
IBM Communications Server pour Linux V6.2.1 - Nouveautés

Le présent document décrit les modifications apportées à la maintenance de la version 6.2.1.0 de Communications Server pour Linux (Produit N° 5724-i33, CS Linux) et de Communications Server pour Linux sur zSeries (Produit N° 5724-i34, CS Linux sur zSeries). Les descriptions concernent les produits Communications Server pour Linux et Communications Server pour Linux sur zSeries.

Les modifications contenues dans cette version de maintenance destinée à Communications Server pour Linux V6.2.1.0 comprennent, notamment :

1. Prise en charge du noyau 2.6 - Mise à jour de la prise en charge du système d'exploitation Linux
2. Linux sur Power - Nouveau packaging et prise en charge supplémentaire de la plate-forme
3. Remote API Client AIX - Nouveau packaging et plate-forme supplémentaire
4. Prise en charge de LU primaire - Interface de programmation LUA permettant de prendre en charge les flux LU 0 primaires
5. Mises à jour des clients Remote API Client - Mises à jour destinées aux clients
6. Mises à jour du guide de programmation LUA - Interface LU primaire supplémentaire
7. Mises à jour du manuel Administration Guide - Commandes supplémentaires permettant de définir et d'interroger les paramètres de l'horloge HPR
8. Mises à jour du manuel NOF Programmer's Guide - Instructions supplémentaires permettant de définir et d'interroger les paramètres de l'horloge HPR

Cette nouvelle fonction est décrite plus en détails ci-après.

Prise en charge du noyau 1. 2.6

Les produits CS Linux et CS Linux sur zSeries prennent désormais en charge des installations serveur sur les distributions basées sur un noyau Linux 2.6 à partir de Red Hat et SuSE. Les distributions prises en charge pour les installations serveur sont RHEL 4 et SLES 9-SP1. Lorsque vous installez le produit CS Linux sur un système Intel, le noyau doit être une distribution Linux 32 bits. Lorsque vous installez le produit CS Linux pour un système Linux sur Power (OpenPower ou Power5), le noyau doit être une distribution Linux 64 bits. Le produit CS Linux sur zSeries peut être installé sur un noyau Linux 31 ou 64 bits. Cependant, à l'avenir, les distributions Linux pour la plate-forme zSeries seront uniquement destinées à des noyaux 64 bits. Les applications peuvent utiliser des bibliothèques 32 ou 64 bits.

Pour Remote API Client, qui disposait d'une prise en charge de distribution de noyau Linux 2.6, les applications SNA client peuvent uniquement appeler des bibliothèques 32 bits installées sur des plates-formes Intel. Ainsi, si l'application client fonctionne sur une distribution Linux x86_64, seules les bibliothèques 32 bits peuvent être utilisées. Pour les plates-formes Linux sur zSeries et Linux sur Power, les applications SNA client peuvent utiliser des bibliothèques 32 ou 64 bits.

Les distributions basées sur un noyau Linux qui sont prises en charge sur différentes plates-formes sont mappées dans le tableau ci-dessous :

Tableau 1. Distributions basées sur un noyau Linux prises en charge

Plate-forme	RHAS 2.1	SLES 8	RHEL 3	SLES 9	RHEL 4	AIX V5
Intel						
Client (i686 - 32 bits)	x	x	x	x	x	
Client (x86_64), API 32 bits		x	x	x	x	

Tableau 1. Distributions basées sur un noyau Linux prises en charge (suite)

Plate-forme	RHAS 2.1	SLES 8	RHEL 3	SLES 9	RHEL 4	AIX V5
Serveur (i686 - 32 bits)	x	x	x	x	x	
<i>OpenPower ou Power 5 (noyau 64 bits, ppc64)</i>						
Client				x	x	
Serveur				x	x	
Linux sur zSeries (s390, 31 bits)						
Client		x	x	x	x	
Serveur		x	x	x*	x*	
Linux sur zSeries (s390x, zSeries, 64 bits)						
Client		x	x	x	x	
Serveur		x	x	x	x	
AIX						
Client - 32 bits						x
Client - 64 bits						x

* indique une prise en charge qui pourra être déconseillée dans les futures versions de Linux.

