DNS : Revue rapide

Par : Abdel YEZZA, Ph.D

Date : Février 2007
Version : 1.0

Sommaire

Liste des figures

Liste des exemples

DNS : Eléments de base

Introduction

Avant tout, j'attire l'attention du lecteur, que des termes susceptibles d'être considérés nouveaux par ce dernier, pouvant être évoqués sans être définis au préalable. Toutefois, je rassure le lecteur qu'ils seront définis dans les sections suivantes et au moment opportun.

Le **DNS** (**D**omain **N**aming **S**ystem) est particulièrement utilisé en zone **Internet** (zone public) et en zone **Intranet** (zone privée de l'entreprise). Il s'agit d'un système standardisé et indépendant de la plate-forme qui lui fait appel malgré que les implémentations de celui-ci et les fonctionnalités offertes par ces dernières, diffèrent d'une plate-forme à une autre.

Le premier objectif du DNS est de résoudre les noms hiérarchiques pour trouver leurs adresses IP et vice-versa ; trouver l'adresse IP à partir du nom d'un host ou inversement, trouver le nom à partir d'une adresse IP.

Le DNS est utilisé par plusieurs services :

- **AD** (Active Directory) utilise le DNS pour nommer les domaines (au sens AD), les clients (serveurs membres ou **PDT** : Poste De Travail faisant partie d'un domaine)
- **AD** utilise aussi le DNS pour localiser les hosts DNS à partir de leurs adresses IP
- Les services et les composants AD comme **LDAP**, **Kerberos**, les **DC** (Contrôleurs de domaine) utilisent aussi le DNS pour déterminer les détenteurs des rôles (comme le **GC** : Global Catalog) et services demandés (comme le **PDC** : Primary Domain Controller)

L'architecture du DNS est basée sur une application du type **Client/Serveur**. Le Client interroge le Serveur et ce dernier lui répond avec les données appropriées. Le cheminement que la requête prend avant de retourner une réponse peut contenir plusieurs allées-retours avant d'aboutir en échec ou en succès. Un **PDT** (Poste De Travail) ou un serveur membre d'un domaine basé sur Windows 2000/2003/2008 doit être authentifié par un DC se trouvant dans le même site que le poste si possible, ou dans le site le plus proche de celui-ci (ayant la meilleure connectivité réseau de préférence). Afin qu'il puisse être authentifié, il doit obtenir l'adresse IP d'au moins un DC ; c'est là où le serveur DNS intervient pour rechercher dans sa base de données afin de répondre au Client (PDT ou serveur) et lui fournir l'adresse IP d'un DC. Une telle requête est réalisée en utilisant le protocole **LDAP** (Lightweight Directory Access Protocol). Voici un exemple d'enregistrement constituant une réponse :

Name	Type	Priority	Weight	Port	Data
_ldap	**SRV**	**0**	**100**	**389**	**DC-Name.MyDomain.com**

Nous détaillerons plus dans la suite chaque partie de cet enregistrement et nous parlerons des différents types pouvant être enregistrés dans la base DNS.

Le DNS le plus répondu au monde est celui basé sur Unix : Unix-based **BIND** (Berkley Internet Naming Domain) service. Il est traditionnellement basé sur des bases de données représentées par des fichiers texte. A partir de Windows 2000, Microsoft a introduit nativement un nouveau type de DNS appelé **Active Directory-Integrated DNS** et continue de supporter d'autres

produits qui gère le DNS. Toutefois, des fonctionnalités propres au DNS intégré dans AD, ne sont compatibles qu'avec la **version 8.2.1 de BIND** et plus, comme le service de localisation des enregistrements de type **SRV**, les mises à jour dynamiques des enregistrements DNS (**DDNS**) etc. Par ailleurs, un nombre important d'entreprises ont opté pour le maintien de systèmes DNS BIND malgré qu'elles utilisent un **SI** (Système d'Information) basé en partie ou entièrement sur des infrastructures Microsoft. La raison est simple, BIND a fait ses preuves de stabilité et de robustesse dans le passé, d'une part, et d'autre part, BIND était déjà en place avant de migrer les plates-formes vers Windows.

Organisation logique du DNS

On peut voir la hiérarchie logique du DNS comme le système de fichier sur une partition de disque (arborescence contenant des dossiers, sous-dossiers et fichiers). Le DNS commence par un point (.) représentant le nœud le plus haut dans la hiérarchie appelé communément le **nœud racine**, comme le lecteur courant (C:\ ou \ tout simplement). Les sous-dossiers contenus dans C:\ correspondent aux **nœuds** organisés sur la base **Parent/Fils**. Une telle organisation évite sûrement toute collision possible, i.e., deux nœuds distincts ne peuvent jamais avoir le même nom complet (**Nom DNS**) contrairement aux noms plats. Néanmoins, ce même modèle permet aussi de faire pointer un nœud vers un autre enregistrement (nœud), ce qu'on appelle **aliasing**. Le nom DNS de chaque nœud est représenté par le chemin allant du nœud courant jusqu'au nœud racine (**du bas vers le haut**) où chaque deux nœuds sont séparés par un point (.). Le nom DNS est qualifié par l'appellation **FQDN** (Fully Qualified Distinguished Name). Afin d'illustrer toutes ces notions, voici un exemple :

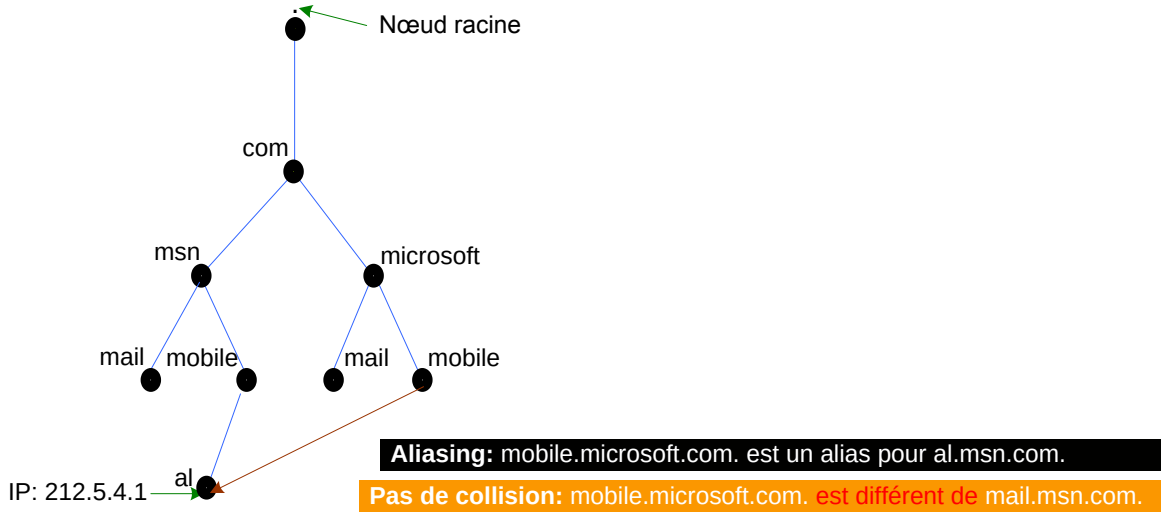

Figure 1 : Organisation logique d'un DNS

L'accent sera mis particulièrement dans la suite de ce document sur le DNS comme implémenté par Microsoft et notamment l'**AD-Integrated DNS**.

