

## 1 Fonctions de déclencheur supportées

Toutes les fonctions supportées dans les [expressions de déclencheur](#) sont listées ici.

| FONCTION   |  |  |
|--|--|--|
| Description  | Paramètres   | Commentaires   |
| <b>abschange</b>   |  |  |
| La différence absolue entre la dernière valeur et les valeurs précédentes. |  | Types de valeurs supportés : flottant, int, str, text, log<br><br>Par exemple :<br>(valeur précédente;dernière valeur=abschange)<br>1;5=4<br>3;1=2<br>0;-2.5=2.5<br><br>Pour les chaînes renvoie :<br>0 - les valeurs sont égales<br>1 - les valeurs diffèrent   |
| <b>avg</b> (sec #num,<time_shift>)   |  |  |
| Valeur moyenne d'un élément au cours de la période d'évaluation définie.   | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)<br><b>time_shift</b> (optionnel) - le point d'évaluation est déplacé du nombre de secondes dans le temps | Types de valeurs supportés : flottant, int<br><br>Exemples :<br>⇒ avg(#5) → valeur moyenne pour les 5 dernières valeurs<br>⇒ avg(1h) → valeur moyenne pour une heure<br>⇒ avg(1h,1d) → valeur moyenne pour une heure il y a 1 jour.<br><br>Le paramètre <b>time_shift</b> est supporté depuis Zabbix 1.8.2. C'est utile lorsqu'il est nécessaire de comparer la valeur moyenne actuelle avec la valeur moyenne d'il y a <b>time_shift</b> secondes.  |
| <b>band</b> (<sec #num>,mask,<time_shift>)                                 |  |  |
| Valeur de "ET binaire" d'une valeur d'élément et d'un masque.              | <b>sec</b> (ignoré, égal #1) ou <b>#num</b> (optionnel) - la Nième valeur plus récente<br><b>mask</b> (obligatoire) - entier non signé 64-bit (0 - 18446744073709551615)<br><b>time_shift</b> (optionnel) - voir avg()                     | Types de valeurs supportés : int<br><br>Notez que <b>#num</b> fonctionne différemment ici par rapport à beaucoup d'autres fonction (voir last()).<br><br>Bien que la comparaison se fasse au niveau du bit, toutes les valeurs doivent être fournies et sont retournées en décimal. Par exemple, la vérification du troisième bit est effectuée en comparant à 4 et non à 100.<br><br>Exemples :<br>⇒ band(,12)=8 ou band(,12)=4 → 3ième ou 4ième bit défini, mais pas les deux en même temps<br>⇒ band(,20)=16 → 3ème bit non activé et 5ème bit activé.<br><br>Cette fonction est supportée depuis Zabbix 2.2.0. |
| <b>change</b>  |  |  |

| FONCTION   |            |  |
|--|------------|--|
| Description  | Paramètres | Commentaires   |
| <b>abschange</b>   |            |  |
| La différence entre la dernière valeur et les valeurs précédentes. |            | Types de valeurs supportés : flottant, int, str, text, log<br><br>Par exemple :<br>(valeur précédente;dernière valeur=change)<br>1;5=+4<br>3;1=-2<br>0;-2.5=-2.5<br><br>Voir aussi : <a href="#">abschange</a> pour comparaison<br><br>Pour les chaînes retourne :<br>0 - les valeurs sont égales<br>1 - les valeurs différent |
| <b>count</b> (sec #num,<pattern>,<operator>,<time_shift>)          |            |  |