2. Linux sur Power

Le produit CS Linux (5724-i33) offre désormais une prise en charge multi-plateformes comprenant notamment le serveur CS Linux pour des plates-formes Linux sur Power, telles que les serveurs OpenPower et Power5 pSeries. Lorsque vous installez le serveur, la prise en charge de ces plates-formes nécessite RHEL 4 ou SLES 9-SP1. Pour installer un serveur CS Linux sur une plate-forme Linux sur Power, voir le fichier README.ppc64.xx_yy (où xx_yy correspond à l'environnement local pour ce système).

Remote API Client de CS Linux pour Linux sur Power est livré avec les produits CS Linux et CS Linux pour zSeries. La nouvelle prise en charge du client se trouve dans le répertoire /ibm-commserver-clients/linux-ppc64 sur le disque d'installation. Remote API Client s'exécute dans l'espace utilisateur sur Linux et n'est pas dépendant du noyau. Pour installer le client, voir le fichier Readme (/ibm-commserver-clients/linux-ppc64/README).

3. Remote API Client AIX

Les serveurs Communications Server qui fonctionnent dans un domaine de serveurs prenant en charge des clients Remote API Client prennent désormais en charge les applications SNA s'exécutant sur des plates-formes AIX V5. Lors du développement d'applications pour la plate-forme AIX, vous devez utiliser, pour la compilation et la liaison, la même description. Cette description figure dans la section Communications Server pour AIX (<http://www.ibm.com/software/network/commserver/library/aix>).

La connexion entre le client AIX et les serveurs Linux s'effectue via TCP/IP. Dans le cadre de Communications Server pour AIX V5 ou ultérieure, toute application SNA sur AIX utilisant les interfaces CPI-C, APPC, LUA et certaines fonctions NOF, peut bénéficier de la solution Remote API Client AIX. Les interfaces CS/AIX V4.2 les plus anciennes ne sont pas prises en charge par le Remote API Client.

4. Prise en charge de LU primaire

Les applications LUA sont généralement connectées à des des systèmes hôtes en tant que LU secondaires. La définition des sessions est donc contrôlée par l'application hôte qui envoie l'option BIND pour démarrer la session. Communications Server peut désormais agir en tant que LU primaire pour les équipements dépendants d'applications SNA en aval via les interfaces LAN et à l'aide de l'interface RUI_INIT_PRIMARY. Cette interface permet à une application de connecter des sessions LU dépendantes en aval sans recourir à un système hôte.

Pour l'utilisation des applications LU primaires, la configuration du noeud doit inclure les LU en aval (ou un modèle de PU en aval) avec le nom de LU hôte : #PRIRUI#. Cela indique au serveur que les applications utilisant RUI_INIT_PRIMARY contrôlent les PU et les ressources LU qui leur sont affectées. L'utilisation des PU est limitée aux ports LAN. Voir les mises à jour du [guide de programmation LUA](#) (ci-dessous) concernant les informations liées à l'interface de programmation des applications pour la prise en charge de la LU primaire.

5. Mises à jour des clients Remote API Client

5.1 Modification du nom

Communications Server Remote API Client devient "IBM Remote API Client".

5.2 Installation des clients Linux et AIX

Si vous installez le serveur ou le client Communications Server sur un système déjà doté d'une version antérieure, vous devez tout d'abord désinstaller le serveur ou le client existant. Les fichiers de configuration ne seront pas supprimés ou modifiés par ce processus de désinstallation et installation.

5.3 Clients Windows

Si vous installez le client Communications Server sur un système déjà doté d'une version antérieure, vous devez, afin de préserver votre configuration actuelle, procéder comme suit :

- Arrêtez le client Communications Server en exécutant la commande **:net stop sxclient**. Cette commande doit être saisie tel quel (sans être traduite).
- Fermez l'icône Monitor du client Windows(généralement située dans le coin inférieur droit du bureau).
- Installez le produit sans supprimer la version précédente.

