

Advanced Printer Driver 6

Manuel de Status API

Avant utilisation

Décrit les informations que les utilisateurs doivent connaître avant d'utiliser ce produit.

Aperçu d'APD6

Fournit un aperçu d'APD6.

Utilisation de Status API

Explique comment établir un environnement de développement, obtenir les statuts ASB et comment gérer les statuts ASB.

Référence pour Win32

Décrit les informations de Status API utilisées dans l'environnement Win32.

Référence pour .NET

Décrit les informations de Status API utilisées dans l'environnement .NET.



Avant utilisation

Ce chapitre décrit les informations que les utilisateurs doivent connaître avant d'utiliser l'EPSON Advanced Printer Driver 6 (désigné ci-après par "APD6").

Packages APD6

APD6 se compose des packages suivants.

- **Package du pilote d'imprimante**
Ces packages sont préparés pour chaque modèle d'imprimante TM. L'installation du pilote d'imprimante permet une impression facile à partir d'applications logicielles. Les manuels suivants sont fournis.
 - **Manuel d'installation**
Cette section décrit l'installation d'APD6, l'enregistrement de l'imprimante TM et comment installer automatiquement le pilote d'imprimante.
 - **Manuel de l'imprimante**
Cette section décrit les procédures de configuration et les fonctions d'APD6.
 - **Spécifications de l'imprimante**
Cette section décrit les spécifications du pilote d'imprimante pour chaque modèle d'imprimante TM.
- **Package de Status API**
Il s'agit d'un package spécial dans APD6 pour toutes les imprimantes TM. Il doit être installé lors du développement d'applications qui contrôlent les imprimantes TM à l'aide de Status API, lors de la surveillance de l'état d'une imprimante en temps réel, et lorsque APD6 coexiste avec d'autres pilotes Epson. Les manuels suivants sont fournis.
 - **Manuel de Status API (ce manuel)**
Cette section décrit comment utiliser Status API pour obtenir l'état d'une imprimante TM à partir d'une application logicielle. Pour les spécifications des API disponibles pour chaque modèle d'imprimante TM, consultez le manuel "Spécifications de l'imprimante" contenu dans le package du pilote d'imprimante.
- **Package du programme échantillon**
Il s'agit d'un package spécial dans APD6 pour toutes les imprimantes TM. Il contient des programmes échantillons et un code échantillon pour le développement d'applications pour l'impression et le contrôle des imprimantes TM. Bien qu'aucun manuel ne soit fourni, il contient des fichiers HTML qui décrivent les programmes.

Télécharger

Pour les clients d'Amérique du Nord : visitez le site web suivant :

<https://www.epson.com/support/>

Pour les clients d'autres pays : visitez le site web suivant :

<https://epson.sn>

Signification des symboles

**REMARQUE**

Indique des explications supplémentaires et des informations que vous devez connaître.

Table des matières

Avant utilisation	2
■ Packages APD6	2
Télécharger	2
■ Signification des symboles.....	3
■ Table des matières	4

Aperçu d'APD6.....	7
■ Fonctionnalités d'APD6.....	7
Glossaire	7
■ Environnement d'exploitation	8
Logiciel nécessitant l'installation du package Status API.....	8
Imprimantes TM prises en charge	8
■ Langage de développement.....	9
■ Installation du package Status API	9
Désinstallation du package	9

Utilisation de Status API.....	10
■ Architecture de l'environnement de développement	10
■ Types de fonctions de Status API	13
■ Flux de programmation pour l'acquisition du statut ASB.....	15
Visual C++	15
Visual C# / Visual Basic .NET	17
■ Flux de programmation pour le contrôle MICR.....	18
Visual C++	18
Visual C# / Visual Basic .NET	19
■ Erreurs et réponse de Status API	20
Statut ASB	20
Erreur d'exécution de Status API	23

Référence pour Win32	25
■ BiOpenMonPrinter	25
■ BiCloseMonPrinter	27

■ BiLockPrinter	28
■ BiUnlockPrinter	30
■ BiDirectIOEx	31
■ BiResetPrinter	34
■ BiForceResetPrinter	35
■ BiGetType	36
■ BiGetStatus.....	37
■ BiSetStatusBackFunction	38
■ BiSetStatusBackFunctionEx	39
■ BiCancelStatusBack	40
■ BiGetBatteryStatus	41
■ BiSetBatteryStatusBackFunctionEx.....	42
■ BiCancelBatteryStatusBack.....	43
■ BiPowerOff	44
■ BiAutoPowerOffTime.....	45
■ BiGetPrnCapability	46
■ BiOpenDrawer	47
■ BiMICRCleaning.....	49
■ BiMICRGetStatus	50
■ BiMICRLoadCheck.....	51
■ BiMICRReadCheck.....	52
■ BiMICREjectCheck.....	54
■ BiMICRCancelWaitCheckInsertion	55
■ BiMICRRetransmissionCheckData	56
■ BiMICRSelectDataHandling	58
■ BiMICRSetReadBackFunction	60
■ BiMICRSetReadBackFunctionEX	61
■ BiMICRCancelReadBack	63

Référence pour .NET64

■ Propriétés	64
IsValid	64
LastError.....	64
État.....	64
AutoPowerOffTime.....	65
■ Méthodes.....	66
OpenMonPrinter.....	66
CloseMonPrinter.....	66

LockPrinter.....	66
UnlockPrinter.....	67
DirectIOEx.....	67
ResetPrinter.....	68
ForceResetPrinter.....	68
GetType.....	68
SetStatusBack.....	69
CancelStatusBack.....	69
GetBatteryStatus.....	69
SetBatteryStatusBack.....	69
CancelBatteryStatusBack.....	70
PowerOff.....	70
GetPrnCapability.....	70
OpenDrawer.....	71
MICRCleaning.....	71
MICRGetStatus.....	71
MICRLoadCheck.....	72
MICRReadCheck.....	72
MICREjectCheck.....	72
MICRCancelWaitCheckInsertion.....	72
MICRRetransmissionCheckData.....	73
MICRSelectDataHandling.....	73
MICRSetReadBack.....	74
MICRCancelReadBack.....	74
■ Événements.....	75
StatusCallback.....	75
StatusCallbackEx.....	75
BatteryStatusCallbackEx.....	75
MICRReadCallback.....	76
MICRReadCallbackEx.....	76
Annexe.....	77
■ Mentions.....	77
■ Mises en garde.....	78
■ Marques.....	78

Aperçu d'APD6

Fonctionnalités d'APD6

L'EPSON Advanced Printer Driver Version 6 (ci-après désigné par "APD6") est un pilote d'imprimante Windows pour les imprimantes Epson TM. En installant le package de Status API, vous pouvez utiliser les fonctions suivantes.

- Vous pouvez partager votre imprimante TM avec des applications qui exécutent d'autres pilotes Epson (tels qu'OPOS). Le contrôle exclusif des applications est effectué automatiquement par le pilote, c'est pourquoi les applications n'ont pas besoin de l'exécuter.
- Le Status API de l'imprimante TM vous permet d'acquérir des informations sur le modèle d'imprimante TM à partir d'une application particulière, de contrôler l'imprimante TM et d'acquérir des informations sur l'état de l'imprimante TM.
- Vous pouvez également utiliser les polices de périphérique de l'imprimante TM pour imprimer dans les applications de l'environnement .NET.



REMARQUE

Pour l'APD6 et versions ultérieures, le package Status API est distribué individuellement.

Glossaire

Termes	Explication
Statut ASB	Retour automatique de l'état : il s'agit d'une fonction de l'imprimante TM. Il s'agit d'un état envoyé automatiquement par l'imprimante lorsque son statut change (ouverture ou fermeture du capot, plus de papier, impression terminée, etc.).

Environnement d'exploitation

Voir le "Manuel d'installation" pour l'APD6.

Logiciel nécessitant l'installation du package Status API

Installez le package Status API dans les cas suivants.

- Lors de l'utilisation de Status API pour développer une application qui contrôle l'imprimante TM
- Lorsque vous utilisez l'une des applications logicielles Epson suivantes avec APD6 sur le même ordinateur

Nom du logiciel	Version compatible
EPSON Advanced Printer Driver 4	Ver.4.56 ou version ultérieure
EPSON Advanced Printer Driver 5	Ver.5.09 ou version ultérieure
EPSON OPOS ADK	Ver.2.68 ou version ultérieure
EPSON OPOS ADK for .NET	Ver.1.11.20 ou version ultérieure
EPSON JavaPOS ADK	Ver.1.11.20 ou version ultérieure
EpsonNet SimpleViewer	Ver.2.30 ou version ultérieure
TM Virtual Port Driver	Ver.7.10a ou version ultérieure

Imprimantes TM prises en charge

Il s'agit des imprimantes TM prises en charge par l'APD6 actuellement installé. Pour plus de détails, consultez le "Manuel de spécifications de l'imprimante TM" inclus dans le package du pilote d'imprimante actuellement installé.

Langage de développement

Win32

- Visual C++

.NET

- Visual Basic .NET
- Visual C#

Installation du package Status API

Suivez la procédure décrite ci-dessous pour installer le package Status API.

- 1** Double-cliquez sur l'installateur du package Status API (APD6_StatusAPI_x.exe) pour démarrer le processus d'installation.
- 2** Suivez les instructions à l'écran pour poursuivre le processus d'installation.
- 3** Si le package Status API a été installé après l'installation du package du pilote d'imprimante et l'enregistrement de l'imprimante, exécutez l'impression afin de changer le port de communication.

Ceci termine le processus d'installation du package Status API.

Désinstallation du package

Voir le "Manuel d'installation" pour les instructions sur la désinstallation de Status API. Un package Status API n'est jamais désinstallé individuellement.

Utilisation de Status API

Ce chapitre présente l'architecture de l'environnement de développement d'applications utilisant Status API et la méthode de programmation.

Architecture de l'environnement de développement

L'architecture de l'environnement de développement d'applications utilisant Status API diffère selon l'outil de développement.

Visual C++: [page 10](#)

Visual Basic .NET: [page 11](#)

Visual C#: [page 12](#)



REMARQUE

Ceci décrit les méthodes pour Visual Studio 2017.

Visual C++

Voici des exemples de l'architecture de l'environnement de développement utilisant C++.

- 1** Démarrez Microsoft Visual C++ pour ouvrir l'Explorateur de solutions.
- 2** Copiez EpsStmApi.h dans le dossier installé avec APD et collez-le dans le dossier d'exploitation utilisé lors du développement d'applications (dossier créé par le projet).
 Enregistrer comme fichier
 OS 32 bits : "C:\Programfiles\Epson\Advanced Printer Tool\StatusAPI"
 OS 64 bits : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
 OS ARM64 : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
- 3** Ouvrez le fichier source. Définissez EpsStmApi.h à l'aide de la directive #include.
 Méthodes de définition : **#include "EpsStmApi.h"**
- 4** L'environnement Visual C++ est prêt pour le développement d'une application utilisant Status API.

Visual Basic .NET

Voici un exemple de création d'un environnement de développement à l'aide de Visual Basic .NET.

- 1** Démarrez Microsoft Visual Basic.NET pour ouvrir l'Explorateur de solutions.
- 2** Faites un clic droit sur [Références] dans l'Explorateur de solutions, et sélectionnez [Ajouter des références].



REMARQUE

Si l'élément [Références] n'apparaît pas, cliquez sur l'icône [Afficher tous les fichiers] dans l'Explorateur de solutions.

- 3** L'écran "Ajouter des références" apparaît. Cliquez sur l'onglet [Parcourir].
- 4** Spécifiez ce qui suit dans [Rechercher dans].
 OS 32 bits : "C:\Programfiles\Epson\Advanced Printer Tool\StatusAPI"
 OS 64 bits : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
 OS ARM64 : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
- 5** Tapez le nom de fichier "EpsonStatusAPI.dll", puis cliquez sur [OK].
- 6** Sélectionnez [Références] - [EpsonStatusAPI] dans l'Explorateur de solutions, et sélectionnez "Faux" pour [Version spécifique] dans Propriétés.
- 7** En utilisant l'instruction Imports au tout début du code source, décrivez comme suit.

```
Imports com.epson.pos.driver
```
- 8** L'environnement Visual Basic.NET est prêt pour le développement d'une application utilisant Status API.