DNS Namespace (Espace de nom DNS), Domaine, Sous-domaine et Nœud

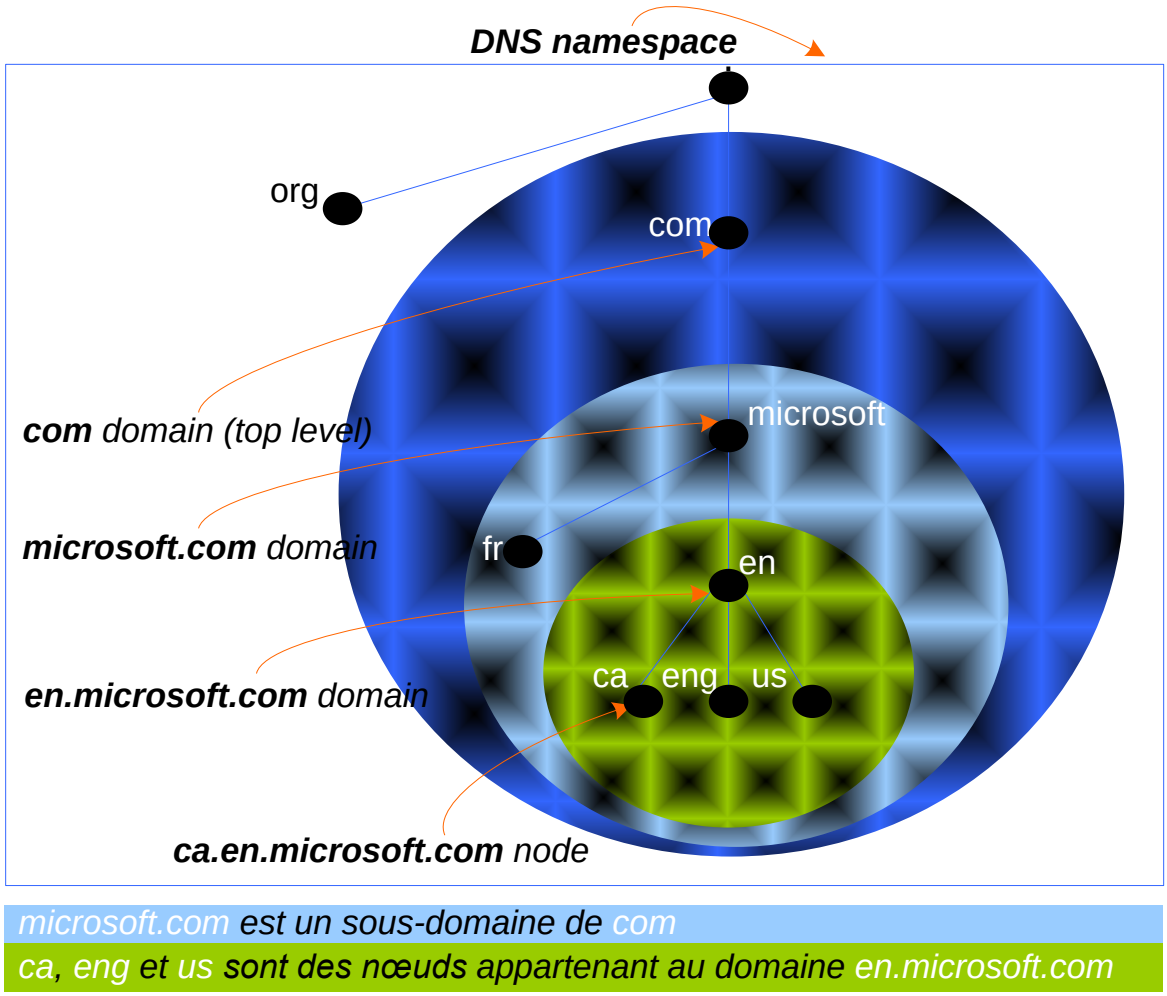

Figure 2 : Vue d'ensemble

NOTE : A ne pas confondre un domaine DNS avec un domaine au sens Active Directory. Tout domaine AD est nécessairement un domaine DNS, mais l'inverse n'est pas vrai ; on peut avoir des domaines DNS ne représentant pas des domaines AD.

Zones

Une zone représente une partie de la base de données stockant un domaine DNS et éventuellement des sous-domaines. Dans la figure ci-dessus (Figure 2 : Vue d'ensemble), le nom DNS en.microsoft.com est un sous-domaine du domaine microsoft.com. Ainsi, on peut définir une zone contenant le domaine en.microsoft.com et une autre zone contenant le domaine microsoft.com. On peut voir une zone comme étant un conteneur pour les principaux éléments formant le DNS comme le **SOA**, le **NS** et les différents types d'enregistrements (**A**, **PTR**, **MX** etc.) qui vont être évoqués dans une section ci-dessous.

L'intérêt principal de définir plusieurs zones est de pouvoir distribuer la base de données DNS sur plusieurs serveurs stockant les différentes zones et du même coup répartir le trafic issu des requêtes DNS envoyées par les clients (systèmes et applications).

Types de zones en fonction de leurs positions et propriétés

Tout d'abord, nous dirons qu'un DNS est du type **AD-Integrated**, si toutes les données du DNS sont incluses dans la base AD et du coup bénéficie des fonctionnalités du service AD comme la réplication (par domaine ou sur toute la forêt), la robustesse, la sécurisation, etc. Pour plus de détails sur les avantages du **AD-Integrated DNS**, vous pouvez consulter l'article chez Microsoft : KB555993.

Nous distinguons trois types de zones en fonction du propriétaire (serveur abritant la zone, **NS**) et des actions permises sur celle-ci :

Type de la Zone	Explication	Peut être AD-Integrated ?
Primary zone (Zone principle)	Il s'agit d'une zone pouvant être accédée en écriture (ajout) et en modification (mise à jour et suppression) sur un serveur abritant la zone (base de données en écriture)	**OUI**
Secondary zone (Zone secondaire)	C'est une zone créée à partir d'une zone primaire. L'objectif principal d'avoir une telle zone, est de pouvoir répondre aux clients se trouvant dans le même site que le serveur, dans un délai plus rapide que d'interroger la zone primaire dont elle dépend. En conséquence, définir plusieurs zones secondaires permet de répartir la charge et de réduire le nombre de requêtes envoyées directement au serveur abritant la zone primaire. Il est à noter qu'une zone secondaire peut être elle-même une zone primaire relativement à d'autres zones.	**NON**
Stub zone (*nouveauté depuis Windows 2003*)	Il s'agit d'une zone créée à partir d'une zone primaire avec la contrainte qu'elle contient uniquement : • Une copie de **SOA** (Start Of Authority) de la zone concernée • Une copie de tous les **NS** ayant autorité sur	**OUI**

	la zone • et tous les enregistrements de type **A** des serveurs ayant autorité sur la zone Son premier rôle est de permettre aux clients de pouvoir contacter les serveurs DNS ayant l'autorité pour la zone concernée dans la requête et pouvant y répondre, du coup elle accélère relativement les temps de réponses. Les contraintes imposées sur ce type de zone la rendent de taille négligeable comparativement à une zone secondaire. En conséquence, le processus de réplication d'une telle zone est très fluide et pris en charge par la réplication standard Active Directory. L'utilisation d'une telle zone s'avère particulièrement utile quand un client essaye de résoudre un nom DNS qui se trouve dans un réseau autre que celui de son entreprise. Dans ce cas, il suffit de créer une zone Stub dans le réseau de son entreprise contenant tous les enregistrements nécessaires pour que sa requête DNS puisse être transférée au bon serveur DNS de l'autre entreprise capable de résoudre le nom. En résumé, l'implémentation de zones Stub permet d'optimiser significativement les requêtes client destinées aussi bien vers des serveurs DNS au sein de l'entreprise que vers des serveurs DNS en-dehors du périmètre de celle-ci.	

