

1 Agent Zabbix

Aperçu

Ces vérifications utilisent la communication avec l'agent Zabbix pour la collecte de données.

Il existe des vérifications [passives et actives](#) d'agents. Lorsque vous configurez un élément, vous pouvez sélectionner le type requis :

- *Zabbix agent* - pour les vérifications passives
- *Zabbix agent (active)* - pour les vérifications actives

Clés d'élément supportées

Le tableau fournit des détails sur les clés d'élément que vous pouvez utiliser avec les éléments d'agent Zabbix.

Voir aussi :

- [Éléments supportés par plateforme](#)
- [Clés d'élément spécifiques à l'agent Zabbix sous Windows](#)
- [Niveau d'autorisation minimum pour les éléments d'agent Windows](#)

Paramètres obligatoires et facultatifs

Les paramètres sans chevrons sont obligatoires. Les paramètres marqués avec des chevrons < > sont facultatifs.

Clé			
Description	Valeur de retour	Paramètres	Commentaires
agent.hostname			
Nom de l'hôte.	Chaîne		Retourne la valeur actuelle correspondant au nom de l'agent dans le fichier de configuration.
agent.ping			
Vérification de la disponibilité de l'agent.	Rien - indisponible 1 - disponible		Utiliser la fonction de déclencheur nodata() pour vérifier l'indisponibilité de l'hôte.
agent.version			
Version de l'agent Zabbix.	Chaîne		Exemple de valeur de retour : 1.8.2
kernel.maxfiles			
Nombre maximum de fichiers ouverts supporté par l'OS.	Entier		
kernel.maxproc			
Nombre maximum de processus supporté par l'OS.	Entier		
log[file,<regex>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Supervision de fichier de log.	Log	<p>file - chemin d'accès complet et nom du fichier de log</p> <p>regexp - expression régulière⁴ décrivant le motif requis</p> <p>encoding - encodage du texte</p> <p>maxlines - nombre maximum de nouvelles lignes par seconde que l'agent enverra au serveur ou proxy Zabbix. Ce paramètre remplace la valeur de 'MaxLinesPerSecond' dans zabbix_agentd.conf</p> <p>mode - valeurs possibles : <i>all</i> (par défaut), <i>skip</i> - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).</p> <p>output - un modèle de mise en forme de sortie optionnel. La séquence d'échappement \0 est remplacée par la partie du texte correspondant (à partir du premier caractère où la correspondance commence jusqu'au dernier caractère où la correspondance se termine) alors qu'une séquence d'échappement \N (où N=1...9) est remplacée par le N^{ième} groupe correspondant (ou une chaîne vide si le N dépasse le nombre de groupes capturés).</p> <p>maxdelay - délai maximum en secondes. Type : flottant. Valeurs : 0 - (par défaut) n'ignore jamais les lignes de fichiers de log ; > 0.0 - ignore les anciennes lignes afin d'obtenir les lignes les plus récentes analysées dans les secondes "maxdelay". Lire les notes maxdelay avant de l'utiliser !</p>	<p>L'élément doit être configuré en tant que vérification active. Si le fichier est manquant ou que les permissions sur le fichier empêche l'accès, l'élément devient non supporté.</p> <p>Si le champ output est laissé vide - la ligne entière contenant la correspondance du texte est retournée. Notez que tous les types d'expressions régulières globales sauf 'Résultat est VRAI' retournent toujours toute la ligne correspondante et le paramètre output est ignoré.</p> <p>L'extraction de contenu à l'aide du paramètre output a lieu sur l'agent.</p> <p>Exemples : ⇒ log[/var/log/syslog] ⇒ log[/var/log/syslog,error] ⇒ log[/home/zabbix/logs/logfile,,,100]</p> <p><i>En utilisant le paramètre <code>output</code> pour extraire un nombre depuis un enregistrement du journal :</i> ⇒ log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,1] → correspondra à l'enregistrement "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" et renverra seulement '6080' au serveur. Car une valeur numérique a été envoyée, le "Type d'information" pour cet élément peut être positionné sur "Numérique (non signé)" et la valeur pourra être utilisée dans les graphiques, les déclencheurs etc.</p> <p><i>En utilisant le paramètre <code>output</code> pour transformer l'enregistrement avant d'être envoyé au serveur :</i> ⇒ log[/app1/app.log,"([0-9 :]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,1 RECORDS: \3, ERRORS: \4, DURATION: \2"] → correspondra à l'enregistrement "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" et envoie l'enregistrement modifié "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08" au serveur.</p> <p>Le paramètre mode est supporté depuis Zabbix 2.0. Le paramètre output est supporté depuis Zabbix 2.2. Le paramètre maxdelay est supporté depuis Zabbix 3.2.</p> <p>Voir aussi des informations supplémentaires sur la supervision des fichiers de log.</p>
log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>]			
Compte le nombre de lignes qui correspondent dans le fichier de log.	Entier	<p>file - chemin d'accès complet et nom du fichier de log</p> <p>regexp - expression régulière⁴ décrivant le modèle requis</p> <p>encoding - encodage du texte</p> <p>maxproclines - nombre maximum de nouvelles lignes par seconde que l'agent va analyser. La valeur par défaut est 10*'MaxLinesPerSecond' dans zabbix_agentd.conf.</p> <p>mode - valeurs possibles : <i>all</i> (par défaut), <i>skip</i> - ignorer le traitement des anciennes valeurs (affecte uniquement les éléments nouvellement créés).</p> <p>maxdelay - délai maximum en secondes. Type : flottant. Valeurs : 0 - (par défaut) n'ignore jamais les lignes du fichier de log ; > 0.0 - ignore les anciennes lignes afin d'obtenir les lignes les plus récentes analysées dans les "maxdelay" secondes. Lire les notes concernant maxdelay avant de l'utiliser !</p>	<p>L'élément doit être configuré en tant que vérification active. Si le fichier est manquant ou que les permissions sur le fichier en empêche l'accès, l'élément devient non supporté.</p> <p>Voir aussi les informations supplémentaires sur la supervision des fichiers de log.</p> <p>Cet élément n'est pas supporté pour les journaux d'événements Windows.</p> <p>Supporté depuis Zabbix 3.2.0.</p>
logrt[file_regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Surveillance du fichier de log avec prise en charge de la rotation.	Log	<p>file_regexp - chemin absolu vers le fichier et l'expression régulière décrivant le modèle de nom de fichier</p> <p>regexp - expression régulière⁴ décrivant le motif requis</p> <p>encoding - encodage du texte</p> <p>maxlines - nombre maximum de nouvelles lignes par seconde que l'agent enverra au serveur ou proxy Zabbix. Ce paramètre remplace la valeur de 'MaxLinesPerSecond' dans zabbix_agentd.conf</p> <p>mode - valeurs possibles : <i>all</i> (par défaut), <i>skip</i> - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).</p> <p>output - un modèle de mise en forme de sortie optionnel. La séquence d'échappement \O est remplacée par la partie du texte correspondant (à partir du premier caractère où la correspondance commence jusqu'au dernier caractère où la correspondance se termine) alors qu'une séquence d'échappement \N (où N=1...9) est remplacée par le Nième groupe correspondant (ou une chaîne vide si N dépasse le nombre de groupes capturés).</p> <p>maxdelay - délai maximum en secondes. Type : flottant. Valeurs : 0 - (par défaut) n'ignore jamais les lignes de fichier de log ; > 0.0 - ignore les anciennes lignes afin d'obtenir les lignes les plus récentes analysées dans les "maxdelay" secondes. Lire les notes liées à maxdelay avant de l'utiliser !</p> <p>options - type de rotation de fichier de log file rotation. Valeurs possibles : <i>rotate</i> (par défaut), <i>copytruncate</i>. Notez que <i>copytruncate</i> ne peut pas être utilisé avec <i>maxdelay</i>. Dans ce cas <i>maxdelay</i> doit être 0 ou non spécifié. Voir les notes liées à copytruncate.</p>	<p>L'élément doit être configuré en tant que vérification active. La rotation de fichier de log est basée sur la date de dernière modification des fichiers.</p> <p>Si le paramètre output est laissé vide la ligne entière contenant le texte recherché est retourné. Notez que tous les types d'expressions régulières globales sauf 'Résultat est VRAI' retournent toujours toute la ligne correspondante et le paramètre output est ignoré.</p> <p>L'extraction de contenu à l'aide du paramètre output a lieu sur l'agent.</p> <p>Exemples : ⇒ logrt["/home/zabbix/logs/^logfile[0-9]{1,3}\$",,,100] → correspondra à un fichier comme "logfile1" (ne correspondra pas à ".logfile1") ⇒ logrt["/home/user/^logfile_*_[0-9]{1,3}\$","pattern_to_match","UTF-8",100] → collectera les données d'un fichier comme "logfile_abc_1" ou "logfile_001".</p> <p><i>En utilisant le paramètre output pour extraire un nombre de l'enregistrement :</i> ⇒ logrt[/app1/^test.*log\$, "task run [0-9.]+ sec, processed ([0-9.]+) records, [0-9.]+ errors",,,\1] → correspondra à l'enregistrement "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" et renverra seulement '6080' au serveur. Car une valeur numérique a été envoyée, le "Type d'information" pour cet élément peut être positionné sur "Numérique (non signé)" et la valeur pourra être utilisée dans les graphiques, les déclencheurs, etc.</p> <p><i>En utilisant le paramètre output pour transformer un enregistrement avant d'être envoyé au serveur :</i> ⇒ logrt[/app1/^test.*log\$, "([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9.]+) records, ([0-9.]+) errors",,,\1 RECORDS: \3, ERRORS: \4, DURATION: \2"] → correspondra à l'enregistrement "2015-11-13 10:08:26 task run 6.08 sec, processed 6080 records, 0 errors" et envoie l'enregistrement modifié "2015-11-13 10:08:26 RECORDS: 6080, ERRORS: 0, DURATION: 6.08" au serveur.</p> <p>Le paramètre mode est supporté depuis Zabbix 2.0. Le paramètre output est supporté depuis Zabbix 2.2. Le paramètre maxdelay est supporté depuis Zabbix 3.2. Le paramètre options est supporté depuis Zabbix 4.0.</p> <p>Voir aussi des informations supplémentaires sur la supervision des fichiers de log.</p>