Visual C#

Voici un exemple de création de l'environnement de développement à l'aide de Visual C#.

- 1** Démarrez Microsoft Visual C# pour ouvrir l'Explorateur de solutions.
- 2** Faites un clic droit sur [Références] dans l'Explorateur de solutions, et sélectionnez [Ajouter des références].



REMARQUE

Si l'élément [Références] n'apparaît pas, cliquez sur l'icône [Afficher tous les fichiers] dans l'Explorateur de solutions.

- 3** L'écran "Ajouter des références" apparaît. Cliquez sur l'onglet [Parcourir].
- 4** Spécifiez ce qui suit dans [Rechercher dans].
 OS 32 bits : "C:\Programfiles\Epson\Advanced Printer Tool\StatusAPI"
 OS 64 bits : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
 OS ARM64 : "C:\Program Files(x86)\Epson\Advanced Printer Tool\StatusAPI"
- 5** Tapez le nom de fichier "EpsonStatusAPI.dll", puis cliquez sur [OK].
- 6** Sélectionnez [Références] - [EpsonStatusAPI] dans l'Explorateur de solutions, et sélectionnez "Faux" pour [Version spécifique] dans Propriétés.
- 7** En utilisant le mot-clé au tout début du code source, décrivez comme suit.
using com.epson.pos.driver
- 8** L'environnement Visual C# est prêt pour le développement d'une application utilisant Status API.

Types de fonctions de Status API

Status API offre les fonctions suivantes. Reportez-vous à "[Référence pour Win32](#)" à la page 25 pour plus de détails concernant les fonctions.



REMARQUE

Les fonctions prises en charge diffèrent selon le modèle d'imprimante. Reportez-vous aux "Spécifications de l'imprimante" pour plus de détails concernant chaque modèle.

Application	Fonction	Description
Début/Fermeture Status API	BiOpenMonPrinter	Sollicite l'imprimante spécifiée pour utiliser Status API.
	BiCloseMonPrinter	Ferme Status API.
Utilisation de l'imprimante TM	BiLockPrinter	Utilise l'imprimante TM. Lorsqu'elle est occupée, l'imprimante n'accepte aucune API provenant d'autres processus.
	BiUnlockPrinter	Annule BiLockPrinter.
Acquisition du statut ASB	BiGetStatus	Acquiert le statut ASB lorsque l'application le requiert.
	BiSetStatusBackFunction	Fournit une notification concernant l'appel de la fonction de rappel notifiant l'application lorsque le statut ASB de Status API change.
	BiSetStatusBackFunctionEx	Fournit une notification concernant l'appel de la fonction de rappel notifiant à l'application lorsque le statut ASB de Status API change. Acquiert également le nombre de ports.
	BiCancelStatusBack	Annule la fonction de notification automatique du statut. Cette fonction est applicable à BiSetStatusBackFunction et BiSetStatusBackFunctionEx.
Acquisition de l'état de la batterie	BiGetBatteryStatus	Acquiert l'état actuel de la batterie de l'imprimante lorsque l'application le requiert.
	BiSetBatteryStatusBackFunctionEx	Acquiert automatiquement le statut de l'imprimante (statut ASB) à l'aide de la fonction de rappel lorsque le statut de l'imprimante change. Identifie le port de l'imprimante à l'origine du rappel, en plus des fonctions de BiSetStatusBackFunction.
	BiCancelBatteryStatusBack	Annule le processus de demande de notification de statut automatique appelé à l'aide de BiSetBatteryStatusBackFunctionEx.
Acquisition des informations de l'imprimante	BiGetType	Acquiert les informations de l'imprimante TM, telles que la présence du capteur BM et la connexion de l'affichage client.
	BiGetPrnCapability	Acquiert les informations de l'imprimante, c'est-à-dire le micrologiciel, etc.
Contrôle du tiroir	BiOpenDrawer	Ouvre le tiroir.

Application	Fonction	Description
Réinitialisation de l'imprimante	BiResetPrinter	Réinitialise les imprimantes à interface parallèle / USB / ethernet. Impossible de réinitialiser les imprimantes à interface série.
	BiForceResetPrinter	Peut réinitialiser également les imprimantes TM occupées avec BiLockPrinter.
Prétraitement de désactivation	BiPowerOff	Définit le mode de désactivation ou de veille. Les processus suivants sont exécutés : • Configure l'interface sur BUSY. • Configure l'imprimante en mode veille.
Gestion de l'alimentation de l'imprimante mobile	BiAutoPowerOffTime	Acquiert et définit le temps restant avant que l'imprimante alimentée par batterie ne tombe en panne. Il ne s'agit pas d'une API servant à éteindre l'imprimante.
Envoie les commandes ESC/POS commande	BiDirectIOEx	Peut envoyer et recevoir les commandes ESC/POS. N'ajoute pas la commande de suppression ASB.
Contrôle de la fonction MICR	BiMICRCleaning	Nettoie le mécanisme MICR.
	BiMICRGetStatus	Acquiert le statut MICR lorsque l'application le requiert.
	BiMICRLoadCheck	Charge le chèque à la position de début d'impression du chèque.
	BiMICRReadCheck	Exécute la lecture de chèque.
	BiMICREjectCheck	Éjecte le chèque.
	BiMICRCancelWaitCheckInsertion	Annule l'attente d'insertion de chèque.
	BiMICRRetransmissionCheckData	Renvoie les résultats de lecture de chèque.
	BiMICRSelectDataHandling	Sélectionne l'opération de lecture de chèque.
	BiMICRSetReadBackFunction	Exécute la lecture de chèques par BiMICRReadCheck et enregistre l'adresse de la fonction de rappel.
	BiMICRSetReadBackFunctionEx	Enregistre les adresses mémoire où les informations lues sont définies.
	BiMICRCancelReadBack	Annule une demande de notification d'informations de lecture qui avait été enregistrée.

Flux de programmation pour l'acquisition du statut ASB

Dans cette section, la méthode d'obtention du statut ASB à l'aide de Status API est expliquée à l'aide du diagramme de séquence.



REMARQUE

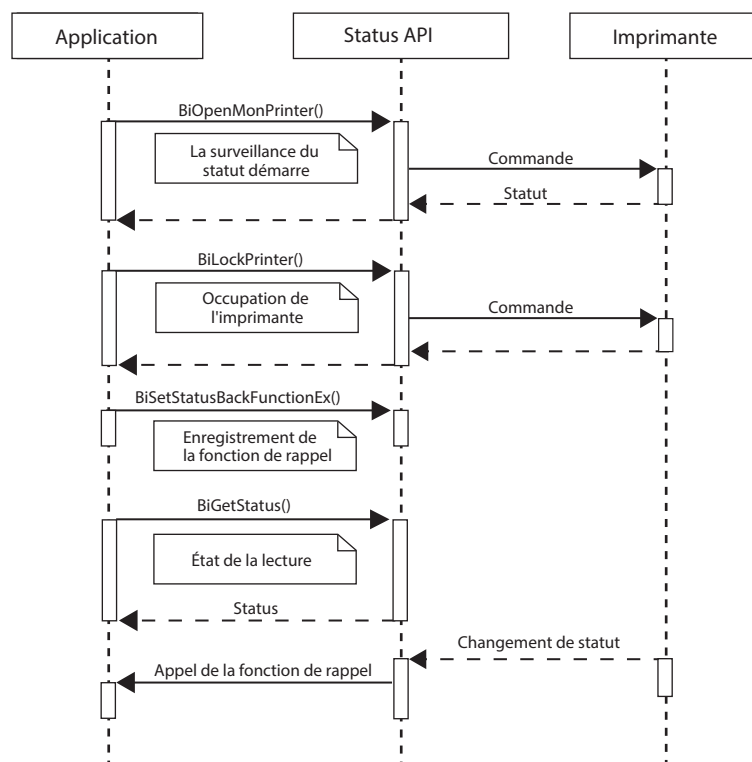
- Le chargement de la DLL n'est pas inclus dans le diagramme de séquence.
- Acquérez le statut ASB et assurez-vous que l'imprimante peut imprimer avant de lancer l'impression.
- Les détails du processus d'impression ne sont pas répertoriés.

Lorsque Status API (BiOpenMonPrinter) est démarrée, l'imprimante TM envoie automatiquement le statut ASB à Status API lorsque le statut change. L'API suivante peut être utilisée pour obtenir le statut ASB envoyé.

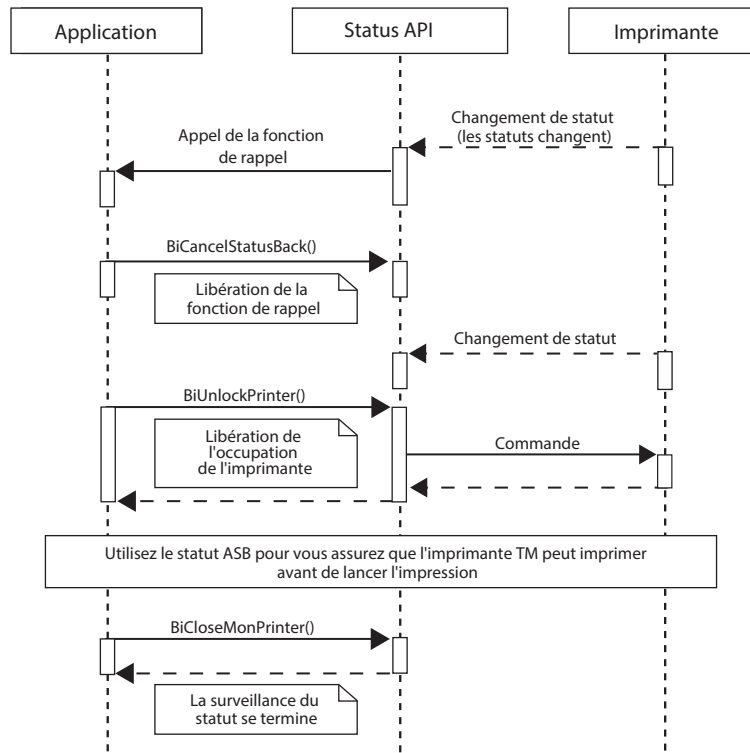
Status API	Description
BiGetStatus	API obtenant le statut ASB lorsque l'utilisateur (ou l'application) en a besoin.
BiSetStatusBackFunction	API appelant la fonction de rappel pour envoyer le dernier statut ASB à l'application.
BiSetStatusBackFunctionEx	API appelant la fonction de rappel pour envoyer le dernier statut ASB à l'application. Le port d'imprimante rappelé est également signalé.