Si vous supprimez le client Windows existant, les informations de configuration sont supprimées de la base de registres Windows et vous devez les saisir de nouveau, une fois la nouvelle version installée.

IBM Remote API Client pour Windows comporte désormais un outil de diagnostic d'état, **snagetpd.exe**. Cet outil crée un fichier compressé, **snapt.exe**. C'est un fichier auto-extractible. Il contient :

- des informations pertinentes concernant les répertoires,
- des informations pour les répertoires concernant tout ce qui est situé dans le répertoire d'installation ou en-dessous,
- des informations sur les versions de tous les fichiers binaires installés,
- les emplacements de tous les fichiers journaux et fichiers de trace,
- les sorties des commandes 'ver', 'ipconfig /all', 'route print', 'netstat -an' et 'netstat -a'.

L'outil copie tous les fichiers journaux et fichiers de trace dans snapd. Si l'exécution de Communications Server pour Linux demande de fournir des informations d'identification des incidents, vous devez envoyer un fichier snapd.exe. Ce fichier aidera à la résolution des incidents rapportés.

Le correctif APAR résout les incidents relatifs au client Remote API Client Windows ; ces incidents sont les suivants :

- LI70604, LI70605 - L'installation a échoué sur certains systèmes Windows XP dont les paramètres nationaux n'ont pas été définis.
- LI70677, LI70678 - L'unité d'exécution principale de l'application WinCPIC ne peut pas exécuter plus d'une commande ALLOCATE.

6. Mises à jour du guide de programmation LUA

Les informations suivantes complètent le [guide de programmation LUA](#) et décrivent l'interface du programme RUI_INIT_PRIMARY. Consultez la partie consacrée à RUI_INIT du [guide de programmation LUA](#) pour tout aspect ou fonction qui n'est pas traité entièrement dans cette section.

6.1 Conception et développement d'applications LUA : Informations sur SNA pour les applications RUI primaires

Cette section traite des aspects relatifs à SNA dans le cadre du développement d'applications RUI primaires CS Linux pour communiquer avec les LU en aval.

Il ne s'agit pas d'une explication détaillée des concepts SNA. Si vous avez besoin d'informations spécifiques sur les flux de messages SNA, consultez la documentation fournie avec l'application hôte pour laquelle vous concevez l'application LUA CS Linux.

6.2 Responsabilités de l'application RUI primaire

Une application RUI primaire contrôle à la fois les sessions LU-SSCP et PLU-SLU au niveau de l'unité requête/réponse(RU). Elle peut, en outre, envoyer et recevoir des RU SNA sur ces sessions. La session PU-SSCP est interne à CS Linux et l'application RUI primaire ne peut y accéder.

L'application RUI primaire fonctionnant au niveau RU, elle dispose de fonctions de contrôle étendues sur le flot de données échangées avec la LU secondaire. Cependant, elle joue un rôle plus important qu'une application LUA classique dans la vérification de la validité des messages SNA envoyés et du bon emploi des protocoles au niveau RU (par exemple, mise entre parenthèses et chaînage). En particulier, CS Linux ne vérifie pas la validité des RU envoyées par une application RUI primaire.

L'application RUI primaire est chargée :

- de l'initialisation des LU en aval à l'aide de RUI_INIT_PRIMARY et de leur arrêt à l'aide de RUI_TERM,
- du traitement des messages de notification provenant de la LU secondaire en tant que démarrage et arrêt d'applications secondaires,
- du traitement de INIT-SELF et TERM-SELF pour activer et désactiver la session PLU-SLU,
- de la création, de l'envoi, de la réception et de l'analyse syntaxique de 3 270 messages de flot de données dans les RU de données,
- de la mise en oeuvre des protocoles au niveau RU (contrôle de requête, mise entre parenthèses, chaînage et sens),
- de la cryptographie (si nécessaire),
- de la compression (si nécessaire).