logrt.count[file_regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Nombre de lignes correspondantes dans la surveillance des fichiers de logs avec prise en charge de la rotation.	Entier	<p>file_regexp - chemin absolu vers le fichier et l'expression régulière⁴ décrivant le modèle de nom de fichier</p> <p>regexp - expression régulière⁴ décrivant le modèle de contenu requis</p> <p>encoding - encodage du texte</p> <p>maxproclines - nombre maximum de nouvelles lignes par seconde que l'agent va analyser. La valeur par défaut est 10*'MaxLinesPerSecond' dans zabbix_agentd.conf.</p> <p>mode - valeurs possibles: <i>all</i> (par défaut), <i>skip</i> - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).</p> <p>maxdelay - délai maximum en secondes. Type: float. Valeurs : 0 - (par défaut) n'ignore jamais les lignes de fichier journal ; > 0.0 - ignore les anciennes lignes afin d'obtenir les lignes les plus récentes analysées dans les secondes "maxdelay". Lire les notes de maxdelay avant de l'utiliser!</p> <p>options - type de rotation de fichier de log. Valeurs possibles : <i>rotate</i> (par défaut), <i>copytruncate</i>. Notez que <i>copytruncate</i> ne peut pas être utilisé avec <i>maxdelay</i>. Dans ce cas <i>maxdelay</i> doit être 0 ou non spécifié. Voir les notes liées à copytruncate.</p>	<p>L'élément doit être configuré en tant que vérification active. La rotation de fichier de log est basée sur la date de dernière modification des fichiers.</p> <p>Le paramètre <code>options</code> est supporté depuis Zabbix 4.0.</p> <p>Cet élément n'est pas supporté pour les journaux d'événements Windows.</p> <p>Supporté depuis Zabbix 3.2.0.</p>
net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]			
Vérifie si le service DNS fonctionne.	<p>0 - le DNS est arrêté (le serveur n'a pas répondu ou la résolution DNS a échoué)</p> <p>1 - le DNS est OK</p>	<p>ip - adresse IP du serveur DNS (laissez vide pour le serveur DNS par défaut, ignoré sous Windows)</p> <p>name - nom du serveur DNS à interroger</p> <p>type - type d'enregistrement à interroger (par défaut, <i>SOA</i>)</p> <p>timeout (ignoré sous Windows) - délai d'expiration de la requête en secondes (la valeur par défaut est 1 seconde)</p> <p>count (ignoré sous Windows) - nombre d'essais pour la requête (la valeur par défaut est 2)</p> <p>protocol - le protocole utilisé pour effectuer des requêtes DNS : <i>udp</i> (par défaut) ou <i>tcp</i></p>	<p>Exemple: ⇒ <code>net.dns[8.8.8.8,zabbix.com,MX,2,1]</code></p> <p>Les valeurs possibles pour <code>type</code> sont : <i>ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS</i> (sauf pour Windows), <i>HINFO, MINFO, TXT, SRV</i>.</p> <p>Les noms de domaine internationalisés ne sont pas supportés, veuillez utiliser les noms codés IDNA à la place.</p> <p>Le paramètre <code>protocol</code> est supporté depuis Zabbix 3.0. Le type d'enregistrement <i>SRV</i> est pris en charge depuis les versions d'agent Zabbix 1.8.6 (Unix) et 2.0.0 (Windows).</p> <p>Nom avant Zabbix 2.0 (toujours supporté) : <code>net.tcp.dns</code></p>
net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]			
Effectue une requête DNS.	Chaîne de caractères avec le type d'information demandé	<p>ip - adresse IP du serveur DNS (laissez vide pour le serveur DNS par défaut, ignoré sous Windows)</p> <p>name - nom DNS à interroger</p> <p>type - type d'enregistrement à interroger (par défaut, <i>SOA</i>)</p> <p>timeout (ignoré sous Windows) - délai d'expiration de la requête en secondes (la valeur par défaut est 1 seconde)</p> <p>count (ignoré sous Windows) - nombre d'essais pour la requête (la valeur par défaut est 2)</p> <p>protocol - le protocole utilisé pour effectuer des requêtes DNS : <i>udp</i> (par défaut) ou <i>tcp</i></p>	<p>Exemple: ⇒ <code>net.dns.record[8.8.8.8,zabbix.com,MX,2,1]</code></p> <p>Les valeurs possible pour <code>type</code> sont : <i>ANY, A, NS, CNAME, MB, MG, MR, PTR, MD, MF, MX, SOA, NULL, WKS</i> (sauf pour Windows), <i>HINFO, MINFO, TXT, SRV</i></p> <p>Les noms de domaine internationalisés ne sont pas supportés, veuillez utiliser les noms codés IDNA à la place.</p> <p>Le paramètre <code>protocol</code> est supporté depuis Zabbix 3.0. Le type d'enregistrement <i>SRV</i> est pris en charge depuis les versions d'agent Zabbix 1.8.6 (Unix) et 2.0.0 (Windows).</p> <p>Nom avant Zabbix 2.0 (toujours supporté) : <code>net.tcp.dns.query</code></p>
net.if.collisions[if]			
Nombre de collisions hors de la fenêtre.	Entier	if - nom de l'interface réseau	
net.if.discovery			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Liste des interfaces réseaux. Utilisé pour la découverte de bas niveau.	Objet JSON		Supporté depuis l'agent Zabbix version 2.0. Sur FreeBSD, OpenBSD et NetBSD supporté depuis l'agent Zabbix version 2.2. Certaines versions Windows (par exemple, Server 2008) peuvent nécessiter l'installation des dernières mises à jour pour prendre en charge les caractères non ASCII dans les noms d'interface.
net.if.in[if,<mode>]			
Statistiques du trafic entrant sur l'interface réseau.	Entier	if - nom de l'interface réseau (Unix); Description complète de l'interface réseau ou adresse IPv4 (Windows) mode - valeurs possibles : <i>bytes</i> - nombre d'octets (par défaut) <i>packets</i> - nombre de paquets <i>errors</i> - nombre d'erreurs <i>dropped</i> - nombre de paquets perdus <i>overruns (fifo)</i> - le nombre d'erreur du buffer FIFO <i>frame</i> - le nombre d'erreurs de trames de paquets <i>compressed</i> - le nombre de paquets compressés transmis ou reçus par le pilote de périphérique <i>multicast</i> - le nombre de trames de multidiffusion reçues par le pilote de périphérique	Sous Windows, l'élément obtient des valeurs à partir des compteurs 64 bits, si disponibles. Les compteurs de statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Les noms d'interface multi-octets sous Windows sont pris en charge depuis la version de l'agent Zabbix 1.8.6. Exemples : ⇒ net.if.in[eth0,errors] ⇒ net.if.in[eth0] Vous pouvez obtenir des descriptions d'interface réseau sur Windows avec les éléments net.if.discovery et net.if.list. Vous pouvez utiliser cette clé lors de l'étape de pré-traitement <i>Change per second</i> pour obtenir des statistiques en octets par seconde.
net.if.out[if,<mode>]			
Statistiques du trafic sortant sur l'interface réseau.	Entier	if - nom de l'interface réseau (Unix); Description complète de l'interface réseau ou adresse IPv4 (Windows) mode - valeurs possibles : <i>bytes</i> - nombre d'octets (par défaut) <i>packets</i> - nombre de paquets <i>errors</i> - nombre d'erreurs <i>dropped</i> - nombre de paquets perdus <i>overruns (fifo)</i> - le nombre d'erreur du buffer FIFO <i>collisions (colls)</i> - le nombre de collision détectées sur l'interface <i>carrier</i> - le nombre de pertes carrier détectées par le pilote de périphérique <i>compressed</i> - le nombre de paquets compressés transmis par le pilote de périphérique	Sous Windows, l'élément obtient des valeurs à partir des compteurs 64 bits, si disponibles. Les compteurs statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Les noms d'interface multi-octets sous Windows sont pris en charge depuis la version de l'agent Zabbix 1.8.6. Exemples : ⇒ net.if.out[eth0,errors] ⇒ net.if.out[eth0] Vous pouvez obtenir des descriptions d'interface réseau sur Windows avec net.if.discovery ou net.if.list items. Vous pouvez utiliser cette clé lors de l'étape de pré-traitement <i>Change per second</i> pour obtenir des statistiques en octets par seconde.
net.if.total[if,<mode>]			
Somme des statistiques du trafic entrant et sortant sur l'interface réseau.	Entier	if - nom de l'interface réseau (Unix); Description complète de l'interface réseau ou adresse IPv4 (Windows) mode - valeurs possibles : <i>bytes</i> - nombre d'octets (par défaut) <i>packets</i> - nombre de paquets <i>errors</i> - nombre d'erreurs <i>dropped</i> - nombre de paquets perdus <i>overruns (fifo)</i> - le nombre d'erreur du buffer FIFO <i>compressed</i> - le nombre de paquets compressés transmis par le pilote de périphérique	Sous Windows, l'élément obtient des valeurs à partir des compteurs 64 bits, si disponibles. Les compteurs statistiques d'interface 64 bits ont été introduits dans Windows Vista et Windows Server 2008. Si les compteurs 64 bits ne sont pas disponibles, l'agent utilise des compteurs 32 bits. Exemples : ⇒ net.if.total[eth0,errors] ⇒ net.if.total[eth0] Vous pouvez obtenir des descriptions d'interface réseau sur Windows avec net.if.discovery ou net.if.list items. Vous pouvez utiliser cette clé lors de l'étape de pré-traitement <i>Change per second</i> pour obtenir des statistiques en octets par seconde. Notez que les paquets perdus ne sont pris en charge que si net.if.in et net.if.out fonctionnent pour les paquets perdus sur votre plate-forme.
net.tcp.listen[port]			