Visual C++

(1/2)



(2/2)

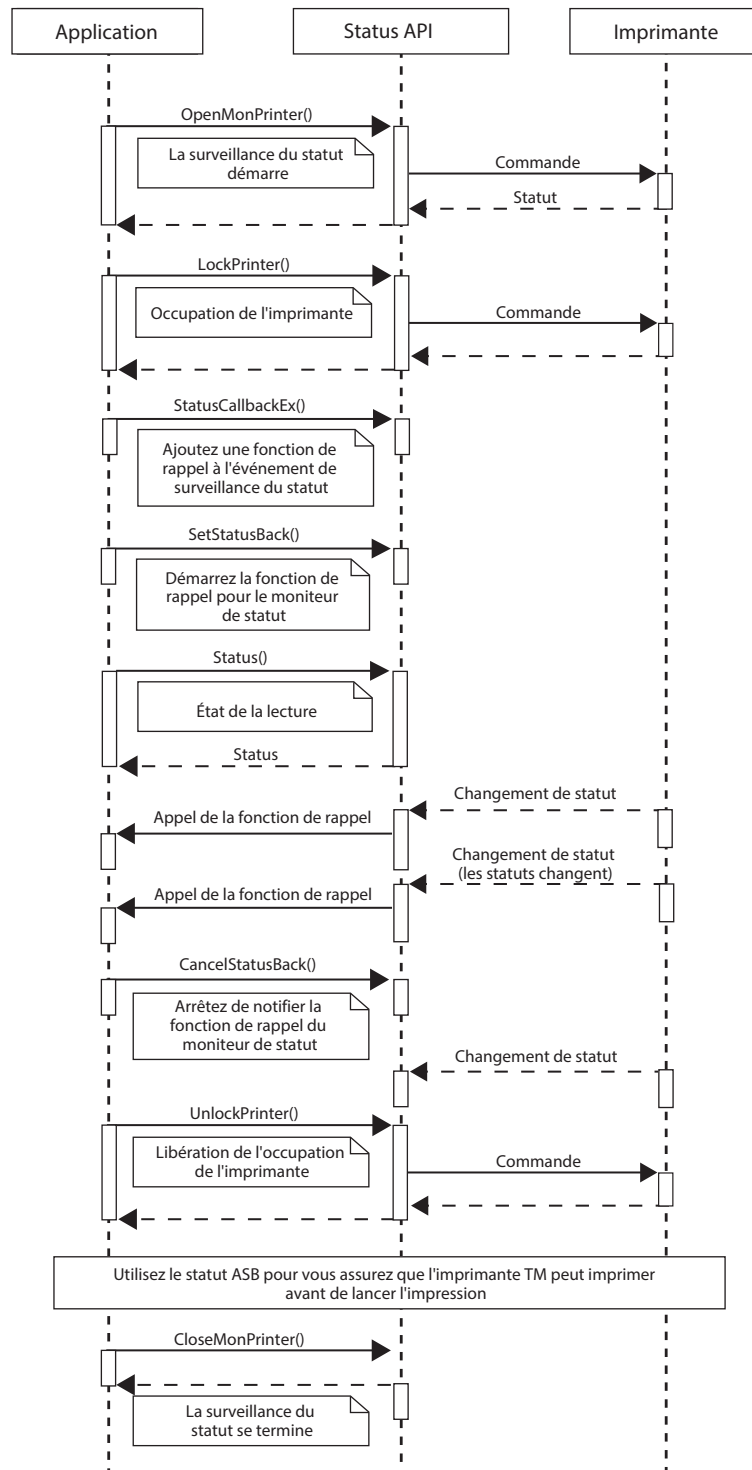


Visual C# / Visual Basic .NET

**REMARQUE**

Lorsque l'environnement de développement est Visual C# ou Visual Basic .Net, l'API .NET est utilisée.

Reportez-vous à "[Référence pour .NET](#)" à la page 64 pour le détail de l'API .NET.



Flux de programmation pour le contrôle MICR

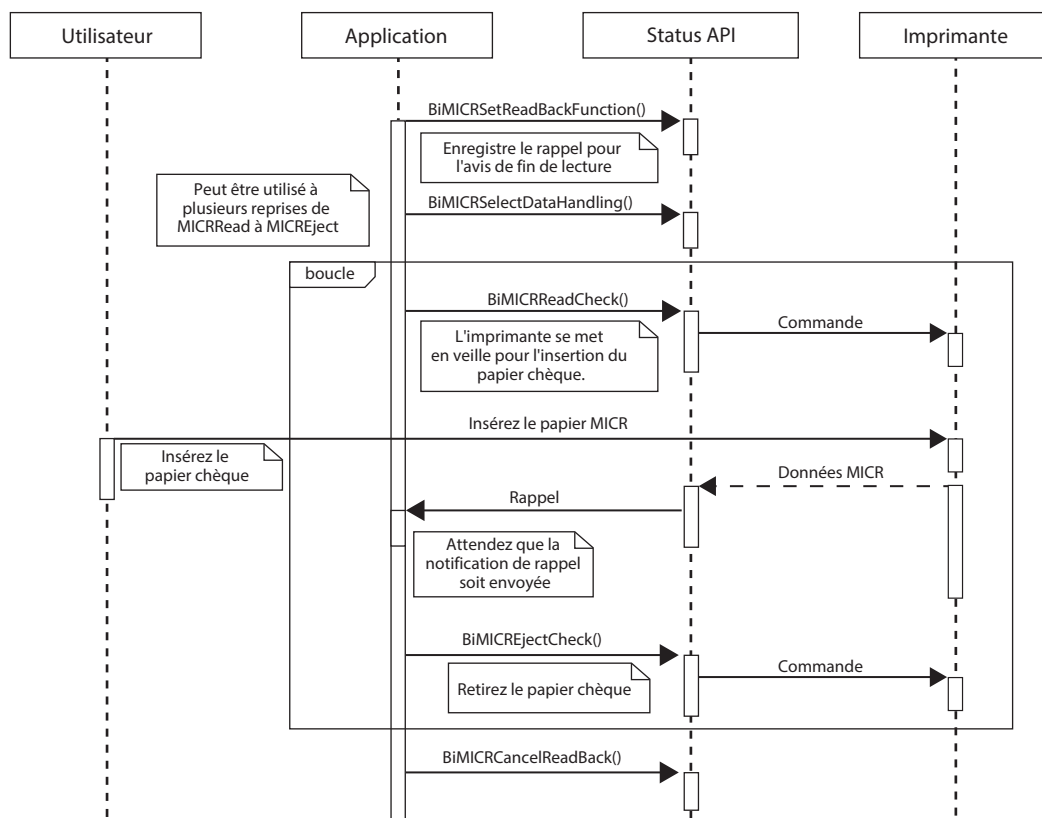
Le point suivant aborde la méthode de contrôle pour le contrôle MICR à l'aide d'une figure de séquence.



REMARQUE

Le chargement de la DLL n'est pas inclus dans le diagramme de séquence.

Visual C++

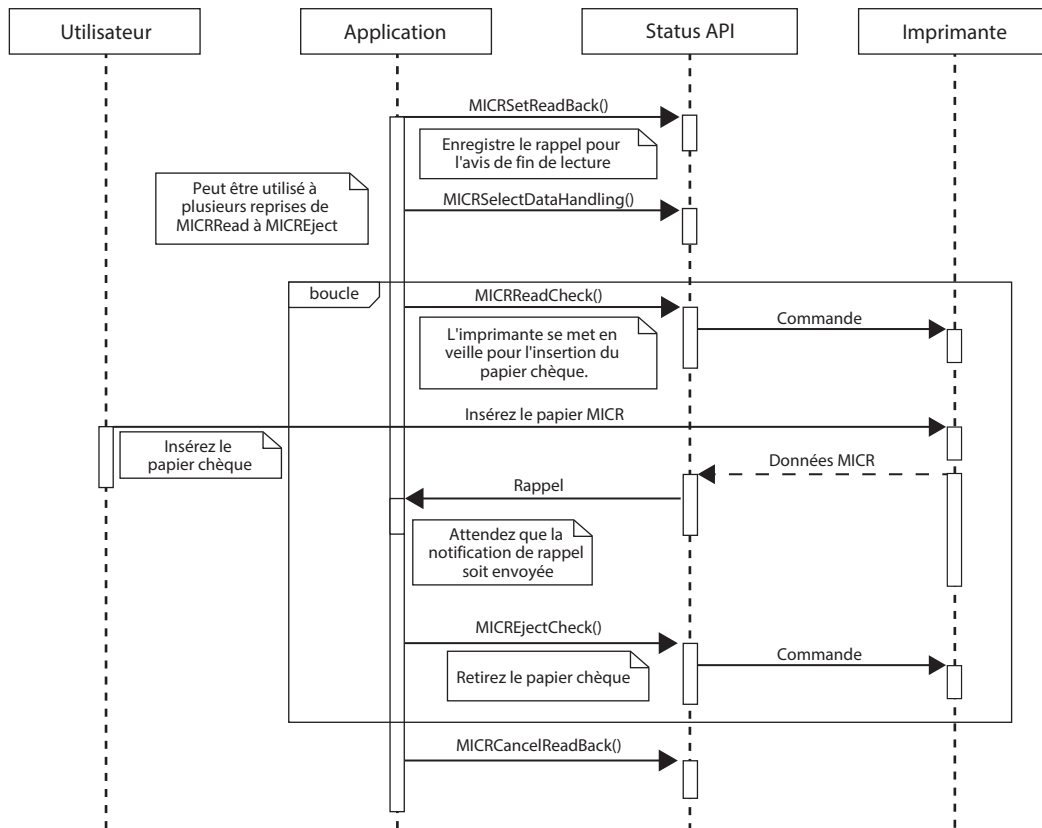


Visual C# / Visual Basic .NET

**REMARQUE**

Lorsque l'environnement de développement est Visual C# ou Visual Basic .Net, l'API .NET est utilisée.

Reportez-vous à ["Référence pour .NET" à la page 64](#) pour le détail de l'API .NET.



Erreurs et réponse de Status API

Les erreurs de Status API sont des erreurs lors de la notification du statut ASB et des erreurs générées lors de l'appel de Status API. Le point suivant explique les détails des erreurs et des réponses. Reportez-vous à ce qui suit et répondez aux erreurs de l'application.

Statut ASB

Voici les erreurs renvoyées lors de l'acquisition du statut ASB. Les détails diffèrent selon le modèle d'imprimante. Reportez-vous aux "Spécifications de l'imprimante" pour plus de détails.

Définition de macro (constante)	Cause	Réponse
ASB_NO_RESPONSE	- L'imprimante n'est pas sous tension. - Le câble de communication n'est pas connecté. - Le nom/port de l'imprimante spécifié est différent.	Confirmez le statut et les ports de l'imprimante, c'est-à-dire les câbles, etc.
ASB_PRINT_SUCCESS	Notifie que l'impression est terminée avec succès. Il n'y a rien d'autre qui soit notifié. Notifie que l'affichage sur l'écran client a été effectué avec succès. Il n'y a rien d'autre à notifier.	-
ASB_DRAWER_KICK	Le tiroir est ouvert.	Il n'y a pas de problème si le tiroir a été laissé ouvert intentionnellement.
ASB_BATTERY_OFFLINE	Hors ligne en raison d'un niveau de batterie faible.	Vérifiez le niveau de la batterie.
ASB_OFF_LINE	Une erreur a été générée provoquant la mise hors ligne de l'imprimante.	Éliminez la cause de la mise hors ligne de l'imprimante.
ASB_COVER_OPEN	Le capot est ouvert.	Fermez le capot de l'imprimante.
ASB_PAPER_FEED	Le papier est en cours d'alimentation.	Il n'y a pas de problème si le papier est en cours d'alimentation.
ASB_WAIT_ON_LINE	En attente de remise en ligne.	Vérifiez et corrigez la cause du statut "En attente de récupération en ligne".
ASB_PANEL_SWITCH	L'interrupteur du panneau de l'imprimante est enfoncé.	- Il n'y a pas de problème si un interrupteur sur le panneau de l'imprimante est enfoncé. - En réglant l'interrupteur ou le bouton de l'imprimante TM sur inactif (Ex. Bouton FEED), l'utilisateur peut utiliser le contrôle du bouton d'alimentation comme déclencheur qui peut générer/gérer des événements depuis l'application.
ASB_MECHANICAL_ERR	Une erreur mécanique s'est produite, p. ex. une erreur de détection de la position d'origine, etc.	Éliminez la cause de l'erreur et rallumez l'imprimante TM ou envoyez une commande de réinitialisation de l'imprimante TM (BiResetPrinter ou BiForceResetPrinter). *