6.3 Restrictions

CS Linux ne prend pas en charge les applications RUI primaires suivantes :

- PU en aval sur demandeur DLU
- LU dépendantes définies dynamiquement (DDDLU)
- STSN d'envoi (pour redéfinir des numéros de séquence, l'application doit déconnecter (UNBIND), puis connecter (BIND) à nouveau la session)

6.4 Informations de configuration : Configuration de la liaison RUI_INIT_PRIMARY

Dans le cas d'applications LU primaires, chaque LU contrôlée par l'application doit être configurée en tant que LU en aval(ou modèle de PU en aval) dont le nom de LU hôte est : #PRIRUI#.

6.5 RUI_VERBS : Description du contrôle de l'instruction RUI_INIT_PRIMARY

L'instruction RUI_INIT_PRIMARY lance la session SSCP-LU pour une application SNA primaire communiquant avec une LU en aval. Si l'application RUI joue le rôle de SNA secondaire et communique avec une LU hôte, elle doit utiliser RUI_INIT, et non RUI_INIT_PRIMARY.

6.5.1 Paramètres pré-définis

L'application fournit les paramètres suivants (Voir /opt/ibm/sna/include/lua_c.h) :

lua_verb

LUA_VERB_RUI

lua_verb_length

Longueur en octets de l'enregistrement de l'instruction LUA. Définissez-la pour la taille (LUA_COMMON).

lua_opcode

LUA_OPCODE_RUI_INIT_PRIMARY

lua_correlator

LUA_OPCODE_RUI_INIT_PRIMARY

lua_luname

Nom ASCII de la LU pour laquelle vous souhaitez démarrer une session. Ce nom doit correspondre au nom d'une LU en aval configurée en vue d'une utilisation avec la passerelle SNA ou à celui d'une LU créée par défaut à partir d'un modèle de LU en aval et dont le nom de LU hôte est : #PRIRUI#.

Ce paramètre doit correspondre à 8 octets. Il doit être saisi en partant de la droite, avec des espaces et en 0x20, si le nom a moins de huit caractères.

lua_max_length

Taille du tampon fourni pour la copie de la RU ACTLU(+RSP) obtenue à partir de la PU en aval. Si l'application n'a pas besoin de cette copie, il est possible d'indiquer un pointeur NULL pour le paramètre lua_data_ptr et dans ce cas, il n'est pas nécessaire de fournir un tampon de données.

lua_data_ptr

Pointeur désignant le tampon fourni pour la copie de la RU ACTLU(+RSP) obtenue à partir de la PU en aval. Si l'application n'a pas besoin de cette copie, un pointeur NULL peut être défini et l'information ne sera pas renvoyée.

lua_post_handle

Pointeur désignant une routine de rappel. Si l'instruction s'exécute de manière asynchrone, LUA appelle cette routine afin d'indiquer l'exécution de l'instruction. Pour plus d'informations, voir la section Conception et développement d'applications LUA .

lua_encr_decr_option

- 0 La cryptographie n'est pas utilisée au niveau session.
- 128 Le chiffrement et le déchiffrement sont effectués.

Remarque : Le chiffrement et le déchiffrement sont effectués par le programme applicatif.

Toute autre valeur renvoie un code retour LUA_ENCR_DECR_LOAD_ERROR.

6.5.2 Paramètres renvoyés

LUA renvoie toujours les paramètres suivants :

lua_flag2.async

Si l'exécution de l'instruction est asynchrone, cette option prend la valeur 1 ; si elle est synchrone, l'option prend la valeur 0. L'exécution de RUI_INIT_PRIMARY est toujours asynchrone, sauf si elle renvoie une erreur du type LUA_PARAMETER_CHECK.

Les autres paramètres renvoyés dépendent du succès de l'exécution de l'instruction ; voir les sections ci-après.

Exécution réussie

Si l'instruction s'exécute correctement, LUA renvoie les paramètres suivants :

lua_prim_rc

LUA_OK

lua_sid

ID session pour la nouvelle session. Cet ID pourra être utilisé par les instructions ultérieures pour identifier cette session.

lua_data_length

Taille de la RU ACTLU(+RSP) obtenue à partir de la PU en aval. LUA envoie les données dans le tampon indiqué par lua_data_ptr.