- Le fait qu'une zone soit primaire ou secondaire dépend de sa position dans l'espace de nom DNS global et du serveur qui l'abrite. Sur un même serveur on peut avoir aussi bien des zones primaires que secondaires.
- Il est possible de convertir une zone d'un type à un autre comme indiqué dans la figure ci-dessous (à noter qu'une zone de type AD-Integrated est obligatoirement primaire et ne peut être secondaire par définition même, voir dans la suite) :

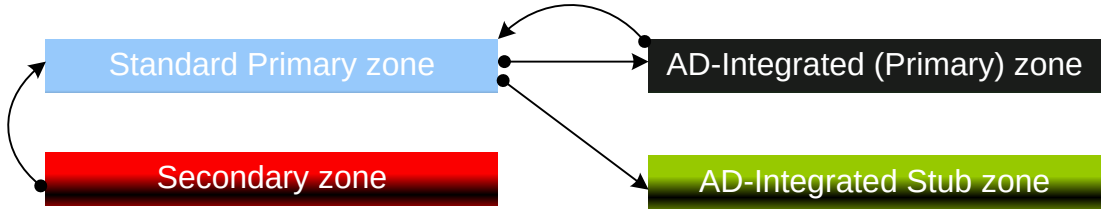

Figure 3 : Conversion de zones DNS

Types de zones en fonction de leurs rôles

Nous distinguons deux zones en fonction de leurs rôles vis-à-vis des requêtes émises par les clients :

Rôle de la zone	Explication
Forward lookup zone (Zone de recherché directe)	Comme son nom l'indique, la zone ayant ce rôle est utilisée pour résoudre les noms par des adresses IP et localiser les serveurs fournisseurs des services demandés : • Donnes-moi le nom de ton nœud et je vais chercher l'adresse IP correspondante ; • Donnes-moi le type de service recherché et je t'informe s'il est disponible et où. Côté client, celui-ci dispose d'un **Client Resolver** qui envoie le host name en tant que nom DNS en utilisant le cas échéant le **suffixe DNS principal** (voir la **Figure 5 : Suffixe principal d'un ordinateur** ci-dessous) au nom du host (flat name) ou les éléments de la **liste des suffixes DNS préférés** relativement à la connexion réseau utilisée (voir la **Figure 6 : Suffixe DNS propre à une connexion réseau** ci-dessous). A noter que le suffixe principal d'un client n'est pas nécessairement identique à celui spécifique à la connexion utilisée ; le premier est global alors que le deuxième est strictement lié à la connexion. A noter aussi même si au niveau du client, aucun suffixe principal n'a été spécifié avant sa jointure au domaine, il est automatiquement ajouté par le système une fois le client ait joint le domaine. Par contre, si un suffixe principal a été spécifié avant la jointure du domaine, il est automatiquement ajouté à son **nom NetBios**.
Reverse lookup zone (Zone de recherche	Comme son nom indique, le rôle principal de cette zone est de fournir aux clients les noms DNS à partir d'adresses IP : • Donnes-moi l'adresse IP et je te fourni le nom DNS

inversée) correspondant

Ceci est référé comme étant un enregistrement de type **PTR**.

La figure : Figure 4 : Illustration des zones et de quelques enregistrements montre les zones de recherche directes (Forward Lookup Zone), indirecte (Reverse Lookup Zone) ainsi que quelques enregistrements de base (SOA & NS).