Définition de macro (constante)	Cause	Réponse
ASB_RECOVER_ERR	Une erreur récupérable s'est produite.	Éliminez la cause de l'erreur et rallumez l'imprimante TM ou envoyez une commande de réinitialisation de l'imprimante TM (BiResetPrinter ou BiForceResetPrinter). *
ASB_AUTOCUTTER_ERR	Une erreur de coupe auto s'est produite.	Éliminez la cause de l'erreur et rallumez l'imprimante TM ou envoyez une commande de réinitialisation de l'imprimante TM (BiResetPrinter ou BiForceResetPrinter). *
ASB_UNRECOVER_ERR	Une erreur non récupérable s'est produite dans l'imprimante TM.	Éteignez immédiatement l'imprimante. *
ASB_AUTORECOVER_ERR	La température de la tête a augmenté.	Si la température de la tête diminue avec le temps, l'erreur s'annule automatiquement. *
ASB_SLIP_INSERT_WAIT	L'imprimante attend l'insertion du bordereau.	Placez le bordereau dans l'imprimante.
ASB_RECEIPT_NEAR_END	Il ne reste qu'une quantité limitée de papier.	Chargez le papier dans l'imprimante.
ASB_PAPER_END	Il n'y a pas de papier en rouleau.	Chargez le papier dans l'imprimante.
ASB_SLIP_REMOVE_WAIT	L'imprimante attend le retrait du bordereau.	Retirez le bordereau de l'imprimante.
ASB_RECEIPT_END	Pas de papier.	Chargez le papier dans l'imprimante.
ASB_SLIP_TOF	Aucun papier n'est détecté. Ceci n'est pas une erreur, mais indique le statut de l'imprimante.	-
ASB_SLIP_BOF	Aucun papier n'est détecté. Ceci n'est pas une erreur, mais indique le statut de l'imprimante.	-
ASB_SLIP_SELECTED	Ceci indique si le bordereau a été sélectionné.	-
ASB_WAIT_REMOVE_LABEL	En attente du retrait de l'étiquette.	Retirez l'étiquette de l'imprimante TM.
ASB_PRINT_SLIP	L'imprimante attend l'insertion du bordereau ou le bordereau est en cours de copie/décharge.	-
ASB_VALIDATION_SELECTED	Ceci indique si la validation a été sélectionnée.	-
ASB_NO_LABEL	Une étiquette/un papier a été détecté par le détecteur de décollement d'étiquette/détecteur de retrait de papier.	Retirez l'étiquette/le papier de l'imprimante TM.
ASB_PRINT_VALIDATION	L'imprimante attend l'insertion du bordereau ou la validation est en cours de copie/décharge.	-
ASB_VALIDATION_TOF	Aucun papier n'est détecté. Ceci n'est pas une erreur, mais indique le statut de l'imprimante.	-

Définition de macro (constante)	Cause	Réponse
ASB_EJECT_DETECT_NO_PAPER	Aucun papier n'est détecté par le capteur d'éjection. Ceci n'est pas une erreur, mais indique le statut de l'imprimante.	-
ASB_VALIDATION_NO_PAPER	Aucun papier n'est détecté par le détecteur de validation. Ceci n'est pas une erreur, mais indique le statut de l'imprimante.	-
ASB_REMOVAL_DELECT_NO_PAPER	Du papier reste dans la fente d'éjection du papier.	Retirez le papier de l'imprimante TM.
ASB_REMOVAL_DETECT_UNKNOWN	Le papier n'est pas détecté correctement.	N'exposez pas la fente d'éjection du papier de l'imprimante TM à une forte lumière.
ASB_SPOOLER_IS_STOPPED	Le spouleur est arrêté.	Redémarrez l'ordinateur.

* Reportez-vous aux manuels d'utilisation détaillés des différentes imprimantes.

 REMARQUE	La définition de la macro est définie à l'aide du fichier EpsStmApi.h ou Module1.bas lors de la construction de l'environnement de développement.
---	---

Erreur d'exécution de Status API

Les erreurs suivantes sont générées lorsque les fonctions de Status API sont appelées. Le contenu diffère selon la fonction de Status API. Référez-vous à ["Référence pour Win32" à la page 25](#) pour plus de détails.

Définition de macro (constante)	Cause	Réponse
ERR_TYPE	Les paramètres de nType diffèrent.	Spécifiez la valeur correcte.
ERR_OPENED	L'imprimante spécifiée est déjà ouverte.	L'imprimante étant déjà ouverte, utilisez la valeur de handle (identifiant) ou spécifiez une imprimante différente.
ERR_NO_PRINTER	Le pilote d'imprimante spécifié n'existe pas.	Confirmez le nom du pilote d'imprimante.
ERR_NO_TARGET	L'imprimante spécifiée ne peut pas être trouvée. Une imprimante non spécifiée est connectée.	Connecté à la bonne imprimante.
ERR_NO_MEMORY	La mémoire est insuffisante.	Ajouter la mémoire disponible.
ERR_HANDLE	La valeur de handle (identifiant) spécifiée par l'imprimante est incorrecte.	Confirmez la valeur de handle.
ERR_TIMEOUT	Ceci est une erreur de temporisation.	Si l'erreur est générée en continu, confirmez si l'imprimante est correctement connectée.
ERR_ACCESS	L'accès ne peut pas être effectué sur l'imprimante. (L'imprimante n'est pas sous tension ou le câble n'est pas correctement connecté, etc.)	Confirmez l'imprimante. (Alimentation de l'imprimante, connexion du câble, etc.)
ERR_BUFFER_OVER_FLOW	Erreur de débordement de tampon.	-
ERR_PARAM	Ceci est une erreur de paramètre.	Vérifiez la syntaxe, car le paramètre spécifié est incorrect.
ERR_NOT_SUPPORT	Ceci est un modèle non pris en charge.	Les modèles non pris en charge ne peuvent pas être utilisés.
ERR_EXIST	Les données spécifiées existent déjà.	Supprimez les données déjà existantes. Exemple : Lorsque cette erreur se produit lors de l'exécution de BiSetStatusBackXXX, réessayez après avoir exécuté BiCancelStatusBack.
ERR_EXEC_FUNCTION	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.	Fermez Status API utilisée par d'autres applications.
ERR_PH_NOT_EXIST	Le PortHandler ne s'exécute pas, ou une erreur de communication est survenue entre le client du PortHandler et le serveur.	Vérifiez la connexion entre eux, puis redémarrez l'ordinateur.
ERR_SPL_NOT_EXIST	Le service du spouleur ne fonctionne pas.	Vérifiez si le service du spouleur d'impression a démarré. (Panneau de configuration - Outils de gestion - Service)

Définition de macro (constante)	Cause	Réponse
ERR_RESET	Cette fonction est indisponible, car l'imprimante est en cours de réinitialisation.	Rappelez après avoir attendu un moment.
ERR_LOCKED	L'imprimante est verrouillée.	Attendez que l'imprimante soit déverrouillée, ou exécutez BiUnlockPrinter dans le programme qui verrouille l'imprimante.
ERR_EXEC_MICR	Impossible d'appeler, car le MICR est en cours de lecture	Exécutez une fois que le traitement MICR est terminé.
ERR_EXEC_SCAN	Impossible d'appeler, car le scanner est en cours de numérisation.	Exécutez une fois que la numérisation est terminée.
 REMARQUE	La définition de la macro est définie à l'aide du fichier EPSStmApi.h ou StatusAPI.bas lors de la construction de l'environnement de développement.	

Référence pour Win32

Ce chapitre explique Status API et la syntaxe utilisée dans l'environnement Win32.



REMARQUE

- Le type de données est décrit dans C++.
- Voir "Spécifications de l'imprimante" pour les détails concernant les API disponibles pour chaque type d'imprimante TM, le statut ASB et les raisons pour lesquelles l'imprimante est hors ligne.

BiOpenMonPrinter

Rend Status API disponible pour l'imprimante et renvoie le handle.

Vous pouvez ouvrir une imprimante à partir de plusieurs processus en même temps.

Lorsque vous ouvrez l'imprimante ouverte à partir du même processus, un nouveau handle différent sera renvoyé. Dans ce cas, les deux handles sont valides.

Syntaxe

int *BiOpenMonPrinter* (int nType, LPSTR pName)

Paramètre

nType : Spécifie le type pName.

Définition de macro (Constante)	Valeur	Description
TYPE_PORT	1	Spécifier le nom du port dans <i>pName</i> .
TYPE_PRINTER	2	Spécifier le nom de l'imprimante dans <i>pName</i> .

pName : Si 1 est spécifié dans nType, spécifier le nom du port (exemple : "ESDPRT001").

Si 2 est spécifié, spécifier le nom de l'imprimante (exemple : "EPSON TM-T88V Receipt").

Exemple)

- Spécification d'une imprimante avec le nom de port (ESDPRT001)
nHandle=BiOpenMonPrinter(1,"ESDPRT001");
- Spécification d'une imprimante avec le nom d'imprimante (EPSON TM-T88V Receipt)
nHandle=BiOpenMonPrinter(2,"EPSON TM-T88V Receipt");
- Spécification de l'imprimante partagée (EPSON TM-T88V Receipt) avec le nom d'hôte "SERVER"
nHandle=BiOpenMonPrinter(2,\\SERVER\\”EPSON TM-T88V Receipt”);

Valeur de retour

Renvoie la variable définie en type INT. Si Status API est utilisé avec succès, le handle identifiant l'imprimante est renvoyé. Le handle est renvoyé même si l'imprimante est hors ligne. Les erreurs d'exécution de Status API (valeur) suivantes sont renvoyées.

Définition de macro (Constante)	Valeur	Description
ERR_TYPE	-10	Erreur de paramètre de <i>nType</i>
ERR_OPENED	-20	L'imprimante spécifiée est déjà ouverte.
ERR_NO_PRINTER	-30	Le pilote d'imprimante spécifié n'existe pas
ERR_NO_TARGET	-40	Imprimante indisponible
ERR_NO_MEMORY	-50	Mémoire insuffisante
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_SPL_NOT_EXIST	-350	Le service du spouleur ne fonctionne pas.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Appelez cette fonction avant d'utiliser d'autres fonctions de Status API. Le handle de la valeur retournée est utilisé comme paramètre par d'autres fonctions de Status API.

Le nombre maximal d'imprimantes pouvant être démarrées en même temps est 32.

Les opérations suivantes sont exécutées en fonction de l'état de l'imprimante lorsque cette fonction est appelée.

État de l'imprimante	Fonctionnement
En ligne	Renvoie le handle.
Hors ligne	Renvoie le handle. Cependant, passez en ligne, car l'imprimante ne peut pas imprimer hors ligne.
Câble retiré/Mise hors tension	Renvoie "ERR_ACCESS".

BiCloseMonPrinter

Annule l'imprimante de surveillance d'état.

Lorsqu'une fonction BiOpenMonPrinter est appelée, annulez toujours l'imprimante de surveillance d'état à l'aide de la fonction BiCloseMonPrinter.

Syntaxe

int **BiCloseMonPrinter** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiLockPrinter

Verrouille l'imprimante.



REMARQUE

- Cette API est utilisée pour une imprimante partagée.
- Lors de l'utilisation d'une imprimante locale, cette API est utilisée pour contrôler plusieurs processus.

Syntaxe

int **BiLockPrinter** (int nHandle, DWORD timeout)

Paramètre

nHandle : Spécifie le handle.

timeout : Spécifie le délai d'attente en ms (millisecondes). Spécifiez-le avec une valeur positive.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Cette API vous permet d'accéder exclusivement à l'imprimante TM. L'API BiUnlockPrinter est fournie pour annuler l'accès exclusif. Pendant que l'imprimante TM est exclusivement accessible, elle n'accepte aucune autre demande d'API d'accès direct à l'imprimante. L'imprimante renverra `ERR_LOCKED` à ces autres demandes d'API.

Le droit d'accès exclusif à l'imprimante TM est accordé à un processus. Par conséquent, l'accès API exclusif est disponible à partir d'autres séquences d'instructions dans le même processus qui verrouille l'imprimante.

L'exécution de l'API dans le même processus peut être répétée. Dans ce cas, l'imprimante est verrouillée avec plusieurs accès. Pour déverrouiller l'imprimante, exécutez BiUnlockPrinter autant de fois que l'API a été exécutée.

Lors de l'exécution d'un accès exclusif d'un client à une imprimante partagée ou à une imprimante locale via Ethernet, l'état d'accès est interrompu si la connexion est perdue, et restauré lors de la récupération de la connexion.

Cependant, pendant que l'état d'accès exclusif est interrompu, un autre processus peut verrouiller l'imprimante TM pour un accès exclusif. Une fois que l'imprimante est verrouillée par un autre processus, l'imprimante renvoie `ERR_LOCKED` à l'API du processus précédent. Lorsqu'un autre processus a fini de déverrouiller l'imprimante, l'état d'accès exclusif du processus précédent est restauré.

Les causes possibles de l'échec de la connexion sont les suivantes.

[Échec durant l'accès exclusif à une imprimante connectée via Ethernet]

- L'imprimante est éteinte, ou la connexion Ethernet entre l'ordinateur et l'imprimante est déconnectée.
- L'ordinateur est passé en mode Veille ou Hibernation.