Echec de l'exécution

Si l'instruction ne s'exécute pas correctement, LUA renvoie un code retour primaire indiquant le type d'erreur et un code retour secondaire détaillant la cause de l'échec de l'exécution.

Voir la section relative aux échecs d'exécution de RUI_INIT du guide de programmation LUA pour consulter la liste des codes retour possibles. En outre, concernant les erreurs particulières à RUI_INIT_PRIMARY, les valeurs suivantes peuvent être renvoyées :

lua_prim_rc

LUA_STATE_CHECK

lua_sec_rc

LUA_DUPLICATE_RUI_INIT_PRIMARY

Une instruction RUI_INIT_PRIMARY est en cours de traitement pour cette session.

6.5.3 Interaction avec d'autres instructions

L'instruction RUI_INIT_PRIMARY doit être la première instruction LUA exécutée sur la session.

Tant que cette instruction ne s'est pas exécutée avec succès, la seule autre instruction LUA qui peut être exécutée sur cette session est RUI_TERM. Celle-ci mettra fin à une instruction RUI_INIT_PRIMARY en cours.

Toute autre instruction exécutée sur cette session doit identifier la session à l'aide de l'un des paramètres suivants renvoyés par l'instruction :

- L’ID session, renvoyé à l’application dans le paramètre lua_sid
- Le nom de LU, fourni par l’application dans le paramètre lua_luname

6.5.4 Emploi et Restrictions

L’instruction RUI_INIT_PRIMARY s’exécute si la LU en aval renvoie une réponse positive ACTLU. Le cas échéant, l’instruction attend le temps nécessaire. Si l’application doit vérifier le contenu de la réponse positive ACTLU, CS Linux renvoie le contenu du message reçu dans le tampon de données fourni par l’application pour RUI_INIT_PRIMARY (à l’aide des paramètres lua_max_length et lua_data_ptr).

Une fois l’exécution de l’instruction RUI_INIT_PRIMARY réussie, la session utilise la LU pour laquelle elle a été démarrée. Aucune autre session LUA (à partir de cette application ou de toute autre) peut utiliser la LU tant que l’instruction RUI_TERM n’est pas exécutée, ou qu’un code retour primaire LUA_SESSION_FAILURE n’est pas reçu.

Si l’instruction RUI_INIT_PRIMARY renvoie un code retour primaire LUA_IN_PROGRESS, l’ID session est renvoyé dans le paramètre lua_sid. Cet ID session est identique à celui renvoyé lors de la réussite de l’exécution de l’instruction. Il peut, en outre, s’utiliser avec l’instruction RUI_TERM pour mettre fin à une instruction RUI_INIT_PRIMARY en suspens.

7. Mises à jour du manuel Administration Guide

Il existe deux nouvelles commandes pour l’outil d’administration snaadmin. Ces commandes sont les suivantes :

define_rtp_tuning

Commande qui définit de nouvelles horloges pour le réglage de la connectivité de la session HPR.

query_rtp_tuning

Commande qui interroge les horloges de changement de chemin HPR

En vue de leur utilisation, vous pouvez voir la syntaxe de ces nouvelles commandes en exécutant la commande d’aide "snaadmin -d -h define_rtp_tuning" ou "snaadmin -d -h query_rtp_tuning". Pour des informations plus détaillées, voir également les mises à jour du manuel [NOF Programmer’s Guide](#).

8. Mises à jour du manuel NOF Programmer’s Guide

8.1 DEFINE_RTP_TUNING

DEFINE_RTP_TUNING définit les paramètres à utiliser pour la configuration des connexions RTP. Une fois cette instruction exécutée, les paramètres que vous avez définis seront utilisés pour toutes les connexions RTP à venir, jusqu’à ce que vous les modifiez en exécutant une nouvelle instruction DEFINE_RTP_TUNING.

