
- Une catégorie
- Un numéro de livraison
- etc.

### État

Un rapport de bug a différents états en fonction de l'avancement de son traitement ; ceux-ci doivent suivre le cycle de vie du rapport défini sous forme de diagramme :



- À l'état N est associé le nom (login) du créateur
- Aux états S et A sont associés le nom d'un développeur ou correcteur
- Une commission de traitement des rapports de bug s'occupe des bugs dans les états N, D et P, et effectue l'assignation, l'étude ou le rejet
- Une autre commission dite d'intégration s'occupe des bugs dans les états R et J et décide la clôture

Attention à ne pas confondre l'état d'un rapport et l'action à effectuer induisant **un changement d'état**, comme par l'exemple l'erreur qui consiste à utiliser des états de transition. Cette confusion mène souvent à des cycles de vie de rapports de bug lourds, compliqués et inefficaces.

**Notifications par courrier électronique** : tout changement d'état d'un rapport de bug doit être signalé par e-mail au créateur du rapport et au correcteur de celui-ci lorsqu'il y travaille. Les modifications du contenu du rapport peuvent également être notifiées.

### Sévérité

Chaque rapport de bug doit également se voir affecté une sévérité, de la valeur 1 pour bloquant à 5 pour une modification sans conséquence fonctionnelle, par exemple.

### Catégorie et doublon

Ainsi, ajouter une étiquette (champ) pour classer le bug dans une catégorie selon sa nature : hardware, software, client, server, etc. Cette étiquette doit s'utiliser comme un masque ou une liste déroulante.

Enfin, une recommandation élémentaire, mais souvent oubliée, est qu'un **rapport de bug ne doit traiter que d'un seul problème**. Il faut prendre la peine d'écrire deux rapports distincts pour deux problèmes connexes mais différents ; l'un pourra être résolu et vérifié indépendamment de l'autre.

### Recherche

Enfin, il apparaît fondamental de pouvoir accéder aux bases de tous les composants logiciels utilisés (génériques ou spécifiques) et pouvoir y faire des recherches par mots-clés.



#### Remarque :

*Dans le monde du logiciel libre, l'augmentation du nombre de personnes utilisant ces logiciels augmente considérablement le nombre de doublons dans les bases de données. Malgré des mécanismes de traitement automatisés pour la recherche de ces doublons, l'état D (duplicated) subsiste toujours car les rapports d'erreurs étant fait manuellement, il est parfois difficile de reconnaître le même bug au travers de deux descriptions totalement différentes.*