[Échec durant l'accès exclusif à une imprimante partagée à partir d'un client]

- La connexion entre le client et le serveur est déconnectée.
- L'ordinateur client est entré en mode Veille ou Hibernation.

BiUnlockPrinter

Déverrouille le verrou de l'imprimante.



REMARQUE

- Cette API est utilisée pour une imprimante partagée.
- Lors de l'utilisation d'une imprimante locale, cette API est utilisée pour contrôler plusieurs processus.

Syntaxe

int **BiUnlockPrinter** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Cette API déverrouille l'imprimante verrouillée par "BiLockPrinter". Une fois le verrou annulé, l'imprimante peut accepter l'API d'autres processus.

Si vous exécutez cette API lorsque l'imprimante n'est pas verrouillée, "ERR_LOCKED" sera renvoyé comme valeur de retour.

BiDirectIOEx

Envoie des commandes spéciales (commande ESC/POS) à l'imprimante. Peut également acquérir les résultats d'exécution des commandes de l'imprimante. La commande de suppression ASB peut être ajoutée pour les différences avec BiDirectIO. Lorsque la commande de suppression ASB est ajoutée, les données séparées (statut ASB, etc.) ne sont pas envoyées par l'imprimante tant que cette fonction n'est pas terminée. Par conséquent, ceci est recommandé lors de la réception des résultats d'exécution de l'imprimante.



REMARQUE

- Référez-vous à l'URL suivante pour les détails de la commande ESC/POS.
https://support.epson.net/publist/reference_en/

Syntaxe

int **BiDirectIOEx** (int nHandle, DWORD writeLen, LPBYTE writeCmd, LPDWORD readLen, LPBYTE readBuff, DWORD timeout, BOOL nullTerminate, BYTE option)

Paramètre

- nHandle : Spécifie le handle.
- writeLen : Spécifie la longueur des données à écrire sur l'imprimante. N'écrit pas sur l'imprimante si la valeur est "0".
- writeCmd : Spécifie les données (commande ESC/POS) à écrire sur l'imprimante.
- readLen : Spécifie la longueur des données lues depuis l'imprimante.
Spécifiez lorsque les résultats d'exécution des commandes sont requis de l'imprimante.
Spécifiez "0" si non requis.
- readBuff : Spécifie le tampon contenant les données lues depuis l'imprimante.
- timeout : Spécifie le délai d'attente en ms (millisecondes).
- nullTerminate : Dans le cas de "True", la lecture est terminée lorsque NULL est reçu de l'imprimante. À ce moment, spécifiez le format de readBuff pour readLen.
Dans le cas de "FALSE", la longueur des données spécifiée dans readLen est lue ou les données sont lues de l'imprimante jusqu'à ce qu'une erreur de temporisation soit générée.
- option : Contrôle la commande de suppression ASB.

Valeur	Description
0	Ne pas acquérir le statut ASB.
1	Acquérir le statut ASB après l'acquisition des données.



REMARQUE

- Bien que la longueur maximale de données qui peut être spécifiée pour readLen/writeLen soit de 2 Go, spécifiez la longueur minimale de données requise.
- Assurez-vous que le format de readBuff est de la même longueur spécifiée dans readLen ou plus long.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_NO_MEMORY	-50	Mémoire insuffisante
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_BUFFER_OVER_FLOW	-140	Manque de capacité du tampon.
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Confirmez la bonne exécution de la fonction en vérifiant la valeur de retour ou la bonne exécution des commandes en confirmant le fonctionnement de l'imprimante. Si les résultats d'exécution sont acquis de l'imprimante (spécifier readLen), confirmez les résultats d'exécution.

Les opérations suivantes sont exécutées en fonction de l'état de l'imprimante lorsque cette fonction est appelée.

État de l'imprimante	Fonctionnement
En ligne	Renvoie "SUCCESS" à la valeur de retour. Exécute la commande.
Hors ligne	• Renvoie "SUCCESS" lorsque la communication avec l'imprimante est terminée avec succès dans le temps spécifié avec la temporisation. • Renvoie "ERR_TIMEOUT" en cas d'échec de la communication avec l'imprimante dans le temps spécifié avec la temporisation.
Câble retiré/Mise hors tension	Renvoie "ERR_ACCESS" à la valeur de retour.
Impression	Renvoie "ERR_LOCKED" à la valeur de retour.

Attention

- Ne pas envoyer de commandes invalides de transmissions de statut ASB à l'aide de cette fonction pendant la surveillance du statut de l'imprimante. Le statut suivant ne peut pas être acquis.
- La commande de suppression ASB (notification de statut automatique) garantit que des données non prévues ne sont pas reçues lors de l'envoi de commandes demandant une réponse de l'imprimante.
Si vous n'utilisez pas la suppression ASB, assurez-vous que la programmation tient compte de la réception de données non prévues.
- La spécification du tampon de réception traite les données reçues de l'imprimante à l'aide de cette fonction ou traite les données à l'aide du même processus que la fonction de surveillance (fonction BiGetStatus, etc.).
Se référer à ce qui suit.

Transmission Commande	Tampon de réception spécifié	Tampon de réception	Fonctionnement du processus de surveillance
Acquisition commande de statut	Oui	Enregistre le statut ASB dans le tampon de réception	• Pas de rappel • Ne renouvelle pas le statut
	Non	-	• Pas de rappel • Ne renouvelle pas le statut
Commande avec réponses d'autres imprimantes	Oui	Saisit la réponse de l'imprimante dans le tampon de réception	N'affecte pas le processus de surveillance
	Non	-	Des rappels anormaux peuvent être générés
Commande sans réponse des autres imprimantes	Oui	Génère une erreur de temporisation	N'affecte pas le processus de surveillance
	Non	-	N'affecte pas le processus de surveillance

BiResetPrinter

Réinitialise les imprimantes de surveillance de statut.



REMARQUE

- Annule les travaux d'impression lorsque cette fonction est appelée pendant l'impression.
- L'imprimante TM avec l'interface série ne peut pas être réinitialisée.

Syntaxe

int **BiResetPrinter** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Confirmez la bonne exécution de la fonction en vérifiant la valeur de retour de *la valeur de retour* ou en réinitialisant l'imprimante et en vérifiant que l'imprimante est en ligne (en confirmant le statut ASB).



REMARQUE

Après l'exécution de cette fonction, l'imprimante ne peut pas recevoir de commande d'impression pendant 15 secondes. Si l'impression est exécutée pendant ce temps, le travail est envoyé au spouleur et le processus d'impression est exécuté après l'écoulement du temps susmentionné.

Les opérations suivantes sont exécutées en fonction de l'état de l'imprimante lorsque cette fonction est appelée.

État de l'imprimante	Fonctionnement
En ligne	Renvoie "SUCCESS" à la valeur de retour et réinitialise.
Hors ligne	Renvoie "SUCCESS" à la valeur de retour et réinitialise.
Câble retiré/Mise hors tension	Renvoie "ASB_NO_RESPONSE" au statut ASB et ne réinitialise pas.
Impression	Annule le travail d'impression et réinitialise.

BiForceResetPrinter

Force la réinitialisation de l'imprimante TM dont le statut est surveillé.

L'imprimante TM peut être réinitialisée même dans des environnements multifils/multiprocédure/multi-utilisateur.

L'imprimante TM peut être réinitialisée même si BiLockPrinter est accédé depuis un programme différent.

Si la connexion à une imprimante réseau est perdue puis rétablie, un certain temps sera nécessaire avant que l'impression ne soit à nouveau possible. L'utilisation de cette API permet de raccourcir ce temps.



REMARQUE

- Cette API forcera la réinitialisation de l'imprimante même si l'impression est en cours, et les données en cours d'impression seront supprimées.
- L'imprimante TM avec l'interface série ne peut pas être réinitialisée.

Syntaxe

```
int BiForceResetPrinter (int nHandle)
```

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiGetType

Acquiert le type ID de police de l'imprimante.



REMARQUE

Pour obtenir des informations sur le type ID de police qui peut être acquis, demandez à votre revendeur.

Syntaxe

int **BiGetType** (int nHandle, LPBYTE typeID, LPBYTE font, LPBYTE exrom, LPBYTE special)

Paramètre

nHandle : Spécifie le handle.
 typeID : Un typeID de l'imprimante sera défini.
 font : La police de périphérique sera définie.
 exrom : Ceci n'est pas applicable.
 special : Un ID spécial de l'imprimante sera défini.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Certaines informations peuvent ne pas être obtenues selon les modèles. Dans ce cas, 0 est renvoyé à typeID.

BiGetStatus

Acquiert le statut actuel de l'imprimante (statut ASB).

Syntaxe

int **BiGetStatus** (int nHandle, LPDWORD lpStatus)

Paramètre

nHandle : Spécifie le handle.

lpStatus : Renvoie le statut ASB enregistré dans Status API.
Le statut ASB est une configuration de 4 octets.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Reportez-vous aux "Spécifications de l'imprimante" concernant le statut ASB qui peut être acquis par chaque imprimante.

BiSetStatusBackFunction

Acquiert automatiquement le statut de l'imprimante (statut ASB) à l'aide de la fonction de rappel lorsque le statut de l'imprimante change.

Syntaxe

```
int BiSetStatusBackFunction (int nHandle, int (CALLBACK EXPORT
                             *pStatusCB)(DWORD dwStatus))
```

Paramètre

nHandle : Spécifie le handle.

*pStatusCB : Spécifie l'adresse de définition de la fonction de rappel.

Paramètre de la fonction de rappel

dwStatus : Renvoie le statut ASB enregistré dans Status API.

Le statut ASB est une configuration de 4 octets.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_EXIST	-210	Les données spécifiées existent déjà.
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Appelez cette fonction pour définir le statut de l'imprimante sur *dwStatus* et appeler la fonction de rappel. Lorsque le statut de l'imprimante change, de nouvelles informations sont automatiquement définies sur *dwStatus* et appellent la fonction de rappel. Annuler cette fonction en utilisant *BiCancelStatusBack*.

Reportez-vous aux "Spécifications de l'imprimante" concernant le statut ASB qui peut être acquis par chaque imprimante.



REMARQUE

Status API ne peut pas être utilisé dans la fonction de rappel enregistrée.

BiSetStatusBackFunctionEx

Acquiert automatiquement le statut de l'imprimante (statut ASB) à l'aide de la fonction de rappel lorsque le statut de l'imprimante change.

Identifie le port de l'imprimante à l'origine du rappel, en plus des fonctions de BiSetStatusBackFunction.

Syntaxe

```
int BiSetStatusBackFunctionEx (int nHandle, int (CALLBACK EXPORT *pStatusCB)
                                (DWORD dwStatus, LPSTR lpcPortName))
```

Paramètre

nHandle : Spécifie le handle.

*pStatusCB : Spécifie l'adresse de définition de la fonction de rappel.

Paramètre de la fonction de rappel

dwStatus : Renvoie le statut ASB enregistré dans Status API. Le statut ASB est une configuration de 4 octets.

lpcPortName : Renvoie le nom du port de l'imprimante à l'origine du rappel.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_EXIST	-210	Les données spécifiées existent déjà.
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Appelez cette fonction pour définir le statut de l'imprimante sur dwStatus et appeler la fonction de rappel. Lorsque le statut de l'imprimante change, de nouvelles informations sont automatiquement définies sur dwStatus et appellent la fonction de rappel. Annuler cette fonction en utilisant BiCancelStatusBack.

Reportez-vous aux "Spécifications de l'imprimante" concernant le statut ASB qui peut être acquis par chaque imprimante.



REMARQUE

Status API ne peut pas être utilisé dans la fonction de rappel enregistrée.

BiCancelStatusBack

Annule le processus de demande de notification automatique du statut appelé à l'aide de la fonction BiSetStatusBackFunction/BiSetStatusBackFunctionEx.

Syntaxe

```
int BiCancelStatusBack (int nHandle)
```

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

- Renvoie "SUCCESS" même lorsqu'il est exécuté si le processus de demande de notification automatique du statut n'est pas enregistré.
- Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiGetBatteryStatus

Acquiert le statut actuel de la batterie de l'imprimante.

