



Genèse du produit	
Fonctionnalités du produit	
Atouts du produit	
Evolution du produit	
Architecture du produit	
Installation du produit sous Debian	
Configuration du produit	
Déploiement du produit	
Exploitation du produit	
Administration du produit	

1. Genèse du produit

MantisBT a été créé en 2000 par Kenzaburo Ito. Initialement, il s'agissait d'un projet personnel conçu pour répondre à ses besoins en matière de gestion des bogues. Rapidement, MantisBT est devenu un projet open source grâce à la contribution d'une communauté croissante de développeurs. Le nom "Mantis" est une référence à la mante religieuse, symbolisant la vigilance, tandis que "BT" signifie Bug Tracker.

2. Fonctionnalités du produit

MantisBT offre un ensemble de fonctionnalités adaptées à la gestion des bugs et au suivi de projets, notamment :

- Suivi des bugs et des incidents : Création, assignation et suivi des bugs.
- Gestion des utilisateurs : Gestion des rôles et permissions, avec des niveaux d'accès configurables.
- Collaboration d'équipe : Notifications par e-mail et journaux d'activités.
- Support multi-projet : Permet de gérer plusieurs projets au sein de la même instance.
- Personnalisation : Plugins, thèmes et configuration des workflows.
- Rapports et statistiques : Visualisation des données via graphiques et rapports exportables.
- Intégration avec d'autres outils : Support d'API REST/ SOAP et intégration avec des systèmes tels que Git, Jenkins ou Slack.

3. Atouts du produit

- Open source : Gratuit, avec une communauté active pour le support et les mises à jour.
- Simplicité : Interface utilisateur claire et intuitive.
- Flexibilité : Hautement personnalisable grâce à sa structure modulaire.
- Accessibilité web : Fonctionne via un navigateur, sans besoin d'installation locale.
- Large adoption : Utilisé par des entreprises de toutes tailles et dans divers secteurs.

4. Évolution du produit

Depuis sa création :

- Des fonctionnalités avancées ont été ajoutées, comme l'API REST, la gestion des workflows personnalisés et des rapports graphiques.
- Des améliorations régulières ont été apportées pour assurer la compatibilité avec les dernières technologies (PHP, MySQL, PostgreSQL).
- La communauté a introduit de nombreux plugins, enrichissant ainsi les fonctionnalités natives.
- MantisBT est passé de versions principalement centrées sur les bugs à une plateforme complète de suivi de projets.

5. Architecture du produit

MantisBT repose sur une architecture simple mais robuste :

- Backend :
 - Écrit en PHP.
 - Supporte les bases de données MySQL, MariaDB, PostgreSQL et autres.
- Frontend :
 - Génère des pages HTML accessibles via un navigateur.
 - Utilise CSS et JavaScript pour améliorer l'expérience utilisateur.
- API REST/SOAP : Fournit des interfaces pour l'intégration avec des outils tiers.
- Structure modulaire : Facilite l'ajout de plugins et la personnalisation.
- Sécurité :
 - Gestion des droits et permissions.
 - Authentification sécurisée via mots de passe ou plugins d'authentification (LDAP, OAuth).

- 6. Installation du produit

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install apache2 mariadb-server php libapache2-mod-php php-{mysql,mbstring,curl,
```

```
sudo mysql_secure_installation
```

```
sudo mysql -u root -p
```

```
CREATE DATABASE mantisdb;  
GRANT ALL PRIVILEGES ON mantisdb.* TO 'mantisuser'@'localhost' IDENTIFIED BY 'mantispass';  
FLUSH PRIVILEGES;  
EXIT;
```

```
wget https://sourceforge.net/projects/mantisbt/files/mantis-stable/2.25.7/mantisbt-2.25.7.
```

```
unzip mantisbt-2.25.7.zip  
sudo mv mantisbt-2.25.7 /var/www/html/mantisbt
```

```
sudo chown -R www-data:www-data /var/www/html/mantisbt  
sudo chmod -R 755 /var/www/html/mantisbt
```

```
sudo nano /etc/apache2/sites-available/mantisbt.conf
```

```
<VirtualHost *:80>  
    ServerAdmin webmaster@votre_domaine  
    DocumentRoot /var/www/html/mantisbt  
    ServerName votre_domaine  
  
    <Directory /var/www/html/mantisbt>  
        Options -Indexes +FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/mantisbt_error.log  
    CustomLog ${APACHE_LOG_DIR}/mantisbt_access.log combined  
</VirtualHost>
```

```
sudo a2ensite mantisbt.conf
sudo a2enmod rewrite
sudo systemctl restart apache2
```

```
sudo rm -rf /var/www/html/mantisbt/admin
```

3. Déploiement du produit

Cette étape vise à rendre le produit accessible aux utilisateurs finaux. Le déploiement implique plusieurs actions :

Préparer l'infrastructure :

Si vous déployez sur un serveur physique ou dans le cloud, assurez-vous que les ressources nécessaires (serveur, mémoire, stockage) sont disponibles. Si nécessaire, provisionnez des serveurs via des fournisseurs comme AWS, Google Cloud, ou Azure.

Déploiement de l'application :

Téléchargez les fichiers du produit sur le serveur cible. Vous pouvez utiliser des outils comme FTP, SSH, ou Git pour déployer les fichiers.