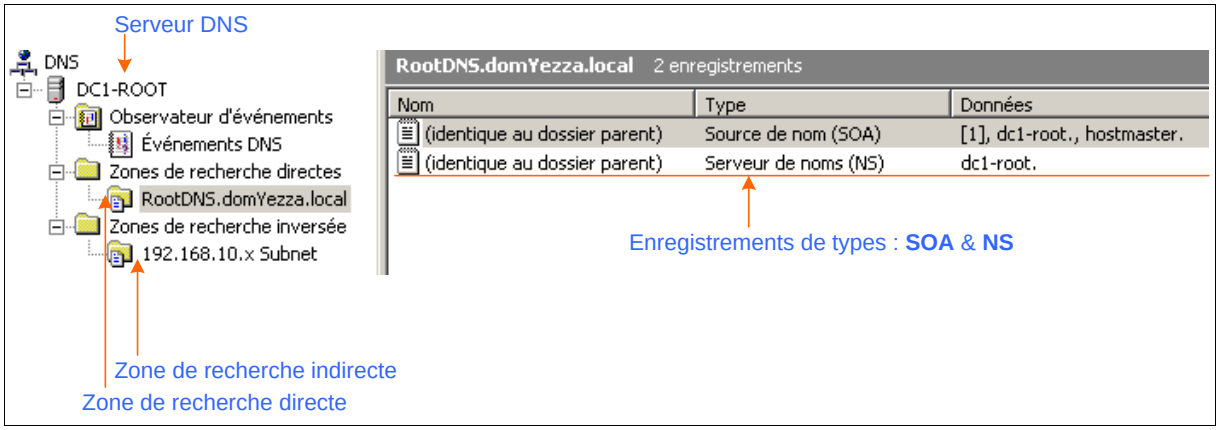

Figure 4 : Illustration des zones et de quelques enregistrements

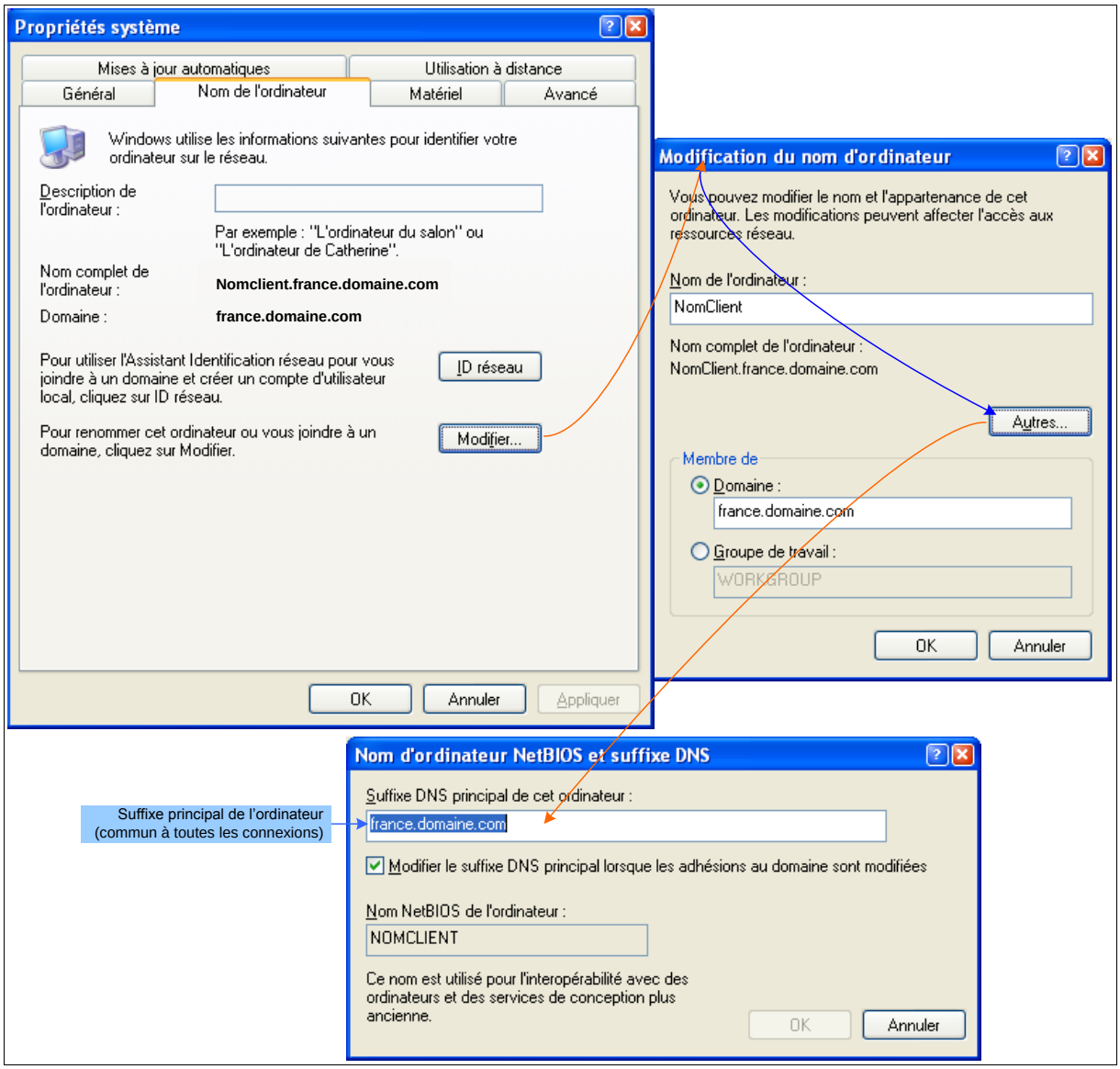

Figure 5 : Suffixe principal d'un ordinateur

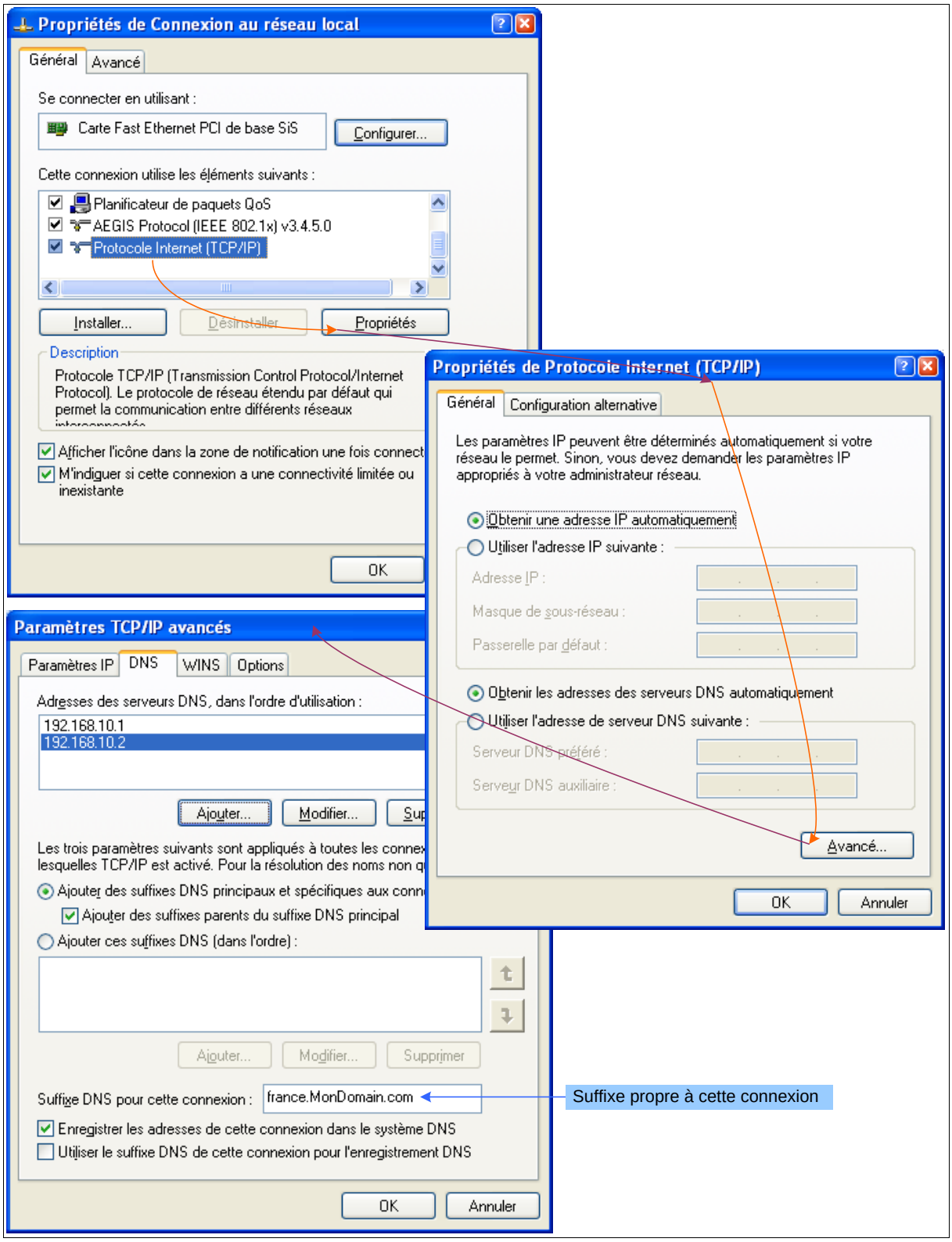

Figure 6 : Suffixe DNS propre à une connexion réseau

Eléments et Propriétés de base d'un DNS

Resource Record (RR)

Un **RR** (**Resource Record**) représente un enregistrement qui réalise la correspondance entre un type d'enregistrement DNS et un nom de domaine DNS. A noter que pour chaque type d'enregistrement, les champs utilisés ne sont pas nécessairement les mêmes. Avant de lister les types de RR les plus utilisés, voici un exemple d'un RR :

Champ :	Owner	Type	Class	TTL	Data
Valeur :	**host1.dom.com**	**A**	**IN**	**10**	**192.168.1.10**

Dans cet exemple, l'enregistrement est de type **A** et fait correspondre le nom DNS du host (**Owner**) : **host1.dom.com** à son adresse IP (**Data**) : **192.168.1.10** de **classe IN** (INTERNET) et avec un **TTL** (**Time To Leave**, voir la suite pour la définition) de : **10 secondes**. Le tableau suivant regroupe les types d'enregistrements DNS les plus utilisés :

Type	Utilisation
A	(Address). Ce type est utilisé pour établir une correspondance entre le nom DNS et une adresse IP (fournir l'adresse IP à partir du nom DNS). A noter que pour un nom canonique il pourrait y avoir plusieurs adresses IP (en fonction du réseau auquel le host est connecté) comme dans cet exemple : **Owner** **Type** **Class** **TTL** **Data** host1.dom.com A IN 10 192.168.1.10 host1.dom1.com A IN 20 10.10.1.1 host1.dom3.com A IN 30 10.11.1.1
PTR	Comme son nom l'indique (PoinTeR), il s'agit de l'opposé de A, autrement dit, il fait correspondre une adresse IP à un nom DNS.
CNAME	(Canonical NAME). Fait correspondre un **alias** du nom canonique et l'adresse IP. L'intérêt d'utiliser ce type d'enregistrement DNS est de faire correspondre des noms alternatifs qui correspondent aux différents services fournis par la même machine. Par exemple un serveur de distribution de packages ayant un nom universel au niveau d'une entreprise (nom unique), ce qu'on appelle un **alias**, mais possède plusieurs adresses IP en fonction de la proximité du client (se trouvant dans le même site que les serveurs de distribution), ce qui constitue le contexte d'une manière générale.
NS	(Name Server). Ce type d'enregistrement est utilisé pour indiquer un serveur NS (voir la définition ci-dessous). Ce type d'enregistrement par exemple doit se trouver au moins une fois dans un serveur abritant une **Stub zone** par exemple.
SOA	(Start Of Authority). Ce type est utilisé pour indiquer l'enregistrement unique représentant le SOA (voir ci-dessous pour la définition). Finalement, le type SOA est

	un NS particulier.
MX	(Mail eXchange). Comme son nom l'indique ce type d'enregistrement sert à faire correspondre un serveur de messagerie. Concrètement, lorsqu'un utilisateur envoie un message (courriel) vers un destinateur comme : dest@DomDest (où **DomDest=mailcorp.com** par exemple), le serveur Exchange auquel le compte de l'émetteur est rattaché, interroge le SOA pour obtenir la donnée correspondante à un enregistrement de type MX dont le champ **Owner** correspond à **DomDest** ayant le cas échéant (dans le cas de répartition de charge entre plusieurs serveurs Exchange) la priorité (**Priority**) la moins élevée et le poids (**Weight**) le plus important. En fait, la requête se fait sur deux étapes, établir d'abord l'enregistrement de type **MX** correspondant, puis l'enregistrement de type **A** qui correspond à la donnée du 1er enregistrement. Voici un exemple :

Owner	Class	Type	Priority	Data
mailcorp.com	IN	MX	10	exsrv1.mailcorp.com
mailcorp.com	IN	MX	20	exsrv2.mailcorp.com
mailcorp.com	IN	MX	30	exsrv3.mailcorp.com
exsrv1.mailcorp.com	IN	A		10.20.30.10

Figure 7 : Résolution d'enregistrements de type MX

HINFO	(Host INFO). Ce type d'enregistrement, très peu utilisé et déconseillé, sert à contenir des informations matérielles caractérisant le host comme le CPU, l'OS etc.