Syntaxe

```
int BiGetStatus (int nHandle, LPBYTE lpbPowerStatus, LPBYTE lpbBatteryStatus)
```

Paramètre

nHandle : Spécifie le handle.

lpbPowerStatus : Spécifie l'adresse mémoire où le statut d'alimentation est défini.

lpbBatteryStatus : Spécifie l'adresse mémoire où la quantité restante de la batterie est définie.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

lpbPowerStatus spécifie si l'alimentation de l'imprimante est fournie par l'adaptateur secteur ou la batterie. lpbBatteryStatus spécifie le niveau de la batterie.

Reportez-vous aux "Spécifications de l'imprimante" concernant le statut de la batterie qui peut être acquis par chaque imprimante.

BiSetBatteryStatusBackFunctionEx

Acquiert automatiquement le statut de l'imprimante (statut ASB) à l'aide de la fonction de rappel lorsque le statut de l'imprimante change. Identifie le port de l'imprimante à l'origine du rappel, en plus des fonctions de BiSetStatusBackFunction.

Syntaxe

```
int BiSetBatteryStatusBackFunctionEx
(int nHandle, int (CALLBACK EXPORT *pBatteryCB)
(BYTE bPowerStatus, BYTE bBatteryStatus, LPSTR lpcPortName))
```

Paramètre

nHandle : Spécifie le handle.
 *pBatteryCB : Spécifie l'adresse de définition de la fonction de rappel.

Paramètre de la fonction de rappel

bPowerStatus : Définit le statut d'alimentation pour lpbPowerStatus, puis envoie l'événement.
 bBatteryStatus : Définit la valeur de statut de la batterie pour lpbBatteryStatus, puis envoie l'événement.
 lpcPortName : Renvoie le nom du port de l'imprimante à l'origine du rappel.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_EXIST	-210	Les données spécifiées existent déjà.
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Appelez cette fonction pour définir le statut de l'imprimante sur bPowerStatus / bBatteryStatus et appelez la fonction de rappel.

Lorsque le statut de l'imprimante change, de nouvelles informations sont automatiquement définies sur bPowerStatus / bBatteryStatus et appellent la fonction de rappel. Annuler cette fonction en utilisant BiCancelBatteryStatusBack.

Reportez-vous aux "Spécifications de l'imprimante" concernant le statut de la batterie qui peut être acquis par chaque imprimante.



REMARQUE

Status API ne peut pas être utilisé dans la fonction de rappel enregistrée.

BiCancelBatteryStatusBack

Annule le processus de demande de notification de statut automatique appelé à l'aide de `BiSetBatteryStatusBackFunctionEx`.

Syntaxe

int ***BiCancelBatteryStatusBack*** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

- Renvoie "SUCCESS" même lorsqu'il est exécuté si le processus de demande de notification automatique du statut n'est pas enregistré.
- Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiPowerOff

Met l'imprimante TM hors tension ou en mode veille.



REMARQUE

Cette API ne peut pas être utilisée lorsque l'imprimante TM est en attente de récupération d'un état hors ligne. De plus, si l'imprimante TM passe hors ligne pendant que cette API est active, l'API ne peut plus être utilisée.

Syntaxe

```
int BiPowerOff(int nHandle)
```

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_NO_MEMORY	-50	Mémoire insuffisante
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiAutoPowerOffTime

Acquiert et définit le temps restant avant que l'imprimante mobile ne s'éteigne. Impossible d'éteindre l'imprimante.

Syntaxe

int **BiAutoPowerOffTime** (int nHandle, BYTE bMode, LPBYTE lpbTime)

Paramètre

nHandle : Spécifie le handle.

bMode : Spécifie le mode de processus.

Définition de macro (Constante)	Valeur	Description
EPS_BI_GET	0	Acquiert le temps d'arrêt automatique.
EPS_BI_SET	1	Définit le temps d'arrêt automatique.

lpbTime : Spécifie l'adresse mémoire pour définir le temps d'arrêt automatique.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_NO_MEMORY	-50	Mémoire insuffisante
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Les éléments suivants sont définis à l'adresse spécifiée par lpbTime.

Valeur	Description
0	Arrêt automatique : Désactivé
1 - 60	1 à 60 minutes

BiGetPrnCapability

Acquiert les informations spécifiées de l'imprimante dans l'ID de l'imprimante.



REMARQUE

- Pour les informations sur les capacités de l'imprimante pouvant être acquises, contactez votre revendeur.
- Il existe des ID d'imprimante qui ne sont pas pris en charge par les imprimantes. Si des ID d'imprimante non pris en charge sont spécifiés, une erreur de temporisation est générée.

Syntaxe

int **BiGetPrnCapability** (int nHandle, BYTE prnID, LPBYTE pBuffSize, LPBYTE pBuff)

Paramètre

- nHandle : Spécifie le handle.
- prnID : Spécifie les informations de l'imprimante en cours d'acquisition.
- pBuffSize : Spécifie la taille de la mémoire pour définir les informations de l'imprimante (1 à 80). Retourne la taille réelle des données lues après l'appel de cette fonction. En cas de capacité de tampon insuffisante, la taille en octets requise est renvoyée.
- pBuff : Spécifie l'adresse mémoire pour définir les informations de l'imprimante.

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_BUFFER_OVER_FLOW	-140	Manque de capacité du tampon.
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiOpenDrawer

Ouvre le tiroir.

 REMARQUE	Ouvre le tiroir même lorsque l'imprimante TM est hors ligne.
---	--

Syntaxe

int **BiOpenDrawer** (int nHandle, BYTE drawer, BYTE pulse)

Paramètre

nHandle : Spécifie le handle.

drawer : Spécifie le tiroir à ouvrir.

Définition de macro (Constante)	Valeur	Description
EPS_BI_DRAWER_1	1	Ouvre le tiroir 1
EPS_BI_DRAWER_2	2	Ouvre le tiroir 2

pulse : Spécifie le temps pendant lequel le signal de déclenchement du tiroir est activé.

Définition de macro (Constante)	Valeur	Description
EPS_BI_PLUSE_100	1	Signal pendant 100 millisecondes
EPS_BI_PLUSE_200	2	Signal pendant 200 millisecondes
EPS_BI_PLUSE_300	3	Signal pendant 300 millisecondes
EPS_BI_PLUSE_400	4	Signal pendant 400 millisecondes
EPS_BI_PLUSE_500	5	Signal pendant 500 millisecondes
EPS_BI_PLUSE_600	6	Signal pendant 600 millisecondes
EPS_BI_PLUSE_700	7	Signal pendant 700 millisecondes
EPS_BI_PLUSE_800	8	Signal pendant 800 millisecondes

Valeur de retour

Définition de macro (Constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Cette fonction est indisponible, car Status API est utilisée par d'autres applications.
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Les opérations suivantes sont exécutées en fonction de l'état de l'imprimante lorsque cette fonction est appelée.

État de l'imprimante	Fonctionnement
En ligne	Renvoie "SUCCESS" à la valeur de retour. Ouvre le tiroir.
Hors ligne	Renvoie "SUCCESS" à la valeur de retour. Ouvre le tiroir.
Câble retiré/Mise hors tension	Renvoie "ERR_ACCESS" à la valeur de retour. N'ouvre pas le tiroir.

BiMICRCleaning

Nettoie le mécanisme MICR.

Syntaxe

int **BiMICRCleaning** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Si cette fonction est appelée, le mécanisme attend l'insertion de la feuille de nettoyage. Insérez la feuille de nettoyage et procédez au nettoyage du mécanisme. Une fois le nettoyage terminé, la fonction MICR est désélectionnée.

BiMICRGetStatus

Spécifie l'adresse mémoire où le statut MICR est défini.

Syntaxe

int **BiMICRGetStatus** (int nHandle, LPBYTE pStatus)

Paramètre

nHandle : Spécifie le handle.

pStatus : Spécifie l'adresse mémoire où le statut MICR est défini.

Voir les états MICR suivants concernant les types de statut MICR qui sont retournés.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Pour les statuts MICR qui peuvent être acquis de chaque imprimante, reportez-vous aux "Spécifications de l'imprimante".

BiMICRLoadCheck

Charge le chèque à la position de début d'impression du chèque.

Syntaxe

int **BiMICRLoadCheck** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiMICRReadCheck

Exécute la lecture MICR pour les chèques.

Syntaxe

int **BiMICRReadCheck** (int nHandle, BYTE readFont, BYTE waitInsertionTime)

Paramètre

nHandle : Spécifie le handle.

readFont : Spécifie la police de lecture.

Valeur	Description
0	E13B
1	CMC7

waitInsertionTime : Spécifie la durée d'attente d'insertion de chèque. Le paramètre par défaut de l'imprimante est 0.

Valeur	Description
0	waitInsertionTime : Désactivé. Continue l'état d'attente de l'insertion du papier chèque.
1 à 15	1 à 15 minutes

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Le temps d'attente pour l'insertion du papier chèque après la lecture MICR est réinitialisé à 0, la valeur par défaut de l'imprimante.

Assurez-vous d'appeler `BiMICRSetReadBackFunction`/`BiMICRSetReadBackFunctionEx` avant d'appeler cette Fonction.

BiMICREjectCheck

Éjecte le chèque.

Syntaxe

int **BiMICREjectCheck** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiMICRCancelWaitCheckInsertion

Annule l'attente d'insertion de chèque.

Syntaxe

int **BiMICRCancelWaitCheckInsertion** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiMICRRetransmissionCheckData

Renvoie les résultats de lecture de chèque.

Syntaxe

int **BiMICRRetransmissionCheckData** (int nHandle, LPBYTE pReadBuffSize, LPBYTE readCharBuff, LPBYTE pStatus, LPBYTE pDetail, DWORD timeout)

Paramètre

nHandle : Spécifie le handle.

pReadBuffSize : Spécifie la taille de la mémoire où les données de lecture sont définies. Après l'exécution de cette fonction, la taille des données de lecture réelles est définie.

readCharBuff : Spécifie les adresses mémoire où les données de lecture sont définies.

pStatus : Spécifie une adresse mémoire de 1 octet où le statut de lecture est défini.

pDetail : Spécifie une adresse mémoire de 1 octet qui définit les informations détaillées après une erreur de lecture retournée dans le cas où l'ajout d'informations détaillées est spécifié par la fonction BiMICRSelectDataHandling.

timeout : Spécifie le temps d'attente pour la lecture des données en millisecondes. Étant donné que ce paramètre est de type DWORD, toutes les valeurs négatives spécifiées sont traitées comme des valeurs positives.

Exemple : Si vous avez spécifié -200, en DWORD, la valeur est FFFFFFF38h, et est traitée comme 4294967096 millisecondes.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Si les données de lecture débordent, le 6^e bit du statut de lecture MICR passe est ON (débordement des données de lecture).

Si cette fonction est appelée lorsque la lecture de chèque n'est pas exécutée par la fonction BiMICRReadCheck, les données de lecture ne sont pas définies et le statut de lecture se termine par une erreur.

BiMICRSelectDataHandling

Sélectionne l'opération de lecture de chèque.

Syntaxe

int **BiMICRSelectDataHandling** (int nHandle, BYTE charSelect, BYTE detailSelect, BYTE errorSelect)

Paramètre

nHandle : Spécifie le handle.

charSelect : Spécifie le traitement des caractères qui ne peuvent pas être analysés.

Valeur	Description
0	Interrompt le traitement d'analyse au point où des caractères qui ne peuvent pas être analysés sont détectés et n'ajoute pas les données de lecture.
1	Remplace les caractères qui ne peuvent pas être analysés par un '?' et continue le traitement d'analyse, puis si la taille des données de lecture est inférieure ou égale à la taille des données de lecture spécifiée dans BiMICRSetReadBackFunction, les données de lecture sont ajoutées.

detailSelect : Spécifie si des informations détaillées doivent être ajoutées après une erreur de lecture.

Valeur	Description
0	Les informations détaillées ne sont pas ajoutées.
1	Les informations détaillées sont ajoutées.

errorSelect : Spécifie si la fonction MICR doit se terminer ou continuer après une erreur. De plus, la fonction MICR continue indépendamment de ce paramètre si la lecture se termine normalement ou si la fonction se termine par une erreur avec detailSelect(1) défini.