| FONCTION   |  |  |
|--|--|--|
| Description  | Paramètres   | Commentaires   |
| <b>abschange</b>   |  |  |
| <p>Nombre de valeurs dans la période d'évaluation définie.</p> | <p><b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)</p> <p><b>pattern</b> (optionnel) - motif requis</p> <p><b>operator</b> (optionnel)</p> <p>Operators supportés :</p> <p><i>eq</i> - égal</p> <p><i>ne</i> - non égal</p> <p><i>gt</i> - supérieur que</p> <p><i>ge</i> - supérieur ou égal</p> <p><i>lt</i> - inférieur</p> <p><i>le</i> - inférieur ou égal</p> <p><i>like</i> - correspond si contient un motif (sensible à la casse)</p> <p><i>band</i> - ET binaire</p> <p><i>regex</i> - correspondance sensible à la casse d'une expression régulière donnée dans <i>pattern</i></p> <p><i>iregexp</i> - correspondance (non sensible à la casse) de l'expression régulière donnée dans <i>pattern</i></p> <p>Notez que :</p> <p><i>eq</i> (défaut), <i>ne</i>, <i>gt</i>, <i>ge</i>, <i>lt</i>, <i>le</i>, <i>band</i>, <i>regex</i>, <i>iregexp</i> sont supportés pour les éléments entier</p> <p><i>eq</i> (défaut), <i>ne</i>, <i>gt</i>, <i>ge</i>, <i>lt</i>, <i>le</i>, <i>regex</i>, <i>iregexp</i> sont supportés pour les éléments flottants</p> <p><i>like</i> (défaut), <i>eq</i>, <i>ne</i>, <i>regex</i>, <i>iregexp</i> sont supportés pour les éléments chaînes, texte et log</p> <p><b>time_shift</b> (optionnel) - voir <i>avg()</i></p> | <p>Types de valeurs supportés : flottant, entier, chaîne, texte, log</p> <p>Les éléments flottants correspondent avec une précision de 0.000001.</p> <p>Avec <i>band</i> comme troisième paramètre, le second paramètre <i>pattern</i> peut être spécifié comme deux nombres, séparé par '<i>'</i>' :</p> <p><b>nombre_à_comparer/masque</b>. <i>count()</i> calcule "ET binaire" depuis la valeur et le <i>masque</i> et compare le resultat à <i>nombre_à_comparer</i>. Si le résultat du "ET binaire" est égale au <i>nombre_à_comparer</i>, la valeur est comptabilisée. Si <i>nombre_à_comparer</i> et le <i>masque</i> sont égaux, seul le <i>masque</i> doit être spécifié (sans '<i>'</i>').</p> <p>Avec <i>regex</i> ou <i>iregexp</i> comme troisième paramètre, le second paramètre <i>pattern</i> peut être une expression régulière normale ou <b>globale</b> (commençant par '@'). En cas d'expressions régulières globales, la sensibilité à la casse est héritée des paramètres globaux d'expressions régulières. Dans le but de faire correspondre les expressions régulières, les valeurs flottantes seront toujours représentées avec 4 chiffres après la virgule après le signe '.'. Notez également que pour les grands nombres, la différence de représentation décimale (stockée dans la base de données) et binaire (utilisée par le serveur Zabbix) peut affecter le 4ème chiffre décimal.</p> <p>Exemples :</p> <p>⇒ <i>count(10m)</i> → nombre de valeurs pour les 10 dernières minutes</p> <p>⇒ <i>count(10m,"error",eq)</i> → nombre de valeurs pour les 10 dernières minutes qui sont égales à 'error'</p> <p>⇒ <i>count(10m,12)</i> → nombre de valeurs pour les 10 dernières minutes qui sont égales à '12'</p> <p>⇒ <i>count(10m,12,gt)</i> → nombre de valeurs pour les 10 dernières minutes qui sont supérieures à '12'</p> <p>⇒ <i>count(#10,12,gt)</i> → nombre de valeurs sur les 10 dernières valeurs qui sont supérieures à '12'</p> <p>⇒ <i>count(10m,12,gt,1d)</i> → nombre de valeurs pour les 10 dernières minutes 24h avant maintenant qui sont supérieures à '12'</p> <p>⇒ <i>count(10m,6/7,band)</i> → nombre de valeurs des 10 dernières minutes ayant '110' (en binaire) dans les 3 bits de poids le plus faible.</p> <p>⇒ <i>count(10m,,,1d)</i> → nombre de valeurs pour les 10 minutes précédentes jusqu'à 24 heures</p> <p>Le paramètre <i>#num</i> est supporté depuis Zabbix 1.6.1.</p> <p>Le paramètre <i>time_shift</i> et les opérateurs de chaîne sont supportés depuis Zabbix 1.8.2.</p> <p>L'opérateur <i>band</i> est supporté depuis Zabbix 2.2.0.</p> <p>Les opérateurs <i>regex</i>, <i>iregexp</i> sont supportés depuis Zabbix 3.2.0.</p> |