Les enregistrements peuvent être ajoutés ou modifiés manuellement ou automatiquement (modification des fichiers constituant les bases DNS, par le biais des consoles ou par scripts ou programmes). Des services système peuvent aussi modifier les enregistrements comme par exemple le **service Netlogon** sur un DC modifie les enregistrements SRV (Service location) au démarrage de la machine pour ajouter celui qui correspond au DC dans le fichier :

```
%windir%\system32\config\netlogon.dns
```

Ce type d'enregistrement permet aux clients ou aux applications dans un domaine de découvrir les DC de ce dernier. Le même mécanisme est valable pour les services comme le **Global Catalog**, **Kerberos**, **LDAP** etc.

Name Server (NS)

Le **NS** (**Name Server**) est un serveur DNS abritant au moins une zone DNS. A noter qu'un NS peut bien être autoritaire pour plus d'une zone à la fois. Dans une telle situation, il est tout-à-fait normal et même conseillé que le NS délègue une partie des zones à d'autres NS afin de répartir la charge induite par les requêtes reçues de la part des clients.

Start Of Authority (SOA)

Le **SOA** (**Start Of Authority**) comme son l'indique, il représente le NS possédant l'autorité pour une zone DNS. Pour chaque zone DNS il doit exister un et un seul NS possédant l'autorité sur celle-ci, autrement un risque de collision est certain quand un serveur DNS essaye de répondre aux requêtes Client. Le SOA constitue la 1ère source d'information à propos de la zone dont il est propriétaire. Il contient également d'autres informations sur la zone comme son état, ses caractéristiques etc.

La figure ci-dessous illustre les propriétés du **SOA** du serveur : **dc1-root.domyezza.com** ayant l'autorité sur la zone : **domyezza.com**. Voir ci-dessous la signification des propriétés indiquées sur cette figure.

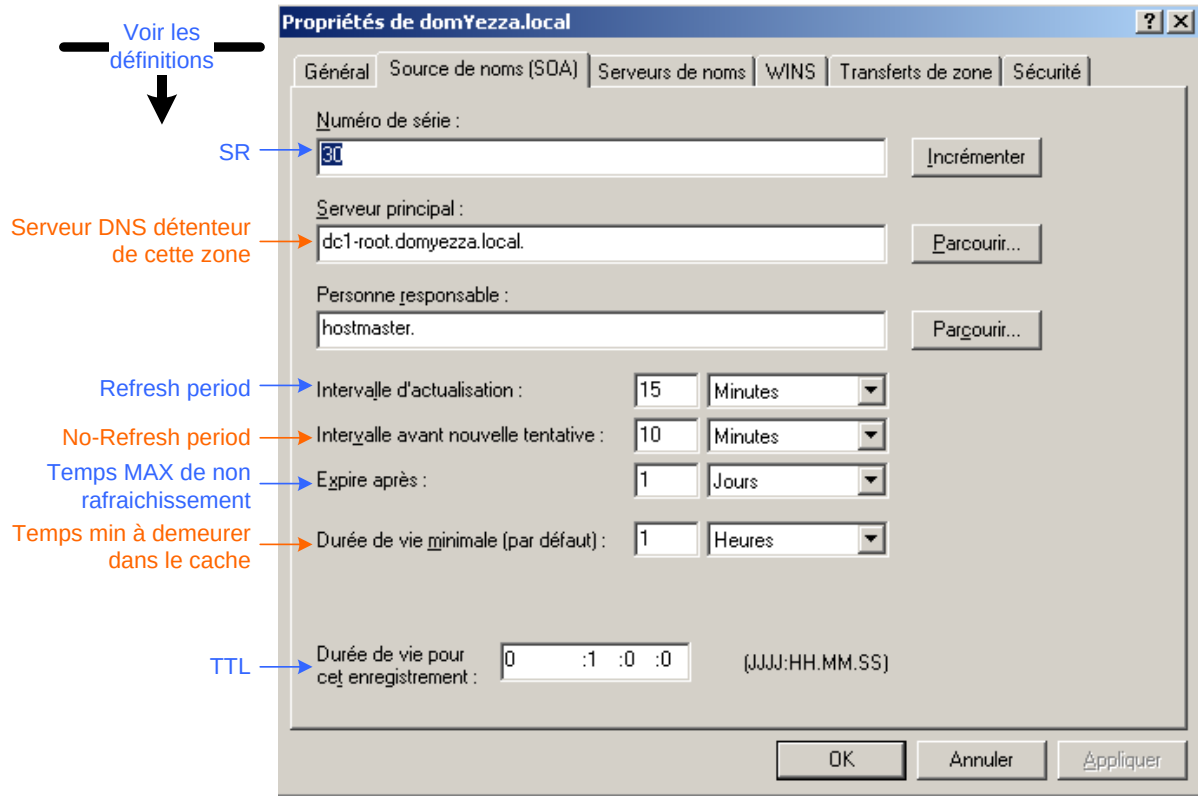

Figure 8 : Les propriétés du SOA relativement à la zone domyezza.com

TTL (Time To Leave)

Pour une zone, le **TTL** représente le temps maximum (en secondes) pour un **NS** de garder dans son cache tous les **RR**s avant des les purger (les rendre invalides).

Pour un **RR**, le **TTL** représente le temps maximum pour un **NS** de le garder dans son cache avant de le supprimer. Par conséquent, le **TTL** relatif à chaque **RR** doit être inférieur à celui de la zone à laquelle il appartient.

Par ailleurs, le TTL relatif à une zone est prioritaire par rapport à l'ensemble des TTLs relatifs aux RRs. La valeur par défaut du TTL d'une zone sur un serveur DNS en Windows 2003 est fixée à 1 heure (3600 secondes).

SN (Serial Number)

Le **SN** d'une zone est rattaché directement au **SOA**. Son rôle est de maintenir la zone à jour par rapport aux zones dont elle dépend. Le SN reflète toutes les modifications qui ont eu lieu depuis la dernière fois la zone à été mise à jour. Il est principalement utilisé au niveau d'une zone secondaire en comparant son SN avec celui de la zone primaire dont elle dépend. La différence entre le SN d'une zone secondaire et celui d'une zone primaire est que le 1er est mis à jour à partir de la zone primaire et le 2ème est mis à jour chaque fois que celle-ci est mise à jour à partir de la zone primaire dont elle dépend. On peut voir le SN comme étant la version d'une zone. Par ailleurs, chaque attribut possède un numéro de série que l'on appelle l'**USN** (Update Serial Number) qui impacte directement le SN de la zone.

Root hints

Dans un NS, les **Root hints** représentent la liste des serveurs DNS situés en haut de la chaîne induite par les requêtes des clients soit dans un réseau d'entreprise ou dans le réseau public (Internet). Pour ce dernier, justement la liste des root hints compte aujourd'hui au moins 13 serveurs pour toute la communauté du réseau Internet. Ces serveurs sont automatiquement ajoutés dans un serveur DNS basé sur Windows 2003 par exemple et la liste peut être mise à jour directement en utilisant une connexion Internet et par exemple le protocole ftp pour télécharger la liste des root hints **named.dns** Internet à partir du site ftp :

```
ftp://ftp.rs.internic.net/domain
```

La liste des root hints en zone public est maintenue par l'organisation **InterNIC** (Internet Network Information Centre). Physiquement, sur un serveur DNS Windows, la liste des root hints est contenue dans le fichier :

```
%windir%\system32\dns\cache.dns
```

Par conséquent, ce fichier doit être absolument sécurisé.