Valeur	Description
0	La fonction MICR se termine uniquement lorsque la fonction se termine par une erreur avec detailSelect(1) défini.
1	Si la lecture se termine en raison d'une erreur causée par l'une des causes suivantes, la fonction MICR continue même après la notification des résultats de lecture. • Un chèque de longueur non standard est inséré. • La forme d'onde magnétique ne peut pas être détectée. • Des caractères qui ne peuvent pas être analysés sont détectés lors du traitement d'analyse. • Des erreurs ont été détectées dans les mesures de bruit.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_TIMEOUT	-70	Erreur de temporisation
ERR_ACCESS	-80	L'accès ne peut pas être effectué sur l'imprimante
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture
ERR_EXEC_SCAN	-330	Impossible d'appeler, car le scanner est en cours de numérisation
ERR_PH_NOT_EXIST	-340	Le PortHandler ne s'exécute pas ou une erreur de communication est survenue entre le client du PortHandler et le serveur.
ERR_RESET	-400	Impossible d'appeler, car l'imprimante redémarre
ERR_LOCKED	-1000	L'imprimante est verrouillée.



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

BiMICRSetReadBackFunction

Exécute la lecture de chèques par BiMICRReadCheck et enregistre l'adresse de la fonction de rappel lorsque les résultats sont notifiés, ainsi que les adresses mémoire où chaque type d'informations lues à partir du chèque est défini.

Syntaxe

```
int BiMICRSetReadBackFunction
(int nHandle, INT (CALLBACK EXPORT *pMicrCB)(VOID), LPBYTE
pReadBuffSize, LPBYTE readCharBuff, LPBYTE pStatus, LPBYTE pDetail)
```

Paramètre

nHandle : Spécifie le handle.

(CALLBACK EXPORT * pMicrCB)(VOID) :

Spécifie l'adresse de la fonction de rappel pour la notification des résultats de la lecture d'un chèque.

pReadBuffSize : Spécifie la taille de la mémoire où les données de lecture sont définies. Après l'exécution de cette fonction, la taille des données qui ont été réellement lues est définie.

readCharBuff : Spécifie l'adresse mémoire où les données de lecture du chèque sont définies.

pStatus : Spécifie une adresse mémoire de 1 octet où le statut de lecture MICR est enregistré.

pDetail : Spécifie une adresse mémoire de 1 octet dans laquelle la lecture d'un chèque se termine par une erreur, qui est retournée dans les cas où des informations détaillées sont ajoutées conformément à la fonction BiMICRSelectDataHandling.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXIST	-210	Les données spécifiées existent déjà.
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Pour les informations MICR et le statut de lecture MICR de chaque imprimante TM, reportez-vous aux "Spécifications de l'imprimante".

BiMICRSetReadBackFunctionEX

Exécute la lecture de chèques avec la fonction BiMICRReadCheck et la fonction d'enregistrement de l'adresse de rappel à appeler lorsque les résultats sont notifiés, ainsi que l'adresse mémoire où chaque type d'informations lues doit être défini. Renvoie également le pointeur vers le nom du port pour reconnaître de quel port provient le rappel.

Syntaxe

```
int BiMICRSetReadBackFunctionEx
(int nHandle, INT (CALLBACK EXPORT *pMircCB)(LPSTR lpcPortName),
LPBYTE pReadBuffSize, LPBYTE readCharBuff, LPBYTE pStatus, LPBYTE
pDetail)
```

Paramètre

nHandle : Spécifie le handle.

(CALLBACK EXPORT * pMircCB)(LPSTR lpcPortName):

Spécifie l'adresse de la fonction de rappel pour la notification des résultats de la lecture d'un chèque. Définit le nom du port dans lpcPortName et appelle la fonction de rappel.

pReadBuffSize : Spécifie la taille de la mémoire où les données de lecture sont définies. Après l'exécution de cette fonction, la taille des données qui ont été réellement lues est définie.

readCharBuff : Spécifie l'adresse mémoire où les données de lecture du chèque sont définies.

pStatus : Spécifie une adresse mémoire de 1 octet où le statut de lecture MICR est enregistré.

pDetail : Spécifie une adresse mémoire de 1 octet dans laquelle la lecture d'un chèque se termine par une erreur, qui est retournée dans les cas où des informations détaillées sont ajoutées conformément à la fonction BiMICRSelectDataHandling.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_PARAM	-90	Erreur de paramètre
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXIST	-210	Les données spécifiées existent déjà.
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Commentaire

Assurez-vous d'appeler cette fonction avant d'appeler la fonction BiMICRReadCheck. L'appel de la fonction BiMICRReadCheck avant l'appel de cette fonction renvoie une erreur.

Si les données de lecture débordent, le 6e bit du statut de lecture MICR est ON (débordement des données de lecture).

Pour les informations MICR et le statut de lecture MICR de chaque imprimante TM, reportez-vous aux "Spécifications de l'imprimante".

BiMICRCancelReadBack

Annule une demande de notification d'informations de lecture enregistrée à l'aide de BiMICRSetReadBackFunction ou BiMICRSetReadBackFunctionEx.

Syntaxe

int **BiMICRCancelReadBack** (int nHandle)

Paramètre

nHandle : Spécifie le handle.

Valeur de retour

Définition de macro (constante)	Valeur	Description
SUCCESS	0	Succès
ERR_HANDLE	-60	Le handle spécifié est invalide
ERR_NOT_SUPPORT	-100	Non pris en charge
ERR_EXEC_FUNCTION	-310	Impossible d'appeler, car une autre Status API est utilisée
ERR_EXEC_MICR	-320	Impossible d'appeler, car le MICR est en cours de lecture



REMARQUE

Pour les informations sur les remèdes aux erreurs d'exécution de Status API, reportez-vous à ["Erreur d'exécution de Status API" à la page 23](#).

Référence pour .NET

Ce chapitre explique Status API et la syntaxe utilisée dans l'environnement .NET.

Propriétés

IsValid

Acquiert le statut d'ouverture de l'imprimante.

Accès : Lecture seule

Type de données : System.Boolean

Explication

Renvoie l'une des valeurs suivantes.

true : Ouvert avec succès.

false : Non ouvert ou échec de l'ouverture.

LastError

Acquiert le code d'erreur de la dernière API exécutée.

Accès : Lecture seule

Type de données : com.epson.pos.driver.ErrorCode

Explication

Peut acquérir un code d'erreur à tout moment, car ce module conserve la dernière API exécutée.

Cette méthode est utilisée pour juger du succès ou de l'échec car les API implémentées dans les propriétés ne peuvent pas renvoyer de codes d'erreur.

Codes d'erreur pour les erreurs pouvant survenir dans toutes les API. Pour plus de détails, voir ["Erreur d'exécution de Status API" à la page 23](#).

État

Acquiert l'état actuel de l'imprimante.

Accès : Lecture seule

Type de données : com.epson.pos.driver.ASB

Explication

Les constantes définies dans com.epson.pos.driver.ASB doivent être utilisées pour la valeur. Pour plus de détails, voir ["BiGetStatus" à la page 37](#).

AutoPowerOffTime

Acquiert et définit le temps restant avant que l'imprimante mobile ne s'éteigne. Pour plus de détails, voir ["BiAutoPowerOffTime" à la page 45](#).

Accès : Lecture/Écriture

Type de données :int

Méthodes

OpenMonPrinter

Démarre le contrôle de l'imprimante spécifiée. Pour plus de détails, voir ["BiOpenMonPrinter" à la page 25](#).

Syntaxe

ErrorCode **OpenMonPrinter** (OpenType type, String name)

Paramètres

OpenType type : Type de nom à spécifier pour le nom. Les constantes définies dans `com.epson.pos.driver.OpenType` doivent être utilisées pour la valeur.

String name : Démarre le contrôle de l'imprimante spécifiée.

CloseMonPrinter

Arrête le contrôle de l'imprimante spécifiée. Pour plus de détails, voir ["BiCloseMonPrinter" à la page 27](#).

Syntaxe

ErrorCode **CloseMonPrinter** ()

LockPrinter

Occupe l'imprimante. Pour plus de détails, voir ["BiLockPrinter" à la page 28](#).

Syntaxe

ErrorCode **LockPrinter** (int timeout)

Paramètres

int timeout : Temps d'attente (en ms).

UnlockPrinter

Arrête d'occuper l'imprimante. Pour plus de détails, voir ["BiUnlockPrinter" à la page 30](#).

Syntaxe

ErrorCode **UnlockPrinter** ()

DirectIOEx

Après l'envoi des données spécifiées à l'imprimante, reçoit des données de la longueur spécifiée de l'imprimante. Pour plus de détails, voir ["BiDirectIOEx" à la page 31](#).

Syntaxe

- ErrorCode **DirectIOEx** (byte[] writeCmd, ref byte[] readBuff, int timeout, bool nullTerminate, byte option)

Description : Envoie les commandes ESC/POS à l'imprimante TM et reçoit le résultat d'exécution (données binaires) de l'imprimante.

- ErrorCode **DirectIOEx** (byte[] writeCmd, out String response, int timeout, byte option)

Description : Envoie uniquement les commandes ESC/POS à l'imprimante TM et reçoit le résultat d'exécution (données de type chaîne de caractères) de l'imprimante.

- ErrorCode **DirectIOEx** (byte[] writeCmd, int timeout)

Description : Envoie uniquement les commandes ESC/POS à l'imprimante TM. Ne reçoit ni le résultat d'exécution ni les statuts ASB de l'imprimante.

Paramètres

byte[] writeCmd :	Données à envoyer à l'imprimante
ref byte[] readBuff :	Données reçues de l'imprimante
int timeout :	Temps d'attente pour la transmission et la réception des données (en ms)
bool nullTerminate :	Dans le cas de "True", la lecture est terminée lorsque NULL est reçu de l'imprimante. À ce moment, spécifiez le format de readBuff pour readLen. Dans le cas de "False", la longueur des données spécifiée dans readLen est lue ou les données sont lues de l'imprimante jusqu'à ce qu'une erreur de temporisation soit générée.
byte option :	Contrôle la commande de suppression ASB.
out String response :	Données reçues de l'imprimante (à convertir en chaînes)

ResetPrinter

Réinitialise l'imprimante. Lors de la réinitialisation de l'imprimante pendant l'impression, annule les travaux d'impression et effectue la réinitialisation de l'imprimante. Pour plus de détails, voir ["BiResetPrinter" à la page 34](#).

Syntaxe

ErrorCode **ResetPrinter**()

ForceResetPrinter

Force la réinitialisation de l'imprimante TM dont le statut est surveillé.

Peut réinitialiser également les imprimantes TM occupées avec LockPrinter. Ceci réinitialise également les imprimantes TM pendant l'impression. Faites preuve de prudence en utilisant cette API. Pour plus de détails, voir ["BiForceResetPrinter" à la page 35](#).

Syntaxe

ErrorCode **ForceResetPrinter**()

GetType

Acquiert le type ID de police de l'imprimante. Pour plus de détails, voir ["BiGetType" à la page 36](#).

Syntaxe

ErrorCode **GetType** (out byte typeid, out byte font, out byte exrom, out byte euspecial)

Paramètres

out byte typeid :	Type ID de l'imprimante
out byte font :	Polices installées dans l'imprimante
out byte exrom :	Capacité de la ROM Flash étendue de l'imprimante.
out byte euspecial :	ID spécial de l'imprimante

SetStatusBack

Démarre la notification de statut via les événements StatusCallback/StatusCallbackEx.

Pour plus de détails, voir ["BiSetStatusBackFunctionEx" à la page 39](#).

Syntaxe

ErrorCode **SetStatusBack** ()

CancelStatusBack

Arrête la notification de statut via les événements StatusCallback/StatusCallbackEx.

Pour plus de détails, voir ["BiCancelStatusBack" à la page 40](#).

Syntaxe

ErrorCode **CancelStatusBack** ()

GetBatteryStatus

Acquiert l'état actuel de la batterie de l'imprimante.

Pour plus de détails, voir ["BiGetBatteryStatus" à la page 41](#).