| FONCTION  |   |   |
|---|---|---|
| Description   | Paramètres  | Commentaires  |
| <b>abschange</b>  |   |   |
| <b>date</b>   |   |   |
| Date actuelle au format YYYYMMDD.   |   | Types de valeurs supportés : <i>any</i><br>Exemple de valeur retournée : 20150731   |
| <b>dayofmonth</b>   |   |   |
| Jour du mois compris entre 1 et 31.   |   | Types de valeurs supportés : <i>any</i><br>Cette fonction est supportée depuis Zabbix 1.8.5.  |
| <b>dayofweek</b>  |   |   |
| Jour de le semaine compris entre 1 et 7 (Mon - 1, Sun - 7).   |   | Types de valeurs supportés : <i>any</i>   |
| <b>delta</b> (sec #num,<time_shift>)  |   |   |
| Différence entre les valeurs maximale et minimale dans la période d'évaluation définie ('max()' moins 'min()'). | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées spécifiées (précédées d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg() | Types de valeurs supportés : flottant, int<br>Le paramètre <code>time_shift</code> est supporté depuis Zabbix 1.8.2.                              |
| <b>diff</b>   |   |   |
| Vérifier si les dernières valeurs et les valeurs précédentes différent.   |   | Types de valeurs supportés : flottant, int, str, text, log<br>Retourne :<br>1 - la dernière valeur et la valeur précédente différent<br>0 - sinon |
| <b>forecast</b> (sec #num,<time_shift>,time,<fit>,<mode>)   |   |   |

| FONCTION   |  |  |
|--|--|--|
| Description  | Paramètres   | Commentaires   |
| <b>abschange</b>   |  |  |
| <p>Valeur future, max, min, delta ou avg de l'élément.</p> | <p><b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées spécifiées (précédées d'un signe dièse)<br/> <b>time_shift</b> (optionnel) - voir avg()<br/> <b>time</b> - horizon de prévision en secondes<br/> <b>fit</b> (optionnel) - fonction utilisée pour ajuster les données historiques</p> <p>Support de fits :<br/> <i>linéaire</i> - fonction linéaire<br/> <i>polynomialN</i> - polynomial de degrés N (1 &lt;= N &lt;= 6)<br/> <i>exponentiel</i> - fonction exponentielle<br/> <i>logarithmique</i> - fonction logarithmique<br/> <i>puissance</i> - fonction puissance</p> <p>Notez que :<br/> <i>linéaire</i> est par défaut, <i>polynomial1</i> est équivalent à <i>linéaire</i></p> <p><b>mode</b> (optionnel) - sortie demandée</p> <p>modes supportés :<br/> <i>valeur</i> - valeur (défaut)<br/> <i>max</i> - maximum<br/> <i>min</i> - minimum<br/> <i>delta</i> - max-min<br/> <i>avg</i> - moyenne</p> <p>Notez que :<br/> la <i>valeur</i> de l'élément est estimée au moment now + time.<br/> <i>max</i>, <i>min</i>, <i>delta</i> et <i>avg</i> étudient l'estimation de la valeur d'un élément dans l'intervalle entre now et now + time</p> | <p>Types de valeurs supportés : flottant, int</p> <p>Si la valeur retournée est plus grande que 99999999999.9999 ou plus petite que -99999999999.9999, la valeur retournée sera rognée à 99999999999.9999 ou -99999999999.9999 en conséquence.</p> <p>Devient non supporté uniquement s'il est mal utilisé dans une expression (type d'élément incorrect, paramètres non valides), sinon renvoie -1 en cas d'erreur.</p> <p>Exemples :<br/> ⇒ forecast(#10,,1h) → prévision de la valeur d'un élément après une heure basée sur les 10 dernières valeurs<br/> ⇒ forecast(1h,,30m) → prévision de la valeur de l'élément après 30 minutes basée sur les données de la dernière heure<br/> ⇒ forecast(1h,1d,12h) → prévision de l'élément après 12 heures sur une heure il y a un jour<br/> ⇒ forecast(1h,,10m,exponential) → prévision de la valeur de l'élément après 10 minutes sur la base des données de la dernière heure et de la fonction exponentielle<br/> ⇒ forecast(1h,,2h,polynomial3,max) → prévision de la valeur maximale que l'élément peut atteindre dans les deux heures à venir en fonction des données de la dernière heure et du polynôme cubique (troisième degré)<br/> ⇒ forecast(#2,-20m) → estimer la valeur d'un élément datant d'il y a 20 minutes en fonction des deux dernières valeurs (cela peut être plus précis que d'utiliser last() ou prev(), surtout si l'élément est mis à jour rarement, par exemple, une fois par heure)</p> <p>Cette fonction est supportée depuis Zabbix 3.0.0. Les valeurs négatives de time sont supportées depuis Zabbix 3.0.6 et 3.2.2. Voir aussi les informations additionnelles sur les <a href="#">fonction de déclencheur prédictives</a>.</p> |
| <b>fuzzytime</b> (sec)                                     |  |  |