Vérifie si ce port TCP est dans l'état LISTEN.	0 - il n'est pas dans l'état d'écoute 1 - il est en état d'écoute	port - numéro du port TCP	Exemple : ⇒ net.tcp.listen[80] Sous Linux supporté depuis l'agent Zabbix version 1.8.4 Depuis Zabbix 3.0.0, sur les noyaux Linux 2.6.14 et supérieurs, les informations sur l'écoute des sockets TCP sont obtenues à partir de l'interface NETLINK du noyau, si possible. Sinon, les informations sont récupérées à partir des fichiers /proc/net/tcp et /proc/net/tcp6.
net.tcp.port[<ip>,<port>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Vérifie s'il est possible d'établir une connexion TCP avec le port spécifié.	0 - connexion impossible 1 - connexion possible	ip - adresse IP (par défaut 127.0.0.1) port - numéro de port	Exemple : ⇒ net.tcp.port[80] → peut être utilisé pour tester la disponibilité du serveur Web exécuté sur le port 80. Pour des tests de performances TCP simples, utilisez net.tcp.service.perf[tcp,<ip>,<port>] Notez que ces vérifications peuvent entraîner des messages supplémentaires dans les fichiers de log du démon système (les sessions SMTP et SSH étant généralement loggées). Ancien nom : <i>check_port[*]</i>
net.tcp.service[service,<ip>,<port>]			
Vérifie si le service est en cours d'exécution et accepte les connexions TCP.	0 - le service est arrêté 1 - le service fonctionne	service - l'un parmi : <i>ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (voir les détails) ip - adresse IP (par défaut 127.0.0.1) port - numéro du port (par défaut, le numéro de port du service standard est utilisé)	Exemple : ⇒ net.tcp.service[ftp,,45] → peut être utilisé pour tester la disponibilité du serveur FTP sur le port TCP 45. Notez que ces vérifications peuvent entraîner des messages supplémentaires dans les fichiers de log du démon système (les sessions SMTP et SSH étant généralement loggées). La vérification des protocoles cryptés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas supportée. Pour contourner ce problème, utilisez net.tcp.port pour les vérifications similaires. La vérification de l'agent LDAP et HTTPS par Windows n'est actuellement pas prise en charge. Notez que la vérification de telnet recherche une invite de connexion (avec ':' à la fin). Voir aussi les problèmes connus concernant la vérification du service HTTPS. Les services <i>https</i> et <i>telnet</i> sont supportés depuis Zabbix 2.0. Ancien nom : <i>check_service[*]</i>
net.tcp.service.perf[service,<ip>,<port>]			
Vérifie les performances du service TCP.	0 - le service est arrêté secondes - le nombre de secondes passées à se connecter au service	service - l'un parmi : <i>ssh, ldap, smtp, ftp, http, pop, nntp, imap, tcp, https, telnet</i> (voir les détails) ip - adresse IP (défaut 127.0.0.1) port - numéro du port (par défaut, le numéro de port de service standard est utilisé)	Exemple : ⇒ net.tcp.service.perf[ssh] → peut être utilisé pour tester la vitesse de la réponse initiale du serveur SSH. La vérification des protocoles cryptés (comme IMAP sur le port 993 ou POP sur le port 995) n'est actuellement pas prise en charge. Pour résoudre ce problème, utilisez net.tcp.service.perf[tcp,<ip>,<port>] pour les vérifications de ce type. La vérification de l'agent LDAP et HTTPS par Windows n'est actuellement pas prise en charge. Notez que la vérification de telnet recherche une invite de connexion (avec ':' à la fin). Voir aussi les problèmes connus concernant la vérification du service HTTPS. Les services <i>https</i> et <i>telnet</i> sont supportés depuis Zabbix 2.0 Ancien nom : <i>check_service_perf[*]</i>
net.udp.listen[port]			
Vérifie si ce port UDP est en état LISTEN.	0 - il n'est pas en état LISTEN 1 - il est en état LISTEN	port - numéro de port UDP	Exemple : ⇒ net.udp.listen[68] Sur Linux, supporté depuis l'agent Zabbix en version 1.8.4
net.udp.service[service,<ip>,<port>]			
Vérifie si le service est en cours d'exécution et répond aux demandes UDP.	0 - le service est arrêté 1 - le service fonctionne	service - <i>nntp</i> (voir les détails) ip - adresse IP (par défaut 127.0.0.1) port - numéro du port (par défaut, le numéro de port du service standard est utilisé)	Exemple : ⇒ net.udp.service[nntp,,45] → peut être utilisé pour tester la disponibilité du service NTP sur le port UDP 45. Cet élément est supporté depuis Zabbix 3.0.0, mais le service <i>nntp</i> était disponible pour les éléments net.tcp.service[] dans les versions précédentes.
net.udp.service.perf[service,<ip>,<port>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Vérifie les performances du service UDP.	0 - le service est arrêté secondes - le nombre de secondes passées à attendre la réponse du service	service - <i>ntp</i> (voir les détails) ip - adresse IP (par défaut 127.0.0.1) port - numéro du port (par défaut, le numéro de port du service standard est utilisé)	Exemple : ⇒ net.udp.service.perf[ntp] → peut être utilisé pour tester le temps de réponse du service NTP. Cet élément est supporté depuis Zabbix 3.0.0, mais le service <i>ntp</i> était disponible pour l'élément net.tcp.service[] dans les versions précédentes.
proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]			
Pourcentage d'utilisation du processeur	Flottant	name - nom du processus (par défaut <i>tous les processus</i>) username - nom de l'utilisateur (par défaut <i>tous les utilisateurs</i>) type - type d'utilisation CPU : <i>total</i> (défaut), <i>utilisateur</i> , <i>system</i> cmdline - filtrer par ligne de commande (il s'agit d'une expression régulière ⁴) mode - mode de récupération des données : <i>avg1</i> (défaut), <i>avg5</i> , <i>avg15</i> zone - zone cible : <i>current</i> (défaut), <i>all</i> . Ce paramètre est supporté uniquement sur les plateformes Solaris.	Exemples : ⇒ proc.cpu.util[,root] → Utilisation du processeur de tous les processus s'exécutant sous l'utilisateur "root" ⇒ proc.cpu.util[zabbix_server,zabbix] → Utilisation du processeur de tous les processus zabbix_server exécutés sous l'utilisateur zabbix La valeur renvoyée est basée sur le pourcentage d'utilisation du cœur du processeur unique. Par exemple, l'utilisation du processeur d'un processus utilisant deux cœurs est de 200%. Les données d'utilisation du processeur sont recueillies par un collecteur qui prend en charge le maximum de 1024 requêtes uniques (par nom, utilisateur et ligne de commande). Les requêtes non accessibles au cours des dernières 24 heures sont supprimées du collecteur. Cette clé est supportée depuis Zabbix 3.0.0 et est disponible sur plusieurs plateformes (voir les éléments supportés par plateforme).
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]			
Mémoire utilisée par le processus en octets.	Entier - avec mode comme <i>max</i> , <i>min</i> , <i>sum</i> Flottant - avec mode comme <i>avg</i>	name - nom du processus (par défaut <i>all processes</i>) user - nom de l'utilisateur (par défaut <i>all users</i>) mode - valeurs possibles : <i>avg</i> , <i>max</i> , <i>min</i> , <i>sum</i> (défaut) cmdline - filtrer par ligne de commande line (il s'agit d'une expression régulière ⁴) memtype - type de mémoire utilisée par processus	Exemples : ⇒ proc.mem[,root] → mémoire utilisée par tous les processus s'exécutant sous l'utilisateur "root" ⇒ proc.mem[zabbix_server,zabbix] → mémoire utilisée par tous les processus zabbix_server exécutés sous l'utilisateur zabbix ⇒ proc.mem[,oracle,max,oracleZABBIX] → mémoire utilisée par le processus le plus gourmand en mémoire sous oracle ayant oracleZABBIX dans sa ligne de commande <i>Remarque</i> : Lorsque plusieurs processus utilisent la mémoire partagée, la somme de la mémoire utilisée par les processus peut générer des valeurs importantes et irréalistes. Voir les notes sur la sélection des processus avec les paramètres name et cmdLine (spécifiques à Linux). Le paramètre memtype est supporté sur plusieurs plateformes depuis Zabbix 3.0.0.
proc.num[<name>,<user>,<state>,<cmdline>]			
Nombre de processus.	Entier	name - nom du processus (par défaut <i>all processes</i>) user - nom de l'utilisateur (par défaut <i>all users</i>) state - valeurs possibles : <i>all</i> (par défaut), <i>disk</i> - sommeil ininterrompu, <i>run</i> - s'exécute, <i>sleep</i> - sommeil interruptible, <i>trace</i> - arrêté, <i>zomb</i> - zombie cmdline - filtre par ligne de commande (il s'agit d'une expression régulière ⁴)	Exemples : ⇒ proc.num[,mysql] → nombre de processus s'exécutant sous l'utilisateur mysql ⇒ proc.num[apache2,www-data] → nombre de processus apache2 s'exécutant sous l'utilisateur www-data ⇒ proc.num[,oracle,sleep,oracleZABBIX] → nombre de processus en état de veille s'exécutant en tant qu'utilisateur oracle ayant oracleZABBIX dans sa ligne de commande Voir les notes concernant la sélection des processus avec les paramètres name et cmdLine (Spécifique à Linux). Sur Windows, seuls les paramètres name et user sont supportés. Les valeurs de <i>disk</i> et <i>trace</i> pour le paramètre state sont supportés depuis Zabbix 3.4.0.
sensor[device,sensor,<mode>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Lecture d'un capteur matériel.	Flottant	device - nom du périphérique sensor - nom du capteur mode - valeurs possibles : <i>avg, max, min</i> (si ce paramètre est omis, le périphérique et le capteur sont traités textuellement).	Lit <code>/proc/sys/dev/sensors</code> sur Linux 2.4. Exemple : ⇒ <code>sensor[w83781d-i2c-0-2d,temp1]</code> Dans les versions antérieures à Zabbix 1.8.4, le format <code>sensor[temp1]</code> était utilisé. Lit <code>/sys/class/hwmon</code> sur Linux 2.6+. Voir une description plus détaillée de l'élément capteur sur Linux. lit la MIB <code>hw.sensors</code> sur OpenBSD. Exemples : ⇒ <code>sensor[cpu0,temp0]</code> → température d'un CPU ⇒ <code>sensor["cpu[0-2]\$,temp,avg]</code> → température moyenne des trois premiers CPU Supporté sur OpenBSD depuis Zabbix 1.8.4.