8.1.1 Paramètres pré-définis

L’application fournit les paramètres suivants (voir la structure define_rtp_tuning dans /opt/ibm/sna/include/nof_c.h):

opcode

AP_DEFINE_RTP_TUNING

path_switch_attempts

Nombre de tentatives de changement de chemin à définir pour les nouvelles connexions RTP.

Indiquez une valeur comprise entre 1 et 255. Si vous indiquez la valeur 0(zéro), CS Linux utilise la valeur par défaut 6.

short_req_retry_limit

Nombre d’envois de requête de statut avant que CS Linux constate l’arrêt d’une connexion RTP et

lance le processus de changement de chemin. Indiquez une valeur comprise entre 1 et 255. Si vous indiquez la valeur 0(zéro), CS Linux utilise la valeur par défaut 6.

path_switch_times

Durée en secondes pendant laquelle CS Linux tente de changer de chemin pour la connexion RTP déconnectée. Ce paramètre est déterminé à l'aide de quatre délais distincts correspondants à chaque priorité de transmission valide. Ces priorités sont, dans l'ordre : AP_LOW, AP_MEDIUM , AP_HIGH et AP_NETWORK. Chacun de ses quatre éléments doit être compris entre 1 et 65 535. La valeur que vous affectez à chacune des priorités de transmission ne doit pas dépasser la valeur attribuée à toute priorité de transmission de niveau inférieur.

Si vous attribuez la valeur 0(zéro) à l'une de ces priorités, CS Linux utilise les valeurs par défaut correspondantes comme suit :

- 480 secondes (8 minutes) pour AP_LOW
- 240 secondes (4 minutes) pour AP_MEDIUM
- 120 secondes (2 minutes) pour AP_HIGH
- 60 secondes (1 minute) pour AP_NETWORK

8.1.2 Paramètres renvoyés

Exécution réussie

Si l'instruction s'exécute correctement, CS Linux renvoie les paramètres suivants :

primary_rc
AP_OK

Echec de l'exécution

Si l'exécution de l'instruction échoue en raison d'une erreur de paramètre, CS Linux renvoie les paramètres suivants :

primary_rc
AP_PARAMETER_CHECK

secondary_rc

Les valeurs possibles sont :

AP_INVALID_PATH_SWITCH_TIMES Le paramètre path_switch_times est incorrect ; par exemple, vous avez peut-être défini une valeur pour l'une des priorités de transmission qui dépasse celle indiquée pour une priorité de transmission inférieure.

Les codes retour Common répertorient les codes retour secondaires associés à AP_PARAMETER_CHECK et qui sont communs à toutes les instructions NOF.

8.2 QUERY_RTP_TUNING

QUERY_RTP_TUNING renvoie des informations relatives aux paramètres qui seront utilisés lors des prochaines connexions RTP. Ces informations ont été préalablement définies à l'aide de DEFINE_RTP_TUNING.

8.2.1 Paramètres pré-définis

L'application fournit les paramètres suivants (voir la structure query_rtp_tuning dans /opt/ibm/sna/include/nof_c.h) :

opcode
AP_QUERY_RTP_TUNING

8.2.2 Paramètres renvoyés

Exécution réussie

Si l'instruction s'exécute correctement, CS Linux renvoie les paramètres suivants :

primary_rc
AP_OK

path_switch_attempts
Nombre de tentatives de changement de chemin à définir pour les nouvelles connexions RTP.

short_req_retry_limit
Nombre d'envois de requête de statut avant que CS Linux constate l'arrêt d'une connexion RTP et lance le processus de changement de chemin.

path_switch_times
Durée en secondes pendant laquelle CS Linux tente de changer de chemin pour la connexion RTP déconnectée. Ce paramètre est déterminé à l'aide de quatre délais distincts correspondants à chaque priorité de transmission valide. Ces priorités sont, dans l'ordre : AP_LOW, AP_MEDIUM , AP_HIGH et AP_NETWORK.

Autres conditions

Les codes retour Common répertorient les combinaisons de codes retour primaires et secondaires qui sont communs à toutes les instructions NOF.