Syntaxe

ErrorCode **GetBatteryStatus** (ref PowerStatus lpbPowerStatus, ref BatteryStatus lpbBatteryStatus)

SetBatteryStatusBack

Démarre la notification de l'état de la batterie via un événement BatteryStatusCallbackEx.

Pour plus de détails, voir ["BiSetBatteryStatusBackFunctionEx" à la page 42](#).

Syntaxe

ErrorCode **SetBatteryStatusBack** ()

CancelBatteryStatusBack

Arrête la notification de l'état de la batterie via un événement `BatteryStatusCallbackEx`.

Pour plus de détails, voir ["BiCancelBatteryStatusBack" à la page 43](#).

Syntaxe

ErrorCode ***CancelBatteryStatusBack*** ()

PowerOff

Exécute le processus de mise hors tension de l'imprimante. Pour plus de détails, voir ["BiPowerOff" à la page 44](#).

Syntaxe

ErrorCode ***PowerOff*** ()

GetPrnCapability

Acquiert les informations sur l'imprimante spécifiée par l'ID de l'imprimante.

Pour plus de détails, voir ["BiGetPrnCapability" à la page 46](#).

Syntaxe

- ErrorCode ***GetPrnCapability*** (byte printerID, out byte[] data)

Description : Acquiert les informations (données binaires) de l'imprimante TM spécifiée avec l'ID d'imprimante.

- ErrorCode ***GetPrnCapability*** (byte printerID, out String data)

Description : Acquiert les informations (données de type chaîne de caractères) de l'imprimante TM spécifiée avec l'ID d'imprimante.

Paramètres

byte printerID :	ID de l'imprimante dont les informations sont acquises.
out byte[] data :	Informations de l'imprimante
out String data :	Informations de l'imprimante
	Le type de données diffère selon les informations de l'imprimante TM (ID d'imprimante) à acquérir.

OpenDrawer

Active le tiroir. Peut être utilisé même lorsque l'imprimante est hors ligne. Pour plus de détails, voir ["BiOpenDrawer" à la page 47](#).

Syntaxe

ErrorCode **OpenDrawer** (Drawer drawer, Pulse pulse)

Paramètres

Drawer drawer : Tiroir à ouvrir. Les constantes définies dans com.epson.pos.driver.Drawer doivent être utilisées pour la valeur.

Pulse pulse : Spécifie le temps pendant lequel le signal de déclenchement du tiroir est activé. Les constantes définies dans com.epson.pos.driver.Pulse doivent être utilisées pour la valeur.

MICRCleaning

Nettoie le mécanisme MICR. Une fois le nettoyage terminé, la fonction MICR est désélectionnée. Pour plus de détails, voir ["BiMICRCleaning" à la page 49](#).

Syntaxe

ErrorCode **MICRCleaning** ()

MICRGetStatus

Acquiert le statut MICR. Pour plus de détails, voir ["BiMICRGetStatus" à la page 50](#).

Syntaxe

ErrorCode **MICRGetStatus** (out byte status)

Paramètres

out byte status : Statut MICR

MICRLoadCheck

Charge le bordereau de chèque à la position de début d'impression.

Pour plus de détails, voir ["BiMICRLoadCheck" à la page 51](#).

Syntaxe

ErrorCode ***MICRLoadCheck*** ()

MICRReadCheck

Lit le bordereau de chèque.

Pour plus de détails, voir ["BiMICRReadCheck" à la page 52](#).

Syntaxe

ErrorCode ***MICRReadCheck*** (MicrFont readFont, int waitInsertionTime)

Paramètres

MicrFont readFont : Police de lecture

int waitInsertionTime : Temps d'attente d'insertion du bordereau de chèque

MICREjectCheck

Éjecte le bordereau de chèque.

Pour plus de détails, voir ["BiMICREjectCheck" à la page 54](#).

Syntaxe

ErrorCode ***MICREjectCheck*** ()

MICRCancelWaitCheckInsertion

Annule le statut d'attente d'insertion du bordereau de chèque.

Pour plus de détails, voir ["BiMICRCancelWaitCheckInsertion" à la page 55](#).

Syntaxe

ErrorCode ***MICRCancelWaitCheckInsertion*** ()

MICRRetransmissionCheckData

Exécute la retransmission des résultats de lecture du bordereau de chèque.

Pour plus de détails, voir ["BiMICRRetransmissionCheckData" à la page 56](#).

Syntaxe

ErrorCode ***MICRRetransmissionCheckData***

(out String response, out MICRStatus status, out MICRDetail detail, int timeout)

Paramètres

out String response :	Données de lecture
out MICRStatus status :	Statut de lecture
out MICRDetail detail :	Informations détaillées sur le statut de lecture
int timeout :	Temps de temporisation (en ms)

MICRSelectDataHandling

Configure l'opération de lecture de bordereau de chèque.

Pour plus de détails, voir ["BiMICRSelectDataHandling" à la page 58](#).

Syntaxe

ErrorCode ***MICRSelectDataHandling***

(CharSelect charSel, DetailSelect detailSel, ErrorSelect errorSel)

Paramètres

CharSelect charSel :	Traitement des caractères qui ne peuvent pas être analysés. Les constantes définies dans com.epson.pos.driver.CharSelect doivent être utilisées pour la valeur.
DetailSelect detailSel :	Disponibilité des informations détaillées en cas de fin anormale de la lecture. Les constantes définies dans com.epson.pos.driver.DetailSelect doivent être utilisées pour la valeur.
ErrorSelect errorSel :	Terminer ou non la fonction MICR en cas de fin anormale. Indépendamment de ce paramètre, la fonction MICR se poursuit en cas de fin normale ou en cas de fin anormale ajoutant les résultats de lecture. Les constantes définies dans com.epson.pos.driver.ErrorSelect doivent être utilisées pour la valeur.

MICRSetReadBack

Démarre la notification de réception des données MICR via les événements MICRReadCallback/MICRReadCallbackEx.

Pour plus de détails, voir ["BiMICRSetReadBackFunction" à la page 60](#), ["BiMICRSetReadBackFunctionEX" à la page 61](#).

Syntaxe

ErrorCode ***MICRSetReadBack*** ()

MICRCancelReadBack

Arrête la notification de réception des données MICR via les événements MICRReadCallback/MICRReadCallbackEx.

Pour plus de détails, voir ["BiMICRCancelReadBack" à la page 63](#).

Syntaxe

ErrorCode ***MICRCancelReadBack*** ()

Événements

StatusCallback

Événement qui gère la notification de statut ASB.

Pour plus de détails, voir ["BiSetStatusBackFunction" à la page 38](#).

Syntaxe

StatusCallbackHandler (ASB asb)

Paramètres

ASB asb : Statut ASB. Pour plus de détails, voir "Spécifications de l'imprimante".

StatusCallbackEx

Événement qui gère la notification de statut ASB. Pour plus de détails, voir ["BiSetStatusBackFunctionEx" à la page 39](#).

Syntaxe

StatusCallbackHandlerEx (ASB asb, String portName)

Paramètres

ASB asb : Statut ASB. Pour plus de détails, voir "Spécifications de l'imprimante".

String portName : Nom du port

BatteryStatusCallbackEx

Événement qui gère la notification de statut d'alimentation. Pour plus de détails, voir ["BiSetBatteryStatusBackFunctionEx" à la page 42](#).

Syntaxe

BatteryStatusCallbackHandlerEx (byte bPowerStatus, byte bBatteryStatus, String
lpcPortName)

Paramètres

bPowerStatus : Statut d'alimentation

bBatteryStatus : État de la batterie

portName : Nom du port

MICRReadCallback

Événement qui gère la notification des données MICR.

Pour plus de détails, voir "[BiMICRSetReadBackFunction](#)" à la page 60.

Syntaxe

MICRReadCallbackHandler (String data, MICRStatus status, MICRDetail detail)

Paramètres

String data :	Données MICR
MICRStatus status :	Statut de lecture. Constantes définies dans com.epson.pos.driver. MICRStatus doivent être utilisées pour la valeur.
MICRDetail detail :	Informations détaillées sur le statut de lecture. Les constantes définies dans com.epson.pos.driver.MICRDetail doivent être utilisées pour la valeur.

MICRReadCallbackEx

Événement qui gère la notification des données MICR.

Syntaxe

MICRReadCallbackHandlerEx
(String data, MICRStatus status, MICRDetail detail, String portName)

Paramètres

String data :	Données MICR
MICRStatus status :	Statut de Lecture. Les constantes définies dans com.epson.pos.driver.MICRStatus doivent être utilisées pour la valeur.
MICRDetail detail :	Informations détaillées sur le statut de lecture. Les constantes définies dans com.epson.pos.driver.MICRDetail doivent être utilisées pour la valeur.
String portName :	Nom du port

Annexe

Mentions

Info-ZIP

"Advanced Printer Driver" incorporate compression code from the Info-ZIP group.

This is version 2009-Jan-02 of the Info-ZIP license. The definitive version of this document should be available at <ftp://ftp.info-zip.org/pub/infozip/license.html> indefinitely and a copy at <http://www.info-zip.org/pub/infozip/license.html>.

Copyright (c) 1990-2009 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals: Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ed Gordon, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Steven M. Schweda, Christian Spieler, Cosmin Truta, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White.

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the above disclaimer and the following restrictions:

1. Redistributions of source code (in whole or in part) must retain the above copyright notice, definition, disclaimer, and this list of conditions.
2. Redistributions in binary form (compiled executables and libraries) must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution. Additional documentation is not needed for executables where a command line license option provides these and a note regarding this option is in the executable's startup banner. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled.
3. Altered versions—including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, versions with modified or added functionality, and dynamic, shared, or static library versions not from Info-ZIP—must be plainly marked as such and must not be misrepresented as being the original source or, if binaries, compiled from the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases—including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or the Info-ZIP URL(s), such as to imply Info-ZIP will provide support for the altered versions.
4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

Windows Template Library

This application uses the Microsoft Windows Template Library (WTL).

Mises en garde

- (1) Aucune partie de ce document ne peut être reproduite, stockée dans un système documentaire ou transmise sous quelque forme ou de quelque manière que ce soit, électronique, mécanique, par photocopie, enregistrement ou autre, sans l'accord préalable écrit de Seiko Epson Corporation.
- (2) Le contenu du présent document peut être modifié sans notification.
- (3) Toutes les précautions ont été prises lors de la préparation du présent document. Seiko Epson Corporation ne peut cependant être tenue responsable en cas d'erreurs ou d'omissions.
- (4) Il en va de même pour les dommages résultant de l'utilisation des informations incluses dans le présent manuel.
- (5) Ni Seiko Epson Corporation ni ses filiales ne pourront être tenues responsables vis à vis de l'acheteur de ce produit ou de tiers en cas de dommages, pertes, frais ou dépenses subis par l'acheteur ou des tiers du fait d'un accident, d'une mauvaise utilisation ou d'un abus d'utilisation de ce produit ou de modifications, de réparations ou d'altérations non autorisées apportées à ce produit ou (à l'exclusion des États-Unis) de l'absence de respect strict des instructions d'utilisation et de maintenance de Seiko Epson Corporation.
- (6) Seiko Epson Corporation ne pourra être tenue responsable des dommages ou problèmes survenus suite à l'utilisation de produits optionnels ou consommables autres que ceux désignés par Seiko Epson Corporation comme étant des Produits d'origine EPSON ou des Produits approuvés par EPSON.

Marques

Microsoft[®], Windows[®], Visual Studio[®], Visual C++[®] et Visual C#[®] sont des marques commerciales du groupe de sociétés Microsoft.

Toutes les autres marques commerciales appartiennent à leurs propriétaires respectifs et sont utilisées à des fins d'identification uniquement.

Système de commande ESC/POS

EPSON ESC/POS est un système de commande d'imprimante POS propriétaire qui comprend des commandes brevetées ou en instance de brevet. ESC/POS est compatible avec la plupart des imprimantes et écrans POS EPSON.

ESC/POS est conçu pour réduire la charge de traitement de l'ordinateur hôte dans un environnement POS. Il comprend un ensemble de commandes très performantes et efficaces, et offre également la flexibilité de procéder facilement à de futures mises à niveau.

©Seiko Epson Corporation 2025