system.boottime			
Heure du dernier démarrage.	Entier (horodatage Unix)		
system.cpu.discovery			
Liste des CPU/cœurs de CPU détectés. Utilisé pour la découverte de bas niveau.	objet JSON		Supporté sur toutes les plateformes depuis Zabbix 2.4.0.
system.cpu.intr			
Interruption du périphérique.	Entier		
system.cpu.load[<cpu>,<mode>]			
Charge CPU.	Flottant	cpu - valeurs possibles : <i>all</i> (par défaut), <i>percpu</i> (charge totale divisée par le nombre de processeurs en ligne) mode - valeurs possibles : <i>avg1</i> (par défaut moyenne sur une minute), <i>avg5</i> , <i>avg15</i>	Exemple : ⇒ <code>system.cpu.load[avg5]</code> <i>percpu</i> est supporté depuis Zabbix 2.0.0. Ancien nom : <code>system.cpu.loadX</code>
system.cpu.num[<type>]			
Nombre de CPUs.	Entier	type - valeurs possibles : <i>online</i> (par défaut), <i>max</i>	Exemple : ⇒ <code>system.cpu.num</code>
system.cpu.switches			
Nombre de commutateurs de contexte.	Entier		Ancien nom : <code>system[switches]</code>
system.cpu.util[<cpu>,<type>,<mode>]			
Pourcentage d'utilisation CPU.	Flottant	cpu - <i><numéro du CPU></i> ou <i>all</i> (par défaut) type - valeurs possibles : <i>idle, nice, user</i> (par défaut), <i>system</i> (par défaut pour Windows), <i>iowait, interrupt, softirq, steal, guest</i> (sur les kernels Linux 2.6.24 et supérieur), <i>guest_nice</i> (sur les kernels Linux 2.6.33 et supérieur). Les paramètres temps <i>user</i> et <i>nice</i> n'incluent plus le temps <i>guest</i> et le temps <i>guest_nice</i> depuis Zabbix 3.0.14, 3.4.5 et 4.0.0. mode - valeurs possibles : <i>avg1</i> (moyenne sur une minute, par défaut), <i>avg5</i> , <i>avg15</i>	Exemple : ⇒ <code>system.cpu.util[0,user,avg5]</code> Anciens noms : <code>system.cpu.idleX</code> , <code>system.cpu.niceX</code> , <code>system.cpu.systemX</code> , <code>system.cpu.userX</code>
system.hostname[<type>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Nom d'hôte du système.	Chaîne	type (Windows uniquement, ne doit pas être utilisé sur les autres systèmes) - valeurs possibles : <i>netbios</i> (par défaut) ou <i>host</i>	<p>La valeur est récupérée par les fonctions GetComputerName() (pour netbios) ou gethostname() (pour host) sur Windows et par la commande "hostname" sur les autres systèmes.</p> <p>Exemples de valeurs de retour :</p> <p><i>sur Linux:</i> ⇒ system.hostname → linux-w7x1 ⇒ system.hostname → www.zabbix.com</p> <p><i>sur Windows :</i> ⇒ system.hostname → WIN-SERV2008-I6 ⇒ system.hostname[host] → Win-Serv2008-I6LonG</p> <p>Le paramètre type pour cet élément est supporté depuis Zabbix 1.8.6.</p> <p>Voir aussi une description plus détaillée.</p>
system.hw.chassis[<info>]			
Informations sur le châssis.	Chaîne	info - full (par défaut), model, serial, type ou vendor	<p>Exemple : system.hw.chassis[full] Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXX Desktop]</p> <p>Cette clé dépend de la disponibilité de la table SMBIOS. Essayera de lire la table DMI de sysfs, si l'accès sysfs échoue alors il essayera de lire directement en mémoire.</p> <p>Les autorisations Root sont nécessaires car la valeur est récupérée en lisant sysfs ou en mémoire.</p> <p>Supporté depuis l'agent Zabbix en version 2.0.</p>
system.hw.cpu[<cpu>,<info>]			
Information sur le processeur.	Chaîne ou entier	cpu - <numéro du CPU> ou <i>all</i> (par défaut) info - valeurs possibles : <i>full</i> (par défaut), <i>curfreq</i> , <i>maxfreq</i> , <i>model</i> ou <i>vendor</i>	<p>Exemple : ⇒ system.hw.cpu[0,vendor] → AuthenticAMD</p> <p>Récupère les informations dans /proc/cpuinfo et /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq.</p> <p>Si le numéro du CPU et <i>curfreq</i> ou <i>maxfreq</i> est spécifié, une valeur numérique est retournée (Hz).</p> <p>Supporté depuis l'agent Zabbix en version 2.0.</p>
system.hw.devices[<type>]			
Liste des périphériques PCI ou USB.	Texte	type - <i>pci</i> (par défaut) ou <i>usb</i>	<p>Exemple : ⇒ system.hw.devices[pci] → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge [..]</p> <p>Renvoie la sortie de l'utilitaire lspci ou lsusb (exécuté sans aucun paramètre) Pris en charge depuis la version 2.0 de l'agent Zabbix.</p>
system.hw.macaddr[<interface>,<format>]			
Liste des adresses MAC.	Chaîne	interface - <i>all</i> (par défaut) ou une expression régulière ⁴ format - <i>full</i> (par défaut) ou <i>short</i>	<p>Liste les adresses MAC des interfaces dont le nom correspond à l'expression régulière⁴ de l'interface donnée (<i>all</i> liste toutes les interfaces).</p> <p>Exemple : ⇒ system.hw.macaddr[eth0\$,full] → [eth0] 00:11:22:33:44:55</p> <p>Si format est spécifié comme <i>short</i>, les noms d'interfaces et les adresses MAC identiques ne sont pas listées.</p> <p>Supporté depuis l'agent Zabbix en version 2.0.</p>
system.localtime[<type>]			
Heure du système.	Entier - avec type à <i>utc</i> Chaîne - avec type à <i>local</i>	type - valeurs possibles : <i>utc</i> - (par défaut) temps écoulé depuis le 1er janvier 1970 00:00:00 UTC, mesuré en secondes. <i>local</i> - l'heure dans le format 'yyyy-mm-dd,hh:mm:ss.nnn, +hh:mm'	<p>Doit être utilisé pour les vérifications passives uniquement.</p> <p>Les paramètres pour cet élément sont supportés depuis l'agent Zabbix en version 2.0.</p> <p>Exemple : ⇒ system.localtime[local] → crée un élément à l'aide de cette clé, puis utilisez-le pour afficher l'heure de l'hôte dans l'élément d'écran Horloge.</p>
system.run[command,<mode>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Exécute la commande spécifiée sur l'hôte.	Résultat de la commande sous forme de texte 1 - avec mode à <i>nowait</i> (peu importe le résultat de la commande)	command - commande à exécuter mode - valeurs possibles : <i>wait</i> - attend la fin de l'exécution (par défaut), <i>nowait</i> - ne pas attendre	Il est possible de renvoyer jusqu'à 512 Ko de données, y compris les espaces de fin tronqués. Pour être traitée correctement, la sortie de la commande doit être du texte. Exemple : ⇒ <code>system.run[ls -l /]</code> → liste détaillée des fichiers du répertoire racine. <i>Remarque</i> : Pour activer cette fonctionnalité, le fichier de configuration de l'agent doit contenir l'option <code>EnableRemoteCommands=1</code> . La valeur de retour de l'élément est la sortie standard avec l'erreur standard produite par la commande. Le code de sortie n'est pas vérifié. Le résultat vide est autorisé à partir de Zabbix 2.4.0. Voir aussi : Exécution de commandes .
system.stat[resource,<type>]			
Statistiques du système.	Entier ou flottant	ent - nombre d'unités de processeur que cette partition est autorisée à recevoir (flottant) kthr,<type> - informations sur l'état des threads du noyau : <i>r</i> - nombre moyen de threads exécutables du noyau (flottant) <i>b</i> - nombre moyen de threads du noyau placés en file d'attente du gestionnaire de mémoire virtuelle (flottant) memory,<type> - informations concernant l'utilisation de la mémoire virtuelle et de la mémoire réelle : <i>avm</i> - pages virtuelles actives (entier) <i>fre</i> - taille de la liste libre (entier) page,<type> - informations concernant les erreurs de page et l'activité de pagination : <i>fi</i> - page-ins fichier par seconde (flottant) <i>fo</i> - page-outs fichier par seconde (float) <i>pi</i> - pages entrantes paginées dans l'espace de pagination (flottant) <i>po</i> - pages sortantes paginées de l'espace de pagination (flottant) <i>fr</i> - pages libérées (remplacement de la page) (flottant) <i>sr</i> - pages scannées par l'algorithme de remplacement de page (flottant) faults,<type> - taux de traps et d'interruptions : <i>in</i> - interruption de périphériques (flottant) <i>sy</i> - appels système (flottant) <i>cs</i> - commutateurs de contexte de threads du noyau (flottant) cpu,<type> - répartition du pourcentage d'utilisation du temps processeur : <i>us</i> - temps utilisateur (flottant) <i>sy</i> - temps système (flottant) <i>id</i> - temps d'inactivité (flottant) <i>wa</i> - temps d'inactivité durant lequel le système avait des requêtes d' E/S disque/NFS en attente (flottant) <i>pc</i> - nombre de processeurs physiques consommés (flottant) <i>ec</i> - le pourcentage de capacité admissible consommée (flottant) <i>lbusy</i> - indique le pourcentage d'utilisation des processeurs logiques survenu lors de l'exécution au niveau de l'utilisateur et du système (flottant) <i>app</i> - indique les processeurs physiques disponibles dans le pool partagé (flottant) disk,<type> - statistiques du disque : <i>bps</i> - indique la quantité de données transférées (lues ou écrites) sur le lecteur en octets par seconde (entier) <i>tps</i> - indique le nombre de transferts par seconde qui ont été émis sur le disque physique/la bande (flottant)	
...	...	Commentaires Cet élément est supporté seulement sur AIX, depuis Zabbix 1.8.1. Les éléments suivants sont supportés seulement sur AIX LPAR de type "Shared" : ⇒ <code>system.stat[cpu,app]</code> ⇒ <code>system.stat[cpu,ec]</code> (également sur "Dedicated" depuis Zabbix 4.0.1 ; retourne toujours 100 (pourcentage)) ⇒ <code>system.stat[cpu,lbusy]</code> ⇒ <code>system.stat[cpu,pc]</code> (également sur "Dedicated" depuis Zabbix 4.0.1) ⇒ <code>system.stat[ent]</code> (également sur "Dedicated" depuis Zabbix 4.0.1)	