Quant à la liste des root hints dans un réseau d'entreprise, doit être aussi bien sécurisée et généralement ne pointe pas vers les serveurs DNS de la zone public (Internet) sauf si un besoin de résolution DNS Internet est justifié.

Forwarders (Redirecteurs)

Les **Forwarders** représentent des serveurs DNS auxquels toutes requêtes DNS non résolues sont transférés par le serveur DNS courant. Ainsi, si un serveur DNS fait partie des **root hints,** ne peut avoir des **Forwarders**, car il est situé au sommet des NS dans l'espace de nom des serveurs DNS.

Un serveur DNS peut bien être configuré en tant que **Forwarder only**, autrement dit, leur rôle est limité uniquement au transfert des requêtes à une liste de serveurs DNS capable d'y répondre. D'une manière générale, un serveur de ce type, est **multi-homed**, i.e., d'un côté il est connecté au réseau de l'entreprise, et de l'autre côté est connecté à un autre réseau comme Internet par exemple.

Il est à noter que les **Forwarders** sont utilisés par zone, i.e., chaque zone possède une liste de **Forwarders**, tandis que les **root hints** sont utilisés par serveur DNS (donc pour toutes zones abritées par le serveur). Ainsi, le serveur DNS utilise la liste des **Forwarders** relatifs à la zone

avant d'utiliser les **root hints** dans le cas de l'impossibilité de résolution de nom DNS (**du bas vers le haut**).

Transferts de Zones

On parle du transfert de zone pour désigner l'opération qui consiste habituellement à transférer des données d'une zone principale (primaire) vers une zone secondaire. On peut désigner la liste des serveurs auxquels on souhaite transférer les mises à jour de la zone comme les NSs en particulier. D'une manière générale, on permet le transfert uniquement vers des NSs sélectionnés afin de répartir la charge induite par le nombre de requêtes envoyées par les clients.

Il existe deux méthodes pour transférer les données d'une zone vers un autre serveur :

- Un transfert de zone complet (**AXFR**) : On transfert toutes les données de la zone primaire vers la zone secondaire, généralement à une fréquence cyclique.
- Un transfert incrémental (**IXFR**) : On transfert uniquement les données qui ont changé depuis le dernier transfert. Le critère de comparaison utilisé se base sur le **numéro de série (SR) du SOA** des deux zones concernées par le transfert. Le mécanisme de déclenchement utilise la **notification** envoyée par le serveur abritant la zone primaire vers celui abritant la zone secondaire. Il est à noter que même dans le cas d'une configuration de transfert incrémental, il peut arriver que le transfert soit quand-même complet comme dans le cas par exemple où :
    - la volumétrie des données modifiées est trop importante (dépasse le volume des données de la zone secondaire)
    - la période de rafraichissement des données (période pendant laquelle les données peuvent être mises à jour) est dépassée. Sur Windows 2003, cette période est fixée par défaut à 24 heures.

  Voir plus bas pour les détails concernant les périodes de rafraichissement et du non rafraichissement ainsi que le mécanisme utilisé.

Resolver

Enfin il est temps d'en parler, il représente la partie cliente (service DNS client) du DNS responsable de la résolution des requêtes DNS initiées par le système et les applications. Au démarrage d'un poste client, le système constitue son cache à partir du fichier :

```
%windir%\system32\drivers\etc\hosts
```

et le cache courant contenant les réponses des requêtes déjà résolues et dont le TTL n'a pas encore était atteint. Afin de voir le contenu du cache DNS, on peut faire appel à la commande suivante :

```
Ipconfig /displayDNS
```

Sur une station Windows (Serveur ou Workstation), le rôle du **Resolver** est attribué au service **Client DNS** faisant partie des services génériques Windows générés par le process **svchost** et la DLL associée : **dnsapi.dll** et utilisant le compte **Service réseau**. Pour plus de détails, vous pouvez consulter par exemple l'article suivant : kb314056.

Aging/Scanvenging

Le **Aging** (vieillissement) est relié à la durée de vie des enregistrements DNS, plus précisément ceux du type **A** ou **PTR** qui ont été ajoutés ou mis à jour dynamiquement (à partir d'un serveur DHCP ou un Client DHCP). Il est à noter que les enregistrements qui ont été ajoutés manuellement ne sont pas concernés par cette fonctionnalité. La raison provient du fait que les enregistrements ajoutés manuellement ont un Timestamp fixé à 0, alors que les autres ont un timestamp correspondant à celui du serveur DNS (voir ci-dessous pour plus de détails).

Le rôle principal du **Aging** est d'éviter des réplications répétitives et fréquentes, ce qui augmente le volume du trafic induit par la réplication suite à une modification des zones DNS.

Le mécanisme de **Aging/Scanvenging** se base principalement sur deux éléments :

1. Période de non-rafraichissement (**No-Refresh time**) : Durant cette période, aucune mise à jour n'est permise.
2. Période de rafraichissement (**Refresh time**) : Durant cette période les rafraichissements sont permis. Pour les enregistrements RR, ils sont automatiquement supprimés si pendant toute cette période aucune demande de mise à jour n'a été reçue de la part des clients DHCP ou par le biais de mise à jour dynamique du serveur DHCP. Cette dernière opération correspond au **Scanvenging**.

D'une manière générale, ces deux périodes sont fixées à **7 jours**, qui représentent la moitié de la période de bail d'un scope DHCP sur un serveur DHCP de **14 jours**.

Le principe du **Aging/Scanvenging** peut être configuré au niveau du serveur ou au niveau de chaque zone. Toutefois, la zone est prioritaire par rapport au serveur.

La figure ci-dessous illustre le principe de fonctionnement :

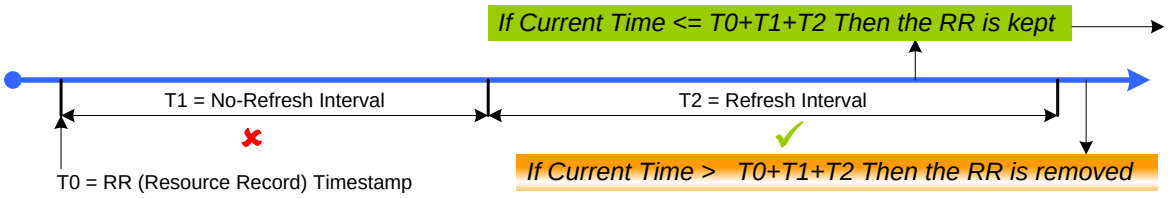

Figure 9 : Principe du Aging/Scanvengin

Comme on a vu précédemment, le DNS interagit avec plusieurs services, le **Netlogon**, le **serveur DHCP**, le **Client DHCP** et aussi le service de **Cluster** qui pourrait être utilisé pour répartir la charge des serveurs DHCP par exemple. Le tableau ci-dessous indique le temps maximum pour chaque service avant de rafraichir des enregistrements qui sont sous sa responsabilité :

Service	Période Max de non rafraichissement
Netlogon	24 heures
Serveur DHCP	4 jours (½ du Bail)
Client DHCP	24 heures. A noter que le service DHCP Client, s'il est configuré, peut aussi envoyer les mises à jour des Clients configurés à avoir des adresses IP

	statiques !
Cluster	24 heures

Outils DNS (Windows)

La liste suivante regroupe les outils pouvant être utilisés pour diagnostiquer les problèmes reliés au DNS et effectuer aussi bien le Troubleshooting que la manipulation et la configuration des serveurs DNS (Windows 2003 et plus) et les Clients DNS (Windows XP et plus) :