| FONCTION   |  |   |
|--|--|---|
| Description  | Paramètres   | Commentaires  |
| <b>abschange</b>   |  |   |
| Vérification de la valeur d'un élément (son timestamp) par rapport à l'heure du serveur Zabbix.      | <b>sec</b> - secondes  | Types de valeurs supportés : flottant, int<br><br>Retourne :<br>1 - si la différence entre la valeur de l'élément (son timestamp) et le timestamp du serveur Zabbix est inférieure ou égale à T secondes<br>0 - sinon.<br><br>Habituellement utilisé avec l'élément 'system.localtime' pour vérifier que l'heure locale est synchronisée avec l'heure locale du serveur Zabbix. <i>Notez</i> que 'system.localtime' doit être configuré comme <a href="#">vérification passive</a> .<br>Peut aussi être utilisé avec la clé vfs.file.time[/path/file,modify] pour vérifier que le fichier n'a pas eu de mises à jour depuis longtemps.<br><br>Exemple :<br>⇒ fuzzytime(60)=0 → détecte un problème si la différence de temps est supérieure à 60 secondes |
| <b>iregexp</b> (<pattern>,<sec #num>)  |  |   |
| Cette fonction est identique à regexp() mais non sensible à la casse.                                | voir regexp()  | Types de valeurs supportés : str, log, text   |
| <b>last</b> (<sec #num>,<time_shift>)  |  |   |
| La valeur la plus récente.   | <b>sec</b> (ignoré, égal #1) ou <b>#num</b> (optionnel) - la Nième valeur la plus récente<br><b>time_shift</b> (optionnel) - voir avg()          | Types de valeurs supportés : flottant, int, str, text, log<br><br>Notez que #num fonctionne différemment ici de beaucoup d'autres fonctions.<br>Par exemple :<br>last() est toujours égal à last(#1)<br>last(#3) - troisième valeur la plus récente (et <i>non</i> trois dernières)<br><br>Zabbix ne garantit pas l'ordre exact des valeurs s'il existe plus de deux valeurs dans la seconde dans l'historique.<br><br>Le paramètre #num est supporté depuis Zabbix 1.6.2.<br>Le paramètre time_shift est supporté depuis Zabbix 1.8.2.   |
| <b>logeventid</b> (<pattern>)  |  |   |
| Vérifier si l'ID d'événement de la dernière entrée de journal correspond à une expression régulière. | <b>pattern</b> (optionnel) - expression régulière décrivant le modèle requis, style <a href="#">Expression régulière compatible Perl</a> (PCRE). | Types de valeurs supportés : log<br><br>Retourne :<br>0 - ne correspond pas<br>1 - correspond<br><br>Cette fonction est supportée depuis Zabbix 1.8.5.  |
| <b>logseverity</b>   |  |   |