system.sw.arch			
Informations sur l'architecture logicielle.	Chaîne		Exemple : ⇒ <code>system.sw.arch</code> → i686 L'information est récupérée via la fonction <code>uname()</code> . Supporté depuis l'agent Zabbix en version 2.0.
system.sw.os[<info>]			
Informations sur le système d'exploitation.	Chaîne	info - valeurs possibles : <i>full</i> (par défaut), <i>short</i> ou <i>name</i>	Exemple : ⇒ <code>system.sw.os[short]</code> → Ubuntu 2.6.35-28.50-generic 2.6.35.11 L'information est récupérée via (notez que tous les fichiers et les options sont présents sur toutes les distributions) : <code>/proc/version</code> (<i>full</i>) <code>/proc/version_signature</code> (<i>short</i>) le paramètre <code>Pretty_Name</code> de <code>/etc/os-release</code> sur le système le supportant, ou <code>/etc/issue.net</code> (<i>name</i>) Supporté depuis l'agent Zabbix en version 2.0.
system.sw.packages[<package>,<manager>,<format>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Liste les packages installés.	Texte	<p>package - <i>all</i> (par défaut) ou une expression régulière⁴</p> <p>manager - <i>all</i> (par défaut) ou un gestionnaire de package</p> <p>format - <i>full</i> (par défaut) ou <i>short</i></p>	<p>Liste (par ordre alphabétique) les paquets installés dont le nom correspond à l'expression régulière⁴ donnée (<i>all</i> : les liste tous).</p> <p>Exemple : ⇒ <code>system.sw.packages[mini,dpkg,short]</code> → python-minimal, python2.6-minimal, ubuntu-minimal</p> <p>Supporte les gestionnaires de packages (exécution de commandes): dpkg (<code>dpkg --get-selections</code>) pkgtool (<code>ls /var/log/packages</code>) rpm (<code>rpm -qa</code>) pacman (<code>pacman -Q</code>)</p> <p>Si <code>format</code> est spécifié à <i>full</i>, les packages sont regroupés par le gestionnaire de package (chaque gestionnaire sur un ligne séparée commençant par son nom entre crochets). Si <code>format</code> est spécifié à <i>short</i>, les packages ne sont pas regroupés et sont listés sur une seule ligne.</p> <p>Supporté depuis l'agent Zabbix en version 2.0.</p>
system.swap.in[<device>,<type>]			
Statistiques du swap entrant (du périphérique à la mémoire).	Entier	<p>device - périphérique utilisé pour le swap (<i>all</i> par défaut)</p> <p>type - valeurs possibles : <i>count</i> (nombre de swapins), <i>sectors</i> (secteurs swappés), <i>pages</i> (pages swappées). Voir le support par plateforme pour plus de détails sur les valeurs pas défaut.</p>	<p>Exemple : ⇒ <code>system.swap.in[,pages]</code></p> <p>La source de cette information est : <code>/proc/swaps</code>, <code>/proc/partitions</code>, <code>/proc/stat</code> (Linux 2.4) <code>/proc/swaps</code>, <code>/proc/diskstats</code>, <code>/proc/vmstat</code> (Linux 2.6)</p>
system.swap.out[<device>,<type>]			
Statistiques du swap sortant (de la mémoire vers le périphérique).	Entier	<p>device - périphérique utilisé pour le swap (<i>all</i> par défaut)</p> <p>type - valeurs possibles : <i>count</i> (nombre de swapouts), <i>sectors</i> (secteurs swappés), <i>pages</i> (pages swappées). Voir le support par plateforme pour plus de détails sur les valeurs par défaut.</p>	<p>Exemple : ⇒ <code>system.swap.out[,pages]</code></p> <p>La source de cette information est : <code>/proc/swaps</code>, <code>/proc/partitions</code>, <code>/proc/stat</code> (Linux 2.4) <code>/proc/swaps</code>, <code>/proc/diskstats</code>, <code>/proc/vmstat</code> (Linux 2.6)</p>
system.swap.size[<device>,<type>]			
Taille de l'espace de swap en octets ou en pourcentage du total	Entier - pour les octets Flottant - pour le pourcentage	<p>device - périphérique utilisé pour le swap (par défaut est <i>all</i>)</p> <p>type - valeurs possibles : <i>free</i> (espace libre du swap, par défaut), <i>free</i> (espace libre du swap, en pourcentage), <i>used</i> (espace utilisé du swap, en pourcentage), <i>total</i> (espace totale du swap), <i>used</i> (espace utilisé du swap)</p>	<p>Exemple : ⇒ <code>system.swap.size[,pfree]</code> → pourcentage d'espace libre du swap</p> <p>Si <code>device</code> n'est pas spécifié l'agent Zabbix ne prendra pas en compte le périphérique d'échange (fichiers), la mémoire physique sera ignorée. Par exemple, sur les systèmes Solaris la commande <code>swap -s</code> inclut une partie de la mémoire physique et de périphérique d'échange (contrairement à <code>swap -l</code>).</p> <p>Notez que cette clé peut renvoyer un pourcentage erroné sur les plateforme Windows virtualisées (VMware ESXi, VirtualBox). Dans ce cas, il est préférable d'utiliser la clé <code>perf_counter{\700_Total}\702</code> pour obtenir les données d'utilisation correctes du swap.</p> <p>Ancien nom : <code>system.swap.free</code>, <code>system.swap.total</code></p>
system.uname			
Identification du system.	Chaîne		<p>Exemple de valeur de retour (Unix) : FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386</p> <p>Exemple de valeur de retour (Windows): Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86</p> <p>Sur Unix, depuis Zabbix 2.2.0, la valeur de cet élément est obtenue avec l'appel système <code>uname()</code>. Précédemment, elle était obtenue en invoquant "uname -a". ". La valeur de cet élément peut différer de la sortie de "uname -a" et n'inclut pas les informations supplémentaires que "uname -a" affiche en fonction d'autres sources.</p> <p>Sur Windows, depuis Zabbix 3.0, la valeur pour cet élément est obtenue par les classes WMI <code>Win32_OperatingSystem</code> et <code>Win32_Processor</code>. Précédemment, elle était obtenue depuis les APIs Windows volatiles et des clés de registre non documentées. Le nom de l'OS (édition incluse) peut être traduit dans la langue d'affichage de l'utilisateur. Sur certaines versions de Windows, il contient des symboles de marque et des espaces supplémentaires.</p> <p>Notez que sous Windows, l'élément renvoie l'architecture du système d'exploitation, tandis que sous Unix, il renvoie l'architecture du processeur.</p>
system.uptime			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Temps de fonctionnement du système en secondes.	Entier		Dans la configuration de l'élément , utilisez les unités s ou uptime pour avoir des valeurs lisibles.
system.users.num			
Nombre d'utilisateurs connectés.	Entier		La commande who est utilisée côté agent pour obtenir la valeur.
vfs.dev.read[<device>,<type>,<mode>]			
Statistiques de lecture de disque.	Entier - avec type à <i>sectors, operations, bytes</i> Flottant - avec type à <i>sps, ops, bps</i> <i>Note</i> : Depuis 4.0.4, si un intervalle de mise à jour de 3h ou plus est utilisé ² , retournera toujours '0'	device - périphérique disque (<i>all</i> par défaut) type - valeurs possibles : <i>sectors, operations, bytes, sps, ops, bps</i> Ce paramètre doit être spécifié, car les valeurs par défaut diffèrent selon les systèmes d'exploitation. <i>sps, ops, bps</i> représentent respectivement : secteurs, opérations, octets par secondes. mode - valeurs possibles : <i>avg1</i> (moyenne par minutes, par défaut), <i>avg5, avg15</i> . Ce paramètre est supporté uniquement avec le type à : <i>sps, ops, bps</i> .	Vous pouvez utiliser des noms de périphériques relatifs (par exemple, <i>sda</i>) ainsi que le préfixe optionnel /dev/ (par exemple, /dev/ <i>sda</i>). Les volumes logiques LVM sont supportés. Les valeurs pas défaut du paramètre 'type' pour différents OS : AIX - <i>operations</i> FreeBSD - <i>bps</i> Linux - <i>sps</i> OpenBSD - <i>operations</i> Solaris - <i>bytes</i> Exemple : ⇒ <code>vfs.dev.read[,operations]</code> <i>sps, ops</i> et <i>bps</i> sur les plates-formes prises en charge sont limités à 8 périphériques (7 individuels et un pour <i>all</i>). Depuis Zabbix 2.0.1 cette limite est de 1024 périphériques (1023 individuels et un pour <i>all</i>). Ancien nom : <i>io[*]</i>
vfs.dev.write[<device>,<type>,<mode>]			
Statistiques d'écriture du disque.	Entier - avec type à <i>sectors, operations, bytes</i> Flottant - avec type à <i>sps, ops, bps</i> <i>Note</i> : Depuis 4.0.4, si un intervalle de 3h ou plus est utilisé ² , retournera toujours '0'	device - périphérique disque (<i>all</i> ³ par défaut) type - valeurs possibles : <i>sectors, operations, bytes, sps, ops, bps</i> Ce paramètre doit être spécifié, car les valeurs par défaut diffèrent selon les systèmes d'exploitation. <i>sps, ops, bps</i> représentent respectivement : secteurs, opérations, octets par secondes. mode - valeurs possibles : <i>avg1</i> (moyenne par minutes, par défaut), <i>avg5, avg15</i> . Ce paramètre est supporté uniquement avec le type à : <i>sps, ops, bps</i> .	Vous pouvez utiliser des nom de périphériques relatifs (par exemple, <i>sda</i>) ainsi que le préfixe optionnel /dev/ prefix (par exemple, /dev/ <i>sda</i>). Les volumes logiques sont supportés. Les valeurs pas défaut du paramètre 'type' pour différents OS : AIX - <i>operations</i> FreeBSD - <i>bps</i> Linux - <i>sps</i> OpenBSD - <i>operations</i> Solaris - <i>bytes</i> Exemple : ⇒ <code>vfs.dev.write[,operations]</code> <i>sps, ops</i> et <i>bps</i> sur les plates-formes prises en charge sont limités à 8 périphériques (7 individuels et un pour <i>all</i>). Depuis Zabbix 2.0.1 cette limite est de 1024 périphériques (1023 individuels et un pour <i>all</i>). Ancien nom : <i>io[*]</i>
vfs.dir.count[<dir>,<regex_incl>,<regex_excl>,<types_incl>,<types_excl>,<max_depth>,<min_size>,<max_size>,<min_age>,<max_age>]			