```
git clone https://github.com/votre-repository/mentis.git
cd mentis
git checkout main
```

Base de données :

- Importez la base de données sur le serveur de production, en utilisant des outils comme `mysqldump` ou des migrations automatisées via un framework (par exemple, avec Laravel ou Symfony).

Configurer le serveur :

- Configurez le serveur web pour pointer vers le répertoire où se trouvent les fichiers du produit. Exemple avec Nginx :

```
server {
    listen 80;
    server_name votre-domaine.com;
    root /var/www/mentis;
    index index.php;
    location / {
        try_files $uri $uri/ /index.php?$args;
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

1. Tests en production :

- Avant de rendre le produit accessible à tous les utilisateurs, effectuez des tests pour vérifier que tout fonctionne correctement dans l'environnement de production. Cela inclut des tests de charge, des tests fonctionnels et des tests de sécurité.

Exploitation du produit

Une fois le produit déployé, il faut le surveiller et l'exploiter.

1. Surveillance :

- Utilisez des outils de surveillance pour vérifier la santé du système (ex. : New Relic, Prometheus, ou des solutions basées sur des logs comme ELK Stack).
- Suivez les erreurs système, les alertes de sécurité, et les temps de réponse des services.

2. Maintenance :

- Effectuez des mises à jour régulières pour corriger les bugs, sécuriser les vulnérabilités, et améliorer les performances.
- Planifiez des interventions régulières pour mettre à jour les dépendances et faire des ajustements.

3. Support utilisateurs :

- Mettez en place une équipe de support technique pour répondre aux demandes des utilisateurs, résoudre les problèmes et gérer les retours sur le produit.

Administration du produit

L'administration du produit implique une gestion continue de son infrastructure et de ses utilisateurs.

1. Gestion des utilisateurs :

- Créez des interfaces administratives permettant de gérer les utilisateurs, leur rôle, et leurs permissions. Cela peut inclure la possibilité de créer, modifier ou supprimer des comptes utilisateurs, ainsi que d'attribuer des rôles (admin, utilisateur, etc.).

2. Sécurité et mises à jour :

- Assurez-vous que les mises à jour de sécurité sont appliquées rapidement, que les accès aux systèmes sont contrôlés, et que des audits réguliers sont effectués.
- Vérifiez les logs d'activité pour détecter toute tentative d'intrusion ou utilisation anormale du système.

3. Optimisation continue :

- Analysez régulièrement la performance du produit et effectuez des optimisations si nécessaire (amélioration de la vitesse, réduction de la consommation de ressources, etc.).
- Réalisez des analyses de données pour déterminer comment les utilisateurs interagissent avec le produit et identifiez les opportunités d'amélioration.

Résultat :



Anonymous | [Login](#) | [Signup for a new account](#)

2009-02-01 03:44 EST



[Main](#) | [My View](#) | [View Issues](#) | [Change Log](#) | [Roadmap](#) | [Docs](#) | [Wiki](#) | [ManTweet](#) | [Repositories](#)

Unassigned [^] (1 - 10 / 2006)

0010092 —	Logging of user action (login, download, ...) feature - 2009-01-30 09:55
0009394 — @	MantisConnect Webservice crashes when trying to get issue information api soap - 2009-01-30 07:43
0010094 —	Email generation if not assigned bugtracker - 2009-01-30 07:02
0010091 —	CHANGE STATUS TO: Assigned vs ASSIGN TO ?? customization - 2009-01-29 23:37
0004640 —	New custom field types: "Version", "User" custom fields - 2009-01-29 22:52
0003790 — @	Additional Custom-Field-Type "users" custom fields - 2009-01-29 22:49
0010065 —	Custom field "date" not saved silently custom fields - 2009-01-29 08:22
0006418 —	Change Resolve status and Assignment at the same time feature - 2009-01-29 05:41
0007104 —	Auto assign on resolve bugtracker - 2009-01-29 05:39
0007150 —	Automatic reassignment on status transition feature - 2009-01-29 05:39

Resolved [^] (1 - 10 / 254)

0010093 —	(phpmailer) class.phpmailer.php => private \$smtp is accessed by mantis email - 2009-01-31 14:37
0010056 — @	Category national character upgrade - 2009-01-30 14:30
0009455 —	Database configuration does not correctly deal with array administration - 2009-01-30 11:18
0010080 —	Updating version field does not trigger history note administration - 2009-01-28 03:55
0008628 —	The "Show Content" link on attached tx file doesn't show text webpage - 2009-01-28 03:47
0007790 — @	Links protected by brackets are not processed properly bugtracker - 2009-01-27 12:27
0009216 — @	Problems with Russian language localization - 2009-01-27 11:40
0007144 —	Unable to set realname because of existing username administration - 2009-01-25 14:21
0010078 —	logging into the site with out entering login details security - 2009-01-24 12:44
0010073 —	Form Validate in JavaScript javascript - 2009-01-22 14:43