| FONCTION  |   |  |
|---|---|--|
| Description   | Paramètres  | Commentaires   |
| <b>abschange</b>  |   |  |
| Consigner la gravité de la dernière entrée du journal.  |   | Types de valeurs supportés : log<br><br>Retourne :<br>0 - sévérité par défaut<br>N - sévérité (entier, utile pour les journaux d'événements Windows : 1 - Information, 2 - Avertissement, 4 - Erreur, 7 - Audit d'échec, 8 - Audit de succès, 9 - Critique, 10 - Verbeux).<br>Zabbix prend la sévérité du journal du champs <b>Information</b> du journal des événements Windows.  |
| <b>logsource</b> (<pattern>)  |   |  |
| Vérifier si la source de la dernière entrée du journal correspond à une expression régulière. | <b>pattern</b> (optionnel) - expression régulière décrivant le modèle requis, style <a href="#">Expression régulière compatible Perl</a> (PCRE).  | Types de valeurs supportés : log<br><br>Renvoie :<br>0 - ne correspond pas<br>1 - correspond<br><br>Normalement utilisé pour les journaux d'événements Windows. Par exemple, logsource("VMware Server").   |
| <b>max</b> (sec #num,<time_shift>)  |   |  |
| Valeur la plus élevée d'un élément dans la période d'évaluation définie.                      | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg()   | Types de valeurs supportés : flottant, int<br><br>Le paramètre <code>time_shift</code> est supporté depuis Zabbix 1.8.2.   |
| <b>min</b> (sec #num,<time_shift>)  |   |  |
| Valeur la plus basse d'un élément de la période d'évaluation définie.                         | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg()   | Types de valeurs supportés : flottant, int<br><br>Le paramètre <code>time_shift</code> est supporté depuis Zabbix 1.8.2.   |
| <b>nodata</b> (sec)   |   |  |
| Vérification de l'absence de données reçues.  | <b>sec</b> - période d'évaluation en secondes.<br>La période ne doit pas être inférieure à 30 secondes car le processus de synchronisation de l'historique calcule cette fonction uniquement toutes les 30 secondes.<br><br>nodata(0) n'est pas autorisé. | Types de valeurs supportés : <i>any</i><br><br>Retourne :<br>1 - si aucune donnée n'est reçue pendant la période définie<br>0 - sinon<br><br>Notez que cette fonction affichera une erreur si, dans la période du 1er paramètre :<br>- il n'y a pas de données et le serveur Zabbix a été redémarré<br>- il n'y a pas de données et la maintenance est terminée<br>- il n'y a pas de données et l'élément a été ajouté ou réactivé<br>Les erreurs sont affichées dans la colonne <i>Info</i> dans la <a href="#">configuration</a> du déclencheur. |
| <b>now</b>  |   |  |
| Nombre de secondes depuis l'époque (00:00:00 UTC, 1er janvier 1970).                          |   | Types de valeurs supportés : <i>any</i>  |
| <b>percentile</b> (sec #num,<time_shift>,percentage)  |   |  |

| FONCTION  |   |   |
|---|---|---|
| Description   | Paramètres  | Commentaires  |
| <b>abschange</b>  |   |   |
| P-ième centile d'une période, où P (pourcentage) est spécifié par le troisième paramètre. | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg()<br><b>percentage</b> - un nombre à virgule flottante compris entre 0 et 100 (inclus) avec un maximum de 4 chiffres après le point décimal     | Types de valeurs supportés : flottant, int<br><br>Cette fonction est supportée depuis Zabbix 3.0.0.   |
| <b>prev</b>   |   |   |
| Valeur précédente.  |   | Types de valeurs supportés : flottant, int, str, text, log<br><br>Retourne la même chose que last(#2).  |
| <b>regexp</b> (<pattern>,<sec #num>)  |   |   |
| Vérifier si la dernière valeur (la plus récente) correspond à l'expression régulière.     | <b>pattern</b> (optionnel) - expression régulière, style <a href="#">Expression régulière compatible Perl</a> (PCRE).<br><b>sec</b> ou <b>#num</b> (optionnel) - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse). Dans ce cas, plusieurs valeurs peuvent être traitées. | Types de valeurs supportés : str, text, log<br><br>Retourne :<br>1 - trouvé<br>0 - sinon<br><br>Si plus d'une valeur est traitée, '1' est renvoyé s'il existe au moins une valeur correspondante.<br><br>Cette fonction est sensible à la casse.  |
| <b>str</b> (<pattern>,<sec #num>)   |   |   |
| Recherche d'une chaîne dans la dernière valeur (la plus récente).                         | <b>pattern</b> (optionnel) - chaîne requise<br><b>sec</b> ou <b>#num</b> (optionnel) - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse). Dans ce cas, plusieurs valeurs peuvent être traitées  | Types de valeurs supportés : str, text, log<br><br>Retourne :<br>1 - trouvé<br>0 - sinon<br><br>Si plus d'une valeur est traitée, '1' est renvoyé s'il existe au moins une valeur correspondante.<br><br>Cette fonction est sensible à la casse.  |
| <b>strlen</b> (<sec #num>,<time_shift>)   |   |   |
| Longueur de la dernière valeur (la plus récente) en caractères (pas en octets).           | <b>sec</b> (ignoré, égal #1) ou <b>#num</b> (optionnel) - la Nième valeur la plus récente<br><b>time_shift</b> (optionnel) - voir avg()   | Types de valeurs supportés : str, text, log<br><br>Notez que #num fonctionne différemment ici de beaucoup d'autres fonctions.<br><br>Exemples :<br>⇒ strlen() (est égal à strlen(#1)) → longueur de la dernière valeur<br>⇒ strlen(#3) → longueur de la troisième valeur la plus récente<br>⇒ strlen(,1d) → longueur de la dernière valeur d'un jour auparavant.<br><br>Cette fonction est supportée depuis Zabbix 1.8.4. |
| <b>sum</b> (sec #num,<time_shift>)  |   |   |