		Clé	
Description	Valeur de retour	Paramètres	Commentaires
Compte le nombre d'entrées d'un dossier.	Entier	<p>dir - chemin absolu du dossier</p> <p>regex_incl - expression régulière⁴ décrivant le modèle de nom du fichier, du répertoire et du lien symbolique à inclure (inclus tous les fichiers, dossiers et liens symboliques si vide; la valeur par défaut est une chaîne vide)</p> <p>regex_excl - expression régulière⁴ décrivant le modèle de nom du fichier, du répertoire et du lien symbolique à exclure (ne rien exclure si vide ; la valeur par défaut est une chaîne vide)</p> <p>types_incl - un ensemble de types d'entrées de dossier à compter, valeurs possibles : <i>file</i> - fichier, <i>dir</i> - sous-dossier, <i>sym</i> - lien symbolique, <i>sock</i> - socket, <i>bdev</i> - périphérique block, <i>cdev</i> - character device, <i>fifo</i>- FIFO, <i>dev</i>- synonyme de "bdev,cdev", <i>all</i>- tous les types mentionnés précédemment, c'est à dire "file,dir,sym,sock,bdev,cdev,fifo". Il s'agit de la valeur par défaut, si le paramètre est laissé vide. Les types multiples doivent être séparés par des virgules et l'ensemble doit être entouré par des doubles guillemets "".</p> <p>types_excl - un ensemble de types d'entrées de dossier à ne pas compter, les mêmes valeurs et syntaxes que pour <types_incl> s'appliquent. Si certaines entrées se trouvent dans <types_incl> et dans <types_excl>, les entrées de ce type ne seront PAS comptées.</p> <p>max_depth - profondeur maximum de sous-dossier à traverser. -1 (par défaut) - illimité, 0 - ne descend pas dans les sous-dossiers.</p> <p>min_size - taille minimum des fichiers à compter. Les fichiers plus petits que la taille spécifiée ne seront pas comptés. La valeur est en octets. Les suffixes mémoire peuvent être utilisés.</p> <p>max_size - taille maximum des fichiers à compter. Les fichiers plus gros que la taille spécifiée ne seront pas comptés. La valeur est en octets. Les suffixes mémoire peuvent être utilisés.</p> <p>min_age - âge minimum d'entrée du dossier à compter. Les entrées modifiées avant cette valeur ne seront pas comptées. Il s'agit d'un entier représentant des secondes. Les suffixes de temps peuvent être utilisés.</p> <p>max_age - âge maximum d'entrée du dossier à compter. Les entrées aussi vieilles ou plus vieilles que la valeur spécifiée ne seront pas comptées (basé sur l'heure de modification). Il s'agit d'un entier représentant des secondes. Les suffixes de temps peuvent être utilisés.</p>	<p>Les variables d'environnement comme %APP_HOME%, \$HOME et %TEMP% ne sont pas supportées.</p> <p>Les pseudo-répertoires "." et ".." ne sont jamais comptabilisés.</p> <p>Les liens symboliques ne sont jamais suivis pour la traversée des répertoires.</p> <p>Pour les répertoires Windows, les liens symboliques sont ignorés et les liens physiques ne sont comptabilisés qu'une seule fois.</p> <p>regex_incl et regex_excl sont des expressions régulières compatibles Perl (PCRE). regex_incl et regex_excl sont appliqués au fichiers et aux répertoires lors du calcul de la taille d'une entrée, mais sont ignorés lors de la traversée d'un dossier (si regex_incl est "(?)^\.+\.zip\$" et max_depth n'est pas positionné, alors tous les sous-répertoires seront traversés, mais seulement les fichiers de type zip seront comptés). Si un nom de fichier correspond à regex_incl et à regex_excl ce fichier ne sera pas compté.</p> <p>Le temps d'exécution sera limité par une valeur de délai d'attente par défaut de 3 secondes (paramètre «Timeout» dans le fichier de configuration de l'agent). Étant donné que la traversée de répertoires volumineux peut prendre plus de temps que cela, aucune donnée ne sera renvoyée et l'élément sera marqué comme "non pris en charge". Le décompte partiel ne sera pas retourné.</p> <p>Lors du filtrage par taille, seuls les fichiers réguliers ont des tailles significatives. Sous Linux et BSD, les répertoires ont également une taille non nulle (quelques Ko généralement). Les périphériques ont une taille nulle, par exemple la taille de /dev/sda1 ne reflète pas la taille de la partition associée. Par conséquent, lors de l'utilisation des paramètres <min_size> et <max_size>, il est conseillé de spécifier <types_incl> à "file", pour éviter les surprises.</p> <p>Exemples : ⇒ vfs.dir.count[/dev] - surveille le nombre de périphériques dans /dev (Linux) ⇒ vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] - surveille le nombre de fichier dans le répertoire temporaire (Windows)</p> <p>Supporté depuis Zabbix 4.0.0.</p>
vfs.dir.size[<i>dir</i>,<regex_incl>,<regex_excl>,<mode>,<max_depth>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Taille d'un répertoire (en octets).	Entier	<p>dir - chemin absolu du répertoire</p> <p>regex_incl - expression régulière⁴ décrivant le modèle de nom de fichier, répertoire et lien symbolique à inclure (inclure tous les fichiers, répertoires et liens symboliques si vide, une chaîne vide est la valeur par défaut)</p> <p>regex_excl - expression régulière⁴ décrivant le fichier, le répertoire et le lien symbolique à exclure (ne pas exclure si vide, une chaîne vide est la valeur par défaut)</p> <p>mode - valeurs possibles : <i>apparent</i> (par défaut) - obtient des tailles de fichiers apparentes plutôt que l'utilisation du disque (agit comme du <code>-sb dir</code>), <i>disk</i> - obtient l'utilisation du disque (agit comme du <code>-s -B1 dir</code>). Contrairement à la commande <code>du</code>, l'élément <code>vfs.dir.size</code> prend en compte les fichiers cachés lors du calcul de la taille du répertoire (agit comme du <code>-sb .[^.]* *</code> dans le répertoire).</p> <p>max_depth - profondeur maximale des sous-répertoires à parcourir. -1 (par défaut) - illimité, 0 - pas de descente dans les sous-répertoires.</p>	<p>Seuls les répertoires avec au moins l'autorisation de lecture pour l'utilisateur <i>zabbix</i> sont calculés.</p> <p>Sur Windows, les liens symboliques et les liens durs ne sont pris en compte qu'une seule fois.</p> <p>Avec des répertoires volumineux ou des lecteurs lents, cet élément peut expirer en raison du paramètre Timeout dans les fichiers de configuration de l'<i>agent</i> et du <i>serveur/proxy</i>. Augmentez les valeurs de délai d'attente si nécessaire.</p> <p>Exemple : ⇒ <code>vfs.dir.size[/tmp,log]</code> - calcule la taille de tous les fichiers dans /tmp qui contiennent 'log' ⇒ <code>vfs.dir.size[/tmp,log,^.\.old\$]</code> - calcule la taille de tous les fichiers dans /tmp qui contiennent 'log', en excluant ceux contenant '.old'</p> <p>La taille limite des fichiers dépend de la prise en charge des fichiers volumineux.</p> <p>Supporté depuis Zabbix 3.4.0.</p>
vfs.file.cksum[file]			
Checksum du fichier, calculé par l'algorithme UNIX cksum.	Entier	file - chemin complet d'accès au fichier	<p>Exemple : ⇒ <code>vfs.file.cksum[/etc/passwd]</code></p> <p>Exemple de valeur de retour : 1938292000</p> <p>Ancien nom : <i>cksum</i></p> <p>La taille du fichier dépend de la prise en charge des fichiers volumineux.</p>
vfs.file.contents[file,<encoding>]			
Récupère le contenu d'un fichier.	Texte	file - chemin complet d'accès au fichier encoding - encodage	<p>Retourne une chaîne vide si le fichier est vide ou contient seulement les caractères LF/CR.</p> <p>Exemple : ⇒ <code>vfs.file.contents[/etc/passwd]</code></p> <p>Cet élément est limité au fichier dont la taille est inférieure à 64 Ko.</p> <p>Supporté depuis l'agent Zabbix en version 2.0.</p>
vfs.file.exists[file]			
Vérifie si le fichier existe.	0 - non trouvé 1 - le fichier ou le lien (symbolique ou dur) vers le fichier existe	file - chemin complet d'accès au fichier	<p>Exemple : ⇒ <code>vfs.file.exists[/tmp/application.pid]</code></p> <p>La valeur de retour dépend de ce que retourne la macro <code>S_ISREG</code> POSIX.</p> <p>La limite de la taille du fichier dépend de la prise en charge des fichiers volumineux.</p>
vfs.file.md5sum[file]			
checksum MD5 du fichier.	Chaîne de caractères (table de hachage MD5 du fichier)	file - chemin complet d'accès au fichier	<p>Exemple : ⇒ <code>vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]</code></p> <p>Exemple de valeur de retour : b5052decb577e0fffd622d6ddc017e82</p> <p>La taille limite (64 Mo) pour cet élément a été supprimée en version 1.8.6.</p> <p>La taille limite du fichier dépend de la prise en charge des fichiers volumineux.</p>
vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Trouve une chaîne dans un fichier.	La ligne contenant la chaîne correspondante, ou comme spécifiée par le paramètre optionnel output	file - chemin d'accès complet au fichier regexp - Expression régulière ⁴ décrivant le modèle requis encoding - encodage start line - Le numéro de la première ligne à chercher (première ligne du fichier par défaut). end line - Le numéro de la dernière ligne à chercher (dernière ligne du fichier par défaut). output - un modèle de mise en forme de sortie en option. La séquence d'échappement \O est remplacée par la partie du texte correspondant (à partir du premier caractère où la correspondance commence jusqu'au dernier caractère où la correspondance se termine) alors qu'une séquence d'échappement \N (où N=1...9) est remplacée par le Nième groupe correspondant (ou une chaîne vide si N dépasse le nombre de groupes capturés).	Seule la première ligne correspondante est retournée. Une chaîne vide est renvoyée si aucune ligne ne correspond à l'expression. L'extraction de contenu à l'aide du paramètre output a lieu sur l'agent. Les paramètres <code>start line</code> , <code>end line</code> et <code>output</code> sont supporté depuis la version 2.2. Exemples : ⇒ <code>vfs.file.regexp[/etc/passwd,zabbix]</code> ⇒ <code>vfs.file.regexp[/path/to/some/file,"([0-9]+)\$",,3,5,1]</code> ⇒ <code>vfs.file.regexp[/etc/passwd,^zabbix:([0-9]+),,,,1]</code> → récupère l'ID de l'utilisateur <i>zabbix</i>
vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]			
Trouve une chaîne dans un fichier.	0 - aucune correspondance trouvée 1 - correspondance trouvée	file - chemin d'accès complet au fichier regexp - Expression régulière ⁴ décrivant le modèle encoding - encodage start line - numéro de la première ligne à chercher (première ligne du fichier par défaut). end line - numéro de la dernière ligne à chercher (dernière ligne du fichier par défaut).	Les paramètres <code>start line</code> et <code>end line</code> sont supporté depuis la version 2.2. Exemple : ⇒ <code>vfs.file.regmatch[/var/log/app.log,error]</code>
vfs.file.size[file]			
Taille du fichier (en octets).	Entier	file - chemin d'accès complet au fichier	Le fichier doit avoir les permission en lecture pour l'utilisateur <i>zabbix</i> . Exemple : ⇒ <code>vfs.file.size[/var/log/syslog]</code> La taille limite du fichier dépend de la prise en charge des fichiers volumineux .
vfs.file.time[file,<mode>]			
Informations temporelles du fichier.	Entier (horodatage Unix)	file - chemin d'accès complet au fichier mode - valeurs possibles : <i>modify</i> (par défaut) - date de modification, <i>access</i> - date du dernier accès, <i>change</i> - date du dernier changement	Exemple : ⇒ <code>vfs.file.time[/etc/passwd,modify]</code> La taille limite du fichier dépend de la prise en charge des fichiers volumineux .
vfs.fs.discovery			
Liste des systèmes de fichiers montés. Utilisé pour la découverte de bas niveau.	Objet JSON		Supporté depuis l'agent Zabbix en version 2.0. La macro <code>{#FSDRIVETYPE}</code> est supportée sur Windows depuis l'agent Zabbix en version 3.0.
vfs.fs.inode[fs,<mode>]			
Nombre ou pourcentage d'inodes.	Entier - pour les nombres Flottant - pour les pourcentages	fs - système de fichier mode - valeurs possibles : <i>total</i> (par défaut), <i>free</i> , <i>used</i> , <i>pfree</i> (libre, pourcentage), <i>pusd</i> (utilisé, pourcentage)	Exemple : ⇒ <code>vfs.fs.inode[/,pfree]</code> Ancien nom : <code>vfs.fs.inode.free[*]</code> , <code>vfs.fs.inode.pfree[*]</code> , <code>vfs.fs.inode.total[*]</code>
vfs.fs.size[fs,<mode>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Espace disque en octets ou en pourcentage du total.	Entier - pour les octets Flottant - pour les pourcentages	fs - Système de fichiers mode - valeurs possibles : <i>total</i> (par défaut), <i>free</i> , <i>used</i> , <i>pfree</i> (<i>free</i> , pourcentage), <i>used</i> (<i>used</i> , pourcentage)	Dans le cas d'un volume monté, l'espace disque du système de fichier local est retourné. Exemple : ⇒ <code>vfs.fs.size[/tmp,free]</code> L'espace réservé d'un système de fichiers est pris en compte et n'est pas inclus lors de l'utilisation du mode <i>free</i> . Ancien nom : <code>vfs.fs.free[*]</code> , <code>vfs.fs.total[*]</code> , <code>vfs.fs.used[*]</code> , <code>vfs.fs.pfree[*]</code> , <code>vfs.fs.pused[*]</code>
vm.memory.size[<mode>]			
Taille de la mémoire en octets ou en pourcentage du total.	Entier - pour les octets Flottant - pour les pourcentages	mode - valeurs possibles : <i>total</i> (par défaut), <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> , <i>used</i> , <i>pused</i> (<i>used</i> , pourcentage), <i>available</i> , <i>pavailable</i> (<i>available</i> , pourcentage)	Cet élément accepte 3 catégories de paramètres : 1) <i>total</i> - quantité totale de mémoire ; 2) types de mémoire spécifiques à la plate-forme : <i>active</i> , <i>anon</i> , <i>buffers</i> , <i>cached</i> , <i>exec</i> , <i>file</i> , <i>free</i> , <i>inactive</i> , <i>pinned</i> , <i>shared</i> , <i>slab</i> , <i>wired</i> ; 3) les estimations au niveau utilisateur sur la quantité de mémoire utilisée et disponible : <i>used</i> , <i>pused</i> , <i>available</i> , <i>pavailable</i> . Voir une description plus détaillée des parameters de <code>vm.memory.size</code> . Ancien nom : <code>vm.memory.buffers</code> , <code>vm.memory.cached</code> , <code>vm.memory.free</code> , <code>vm.memory.shared</code> , <code>vm.memory.total</code>
web.page.get[host,<path>,<port>]			
Récupère le contenu d'une page web.	Source de la page Web en tant que texte (y compris les en-têtes)	host - nom de l'hôte path - chemin du document HTML (par défaut : /) port - numéro de port (par défaut 80)	Retourne une chaîne vide en cas d'échec. Exemple : ⇒ <code>web.page.get[www.zabbix.com,index.php,80]</code>
web.page.perf[host,<path>,<port>]			
Temps de chargement de la page Web complète (en secondes).	Flottant	host - nom de l'hôte path - chemin du document HTML (par défaut : /) port - numéro de port (par défaut 80)	Retourne 0 en cas d'échec. Exemple : ⇒ <code>web.page.perf[www.zabbix.com,index.php,80]</code>
web.page.regex[host,<path>,<port>,<regexp>,<length>,<output>]			
Trouve une chaîne sur une page Web.	La chaîne correspondante, ou comme spécifiée par le paramètre <code>output</code> optionnel	host - nom de l'hôte path - chemin du document HTML (par défaut : /) port - numéro de port (par défaut 80) regexp - Expression régulière ⁴ décrivant le modèle length - nombre maximum de caractères à retourner output - un modèle de mise en forme de sortie facultatif. La séquence d'échappement <code>\0</code> est remplacée par la partie du texte correspondant (à partir du premier caractère où la correspondance commence jusqu'au dernier caractère où la correspondance se termine) alors qu'une séquence d'échappement <code>\N</code> (où N=1...9) est remplacée par le Nième groupe correspondant (ou une chaîne vide si N dépasse le nombre de groupes capturés).	Retourne une chaîne vide si aucune correspondance n'a été trouvée ou en cas d'échec. L'extraction de contenu à l'aide du paramètre <code>output</code> se fait au niveau de l'agent. Le paramètre <code>output</code> est supporté depuis la version 2.2. Exemple : ⇒ <code>web.page.regex[www.zabbix.com,index.php,80,OK,2]</code>
zabbix.stats[<ip>,<port>]			
Renvoie un ensemble de métriques internes du serveur Zabbix ou du proxy à distance.	Objet JSON	ip - Liste d'IP/DNS/masques de réseaux des serveurs / proxies à interroger à distance (par défaut : 127.0.0.1) port - port du serveur / proxy à interroger à distance (par défaut : 10051)	Notez que la demande de statistiques ne sera acceptée qu'à partir des adresses répertoriées dans le paramètre <code>serveur / proxy 'StatsAllowedIP'</code> sur l'instance cible. Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Surveillance à distance des statistiques Zabbix . Cet élément est supporté depuis Zabbix 4.0.5.
zabbix.stats[<ip>,<port>,queue,<from>,<to>]			