Outil	Où se trouve-t-il ?	Utilisation
Console DNS	• Fait partie des outils d'administration : **Dnsmgmt.msc** • Peut être disponible sur une station Windows XP et plus en installant le package **AdminPack** à télécharger sur le site de Microsoft	La console est utilisée principalement pour configurer les serveurs DNS local ou distants et effectuer les opérations quotidiennes (ajout/modification/suppression/délégation etc.) et non pas pour le diagnostic.
Nslookup	• Le dossier **System32** • Présent aussi bien sur le serveur que la station de travail	Cet outil est utilisé pour résoudre les noms et localiser les détenteurs de services. Il fonctionne de la même façon que le **Resolver** natif à l'exception qu'il ne prend en compte dans le processus de résolution que le 1er serveur DNS indiqué dans la liste des serveurs DNS préférés relatifs à la configuration IP.
DnsCmd	**Support Tools :** Compris dans le media d'installation du l'OS du serveur	Cet outil est un utilitaire en ligne de commande à utiliser pour configurer les serveurs DNS, automatiser l'installation de serveurs DNS. Il traite aussi les zones, les partitions, les enregistrements, le cache DNS etc.
dnsLint	Support Tools	Utilisé pour vérifier la prise en compte des enregistrements par les DCs et vérifier également les problèmes de délégation.
NetDiag	Support Tools	Cet outil est utilisé pour diagnostiquer des problèmes de connectivité y compris le DNS. Voici un exemple pour tester le DNS :

		`NetDiag /test:DNS`
DCDiag	Support Tools	Cet outil est utilisé pour diagnostiquer des problèmes reliés à la réplication, à DCPromo (Promotion d'un serveur vers un DC), le FSMO etc. Il peut être utilisé en particulier pour diagnostiquer le DNS : `DCDiag /test:DNS`
IPConfig	• Le dossier **System32** • Présent aussi bien sur le serveur que la station de travail	Cet outil est utilisé pour configurer l'adressage IP d'un client, gérer le client DHCP (**renew/release**), enregistrer une adresse IP, gérer le cache local DNS, enregistrer des données pour des classes DHCP personnalisées définies par l'utilisateur ou un vendeur d'une solution utilisant le serveur DHCP.
netsh	• Le dossier **System32**	Peut être utilisé aussi bien côté client pour configurer la connexion IP y compris le DNS que côté serveur pour configurer le DNS, le DHCP etc. Voici un exemple :
Exemple : Configurer l'interface appelée **Local Area Connection** en fixant une adresse IP statique : **192.168.10.132** avec le masque : **255.255.255.128**, la passerelle : **192.168.10.254** et une métrique de **1** `netsh interface ip set address name="Local Area Connection" source=static addr=192.168.10.130 mask=255.255.255.128 gateway=192.168.10.254 gwmetric=1`		
Monitoring & journalisation	• Utilisation du moniteur de performances • Les différentes méthodes de journalisation fournies par la console DNS • Le journal d'évènements relatif au DNS	A l'exception du moniteur de performances plus robuste, les autres méthodes sont accessibles directement dans la console. Le moniteur système offre plus de possibilités et d'éventails en termes de choix des objets et des compteurs pouvant être rajoutés.
Programmation	• Scripting • Langages avancés	Voir la section suivante.

DNS : Automatisation

Afin d'automatiser des opérations répétitives ou de déploiement relatives au DNS, le moyen le plus rapide et le plus flexible, est d'utiliser des scripts codés en VBScript, JavaScript (JScript), Perl ou autres langages de Scripting moins répandus. L'automatisation constitue un moyen rapide pour traiter un nombre important de serveurs ou postes de travail et évite des interventions de support ainsi que des erreurs de manipulation humaines.

Nous nous limitons ici à l'utilisation de VBScript en utilisant **WMI**, plus précisément le **CIM** (Common Information Model) V2 et plus. Ce dernier expose quelques classes DNS pouvant être utilisées dans le but d'automatiser plusieurs opérations sur le DNS. Toutes ces classes font partie du Namespace (espace de classes) DNS **MicrosoftDNS** qui regroupe les classes suivantes les plus utilisées du côté serveur et auto-explicatives :

- **MicrosoftDNS_Server**
- **MicrosoftDNS_Statistic**
- **MicrosoftDNS_Zone**
- **MicrosoftDNS_ResourceRecord**
- **MicrosoftDNSRootHints**

Du côté client, les classes suivantes peuvent être utilisées pour entre autres traiter la configuration DNS Client :

- **Win32_ComputerSystem**
- **Win32_OperatingSystem**
- **Win32_NetworkAdapterConfiguration**
- **Win32_NTDomain**

Afin d'avoir plus de détails sur ces classes et d'autres, leurs propriétés et méthodes, je vous invite à consulter MSDN et les articles Technet de Microsoft sur le sujet. Une autre méthode pour automatiser le traitement de DNS est d'utiliser le **SDK** (Software Developement Kit) disponible en fonction de la plate-forme cible. Le SDK est généralement utilisé pour faire appel aux API (Application Programing Interface) Windows dans des langages de programmation comme C/C++/C#/VB/Delphi/Borland C++ etc. Nous ne traitons pas cette partie dans cet article, car il s'agit d'un sujet en-dehors du contexte. Pour finir, voici 4 exemples d'utilisation de VBScript pour automatiser quelques opérations relatives au DNS serveur ou client. Les trois 1ers exemples sont destinés uniquement aux serveurs DNS, tandis-que le dernier traite la partie configuration du client DNS.

Exemple 1: Obtenir des statistiques d'un serveur DNS

```
'   But :       Obtenir et afficher des stats d'un serveur DNS
'   strServer : FQDN d'un serveur DNS
sub GetDNSStatistics(strServer)
    dim objDNS, objDNSServer, objStats, objStat
    '   objet définissant une instance de la classe MicrosoftDNS
    set objDNS = GetObject("winMgmts:\\" & strServer & "\root\MicrosoftDNS")
    '   objet définissant le serveur DNS local
    set objDNSServer = objDNS.Get("MicrosoftDNS_Server.Name="".""")
    '   objet exécutant la requête sur la classe MicrosoftDNS_Statistic
    set objStats = objDNS.ExecQuery("Select * from MicrosoftDNS_Statistic")
    '   Afficher les résultats de la requête
    for each objStat in objStats
       WScript.Echo " " & objStat.Name & " : " & objStat.Value
```

```
    Next
end sub

GetDNSStatistics "MyDNSServer.MyDomain.com"
```

Enregistrer ce fichier sous le nom **Stats.vbs**. Pour le tester, lancer en ligne de commande par exemple :

```
CScript Stats.vbs > c:\out.txt
```

Toutes les statistiques sont incluses dans le fichier de sortie **out.txt**.

Exemple 2: Obtenir des enregistrements de type SRV d'un serveur DNS

```
option explicit

Dim objDNS, objRRs, objRR
set objDNS = GetObject("winMgmts:root\MicrosoftDNS")

sub GetGlobalCatalogs(strDomain)
    '   Get the Global Catalogs records
    set objRRs = objDNS.ExecQuery("Select * from MicrosoftDNS_SRVType " & _
                                  " Where OwnerName = '_ldap._tcp.gc._msdcs." & _
                                  strDomain & "'")
    WScript.Echo "Global Catalogs for " & strDomain
    for Each objRR in objRRs
       Wscript.Echo " " & objRR.DomainName
    next
end sub

sub GetDomainControllers(strDomain)
    Wscript.Echo
    '   Get the Domain Controllers records
    set objRRs = objDNS.ExecQuery("Select * from MicrosoftDNS_SRVType " & _
                                  " Where OwnerName = '_ldap._tcp.dc._msdcs." & _
                                  strDomain & "'")
    WScript.Echo "Domain Controllers for " & strDomain
    for Each objRR in objRRs
       Wscript.Echo " " & objRR.DomainName
    next
end sub

sub GetPDC(strDomain)
    Wscript.Echo
    '   Get the PDC holders records
    set objRRs = objDNS.ExecQuery("Select * from MicrosoftDNS_SRVType " & _
                                  " Where OwnerName = '_ldap._tcp.pdc._msdcs." & _
                                  strDomain & "'")
    WScript.Echo "PDC holders for " & strDomain
    for Each objRR in objRRs
       Wscript.Echo " " & objRR.DomainName
    next
end sub

Dim strDomain
strDomain="MyDomain.com"
GetGlobalCatalogs strDomain
GetDomainControllers strDomain
GetPDC strDomain
```

Enregistrer ce fichier sous le nom **GetSRV.vbs**. Pour le tester, lancer en ligne de commande par exemple :

```
CScript GetSRV.vbs > c:\out.txt
```

En sortie on a les serveurs **GC**, les **DCs** et le **PDC** du domaine **MyDomain.com** dans le fichier de sortie **out.txt**.