| FONCTION  |  |   |
|---|--|---|
| Description   | Paramètres   | Commentaires  |
| <b>abschange</b>  |  |   |
| Somme des valeurs collectées au cours de la période d'évaluation définie.   | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées (précédée d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg()  | Types de valeurs supportés : flottant, int<br>Le paramètre <code>time_shift</code> est supporté depuis Zabbix 1.8.2.  |
| <b>time</b>   |  |   |
| Heure actuelle au format HHMMSS.  |  | Types de valeurs supportés : <i>any</i><br>Exemple de valeur retournée : 123055   |
| <b>timeleft</b> (sec #num,<time_shift>,threshold,<fit>)                     |  |   |
| Durée en secondes nécessaire à un élément pour atteindre un seuil spécifié. | <b>sec</b> ou <b>#num</b> - période d'évaluation en secondes ou dans les dernières valeurs collectées spécifiées (précédées d'un signe dièse)<br><b>time_shift</b> (optionnel) - voir avg()<br><b>threshold</b> - valeur à atteindre<br><b>fit</b> (optionnel) - voir forecast() | Types de valeurs supportés : flottant, int<br><br>Si la valeur à renvoyer est supérieure à 9999999999.9999, la valeur de retour est rognée à 9999999999.9999.<br><br>Retourne 9999999999.9999 si le seuil ne peut pas être atteint.<br><br>Deviens non supporté uniquement s'il est mal utilisé dans une expression (type d'élément incorrect, paramètres non valides), sinon renvoie -1 en cas d'erreur.<br><br>Exemples :<br>⇒ <code>timeleft(#10,,0)</code> → temps jusqu'à ce que la valeur de l'élément atteigne zéro sur la base des 10 dernières valeurs<br>⇒ <code>timeleft(1h,,100)</code> → temps jusqu'à ce que la valeur de l'élément atteigne 100 sur la base des données de la dernière heure<br>⇒ <code>timeleft(1h,1d,0)</code> → temps jusqu'à ce que la valeur de l'élément atteigne 0 sur la base d'une heure il y a un jour<br>⇒ <code>timeleft(1h,,200,polynomial2)</code> → temps jusqu'à ce que l'élément atteigne 200 sur la base des données de la dernière heure et en supposant que l'élément se comporte comme un polynôme quadratique (au deuxième degré)<br><br>Cette fonction est supportée depuis Zabbix 3.0.0. Les <a href="#">symboles d'unités</a> dans le paramètre <code>threshold</code> sont supportés depuis Zabbix 3.0.6 et 3.2.2.<br>Voir aussi les informations supplémentaires sur les <a href="#">fonctions de déclenchement prédictives</a> . |

- 1) Toutes les fonctions renvoient uniquement des valeurs numériques. Par exemple, la comparaison avec des chaînes n'est pas supportée.
- 2) Certaines fonctions ne peuvent pas être utilisées pour des paramètres non numériques !
- 3) Les arguments de chaîne doivent être entre guillemets doubles. Sinon, ils pourraient être mal interprétés.
- 4) Pour toutes les fonctions de déclencheur **sec** et **time\_shift** doivent être un entier avec un [suffixe temporel](#) facultatif et n'ont absolument rien à voir avec le type de données de l'élément.

## Fonctions et éléments non supportés

À partir de Zabbix 3.2, les fonctions **nodata()**, **date()**, **dayofmonth()**, **dayofweek()**, **now()** et **time()** sont également calculées pour les éléments non supportés. D'autres fonctions nécessitent que l'élément référencé soit dans un état supporté.

From:

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

Permanent link:

<https://www.zabbix.com/documentation/4.0/fr/manual/appendix/triggers/functions>

Last update: **2019/03/26 13:15**