Clé			
Description	Valeur de retour	Paramètres	Commentaires
Renvoie le nombre d'éléments surveillés dans la file d'attente qui sont retardés sur le serveur Zabbix ou le proxy à distance.	Objet JSON	<p>ip - Liste d'IP/DNS/masques de réseaux des serveurs / proxys à interroger à distance (par défaut : 127.0.0.1)</p> <p>port - port du serveur / proxy à interroger à distance (par défaut : 10051)</p> <p>queue - constante (à utiliser telle quelle)</p> <p>from - retardé d'au moins (par défaut : 6 secondes)</p> <p>to - retardé d'au plus (par défaut : infini)</p>	<p>Notez que la demande de statistiques ne sera acceptée qu'à partir des adresses répertoriées dans le paramètre <code>serveur / proxy 'StatsAllowedIP'</code> sur l'instance cible.</p> <p>Cet élément est supporté depuis Zabbix 4.0.5.</p>

Notes de bas de page

¹ Remarque spécifique à Linux : L'agent Zabbix doit avoir un accès en lecture seule au système de fichiers `/proc`. Les correctifs du noyau de www.grsecurity.org limitent les droits d'accès des utilisateurs non privilégiés.

² `vfs.dev.read[]`, `vfs.dev.write[]` : à partir de Zabbix 4.0.4, l'agent Zabbix mettra fin à "stale" connexions de périphérique si les valeurs des éléments ne sont pas consultées pendant plus de 3 heures. Cela peut se produire si un système comporte des périphériques dont les chemins changent de manière dynamique ou si un périphérique est supprimé manuellement. Notez également que ces éléments, si vous utilisez un intervalle de mise à jour de 3 heures ou plus, renverront toujours "0".

³ `vfs.dev.read[]`, `vfs.dev.write[]` : Si par défaut `all` est utilisé pour le premier paramètre, alors la clé renverra des statistiques récapitulatives, y compris tous les périphériques de blocs tels que `sda`, `sbd` et leurs partitions (`sda1`, `sda2`, `sdb3...`) et plusieurs périphériques (MD RAID) basés sur ces périphériques/partitions de blocs et volumes logiques (LVM) en fonction de ces périphériques/partitions de blocs. Dans ce cas, les valeurs renvoyées doivent être considérées uniquement comme des valeurs relatives (dynamiques dans le temps), mais pas comme des valeurs absolues.

⁴ [Expression Régulière Compatible Perl](#) (PCRE) depuis Zabbix 3.4 ; Expression régulière POSIX étendue avant cela. Voir aussi : [Les expressions régulières supportées par emplacement](#).

Encodages disponibles

Le paramètre `encoding` est utilisé pour spécifier l'encodage pour le traitement des vérifications d'éléments correspondantes, afin que les données acquises ne soient pas corrompues. Pour obtenir une liste des codages pris en charge (identificateurs de page de codes), consultez la documentation correspondante, telle que la documentation relative à [libiconv](#) (GNU Project) ou Microsoft Windows SDK pour les "Identificateurs de page de code".

Si la valeur `encoding` est vide, alors UTF-8 (paramètres régionaux par défaut pour les nouvelles distributions Unix/Linux, voir les paramètres de votre système) ou ANSI avec une extension spécifique au système (Windows) est utilisé par défaut.

Dépannage des éléments d'agent

1. Si elle est utilisée avec un agent passif, la valeur du *Timeout* dans la configuration du serveur peut être supérieure à *Timeout* dans le fichier de configuration de l'agent. Sinon, l'élément peut ne pas avoir de valeur car la demande du serveur vers l'agent a expiré en premier.

From:
<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:
https://www.zabbix.com/documentation/4.0/fr/manual/config/items/itemtypes/zabbix_agent

Last update: **2019/04/02 17:46**