Exemple 3 : Utilisation d'un compte priviligié

Comment utiliser un autre compte priviligié pour créer des RRs sur un serveur DNS distant. Assurez-vous que le serveur DNS distant ne représente pas le serveur local, sinon vous aurez certainement une erreur. Le même script peut être adapté pour traiter les enregistrements PTR (voir la note qui suit).

```
'//////////////////////////////////////////////////////////////
'   Purpose :   Create the RR (Resource Record) of tye A
'               using alternate credentials
'//////////////////////////////////////////////////////////////

option explicit

Call Main()

sub Main()
    Dim strComputer, strUserName , strPassword
    Dim strRR, strReverseRR, strDomain, strReverseDomain
    '   define variables
    strComputer = "dc2-root.domyezza.local"    'Remote DNS Server
    strUserName = "administrator@domyezza.local"    'Privileged User
    strPassword = "<Type administrator password>"
     ' A record to add to the DNS Server
    strRR = "NewRR.domyezza.local. IN A 192.168.10.20"
    strDomain = "domyezza.local"

    AddRR strComputer, strUserName , strPassword, strRR, strDomain
end sub


sub AddRR(strComputer, strUserName , strPassword, strRR, strDomain)
    '   define variables

    Dim objLocator, objDNS, objDNSServer, objOutParam, objRR, objRR2

    '   object used to connect to the remote server repository cimv2
    '   using alernate credentials
    set objLocator = CreateObject("WbemScripting.SWbemLocator")
    set objDNS = objLocator.ConnectServer(strComputer, _ "root\MicrosoftDNS", strUserName,
strPassword)

    set objDNSServer = objDNS.Get("MicrosoftDNS_Server.Name="".""")

    '   create RR object to be used for adding strRR defined above
    set objRR = objDNS.Get("MicrosoftDNS_ResourceRecord")

    ' Create the A record
    Dim strNull
    strNull = objRR.CreateInstanceFromTextRepresentation( _
                      objDNSServer.Name, _
                      strDomain, _
                      strRR, _
                      objOutParam)

    set objRR2 = objDNS.Get(objOutParam)
    WScript.Echo "Created Record: " & objRR2.TextRepresentation

    '   Cleanup objects
    set objOutParam = Nothing
    set objDNSServer= Nothing
    set objRR       = Nothing
    set objRR2      = nothing
end sub
```

Note

Pour utiliser le même script que le précédent afin d'ajouter un enregistrement correspondant de type **PTR** dans la zone de recherche indirecte (**Reverse Lookup Zone) :** 10.168.192.in-

addr.arap, il suffit juste de modifier l'expression **strRR** et le domaine DNS **strDomain** et fixer leur valeur à :

```
strRR = "20.10.168.192.in-addr.arap IN PTR newRR.domyezza.local."
strDomain = "10.168.192.in-addr.arap."
```

Il est à noter que le **ID réseau** est écrit à l'envers **10.168.192**, et que la 1ère partie **20** représente le **host ID**.

Afin de tester qu'effectivement l'enregistrement a bien été ajouté, il suffit par exemple d'utiliser l'outil **Nslookup** comme suit :

```
Nslookup -type=PTR 192.168.10.20
```

Vous devez avoir en sortie le serveur DNS qui a été interrogé suivi de la résolution de l'adresse IP 192.168.10.20 :

```
Server:  dc1-root.domyezza.com
Address:  192.168.10.1

20.10.168.192.in-addr.arpa      name = newRR.domyezza.com
```

Exemple 4 : Modification du domaine d'une connexion réseau

Le script suivant modifie le domaine d'une connexion réseau de l'ordinateur local, autrement dit, il modifie le « **Suffixe DNS pour cette Connexion** » pour une connexion réseau.

Usage : `CScript ChangeDNSDomain.vbs <Nom_Connexion> <Nom_Domaine>`

```
option explicit

call Main()

'//////////////////////
' Main procedure
'//////////////////////
sub Main
    dim oArgs
    set oArgs=wscript.Arguments
    if oArgs.count<2 then
        set oArgs=Nothing
        Usage
        QuitScript(1)
    end if

    if ChangeDNSDomain(oArgs(0), oArgs(1))<>True then
        QuitScript(2)
    end if
end sub

Sub QuitScript(Code)
    wscript.quit(Code)
End Sub

Sub Usage()
    wscript.Echo "Usage: cscript|wscript SetDomain.vbs <Connexion name> <Domain>" & VBCRLF
```

```
End Sub

'////////////////////////////////////////////////
' Purpose : Change Domain of a named NIC adapter
'////////////////////////////////////////////////
function ChangeDNSDomain(ConnexionName, strDomain)
    Dim strComputer, strMACAddress, intDomain
    strComputer="."

    Dim objWMIService, colNicConfigs, objNicConfig, objItem, colItems

    ' Connect to the WMI service.
    Set objWMIService = GetObject("winmgmts:" _
         & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")

    ' Get Corresponding NIC MAC address
    Set colItems = objWMIService.ExecQuery _
    ("Select * From Win32_NetworkAdapter " _
        & "Where NetConnectionID = '" & ConnexionName & "'")

    For Each objItem in colItems
        strMACAddress = objItem.MACAddress
    Next

    ' Get NIC adapters collection
    Set colNicConfigs = objWMIService.ExecQuery("SELECT " & _
     "* FROM Win32_NetworkAdapterConfiguration WHERE IPEnabled=True")

    ' Change connection DNS domain
    For Each objNicConfig In colNicConfigs
        wscript.Echo objNicConfig.Description
        wscript.Echo objNicConfig.MACAddress
        If objNicConfig.MACAddress = strMACAddress then
           intDomain = objNicConfig.SetDNSDomain(strDomain)
           if intDomain = 0 then
                ChangeDNSDomain = True
           else
                ChangeDNSDomain = False
           end if
           Exit For
        End If
    Next

    set objWMIService = Nothing
    set colNicConfigs = Nothing
End function
```

Enregistrer ce fichier sous le nom **ChangeDNSDomain.vbs**. Pour le tester, lancer en ligne de commande par exemple :

```
CScript ChangeDNSDomain.vbs <Nom_Connexion> <Nom_Domaine>
```

Comme par exemple :

```
CScript ChangeDNSDomain.vbs "Connexion au réseau local"  MyDomain.local
```

Pour vérifier, il suffit de (d') (Procéder comme dans : Figure 6 : Suffixe DNS propre à une connexion réseau) :

❶ Ouvrir la fenêtre des propriétés de la connexion que vous avez passée en paramètre

❷ Sélectionner l'entrée **Protocole Internet TCP/IP**

❸ Cliquer sur **Propriété**

❹ Cliquer sur **Avancé…**

❺ Se positionner sur l'onglet DNS et vérifier que l'entrée « Suffixe DNS pour cette Connexion » a bien été modifiée correctement

Note

La modification du suffixe d'une connexion réseau est équivalente à la modification de la donnée correspondante à la clé de registre suivante :

Clé : `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\ID`
Valeur : `Domain`
Type de donnée : chaîne de caractère (`REG_SZ`)

Où `ID` correspond à l'identificateur de l'interface réseau, ce qui est représentée par la propriété `SettingID` de l'objet `objNicConfig` référencé dans le code précédent, i.e., `objNicConfig.SettingID`. Voir la figure ci-dessous : Figure 10 : Propriétés d'une connexion réseau dans la BDR) faisant référence à la donnée : `Domain` avec la valeur : `yourdomain.com`.

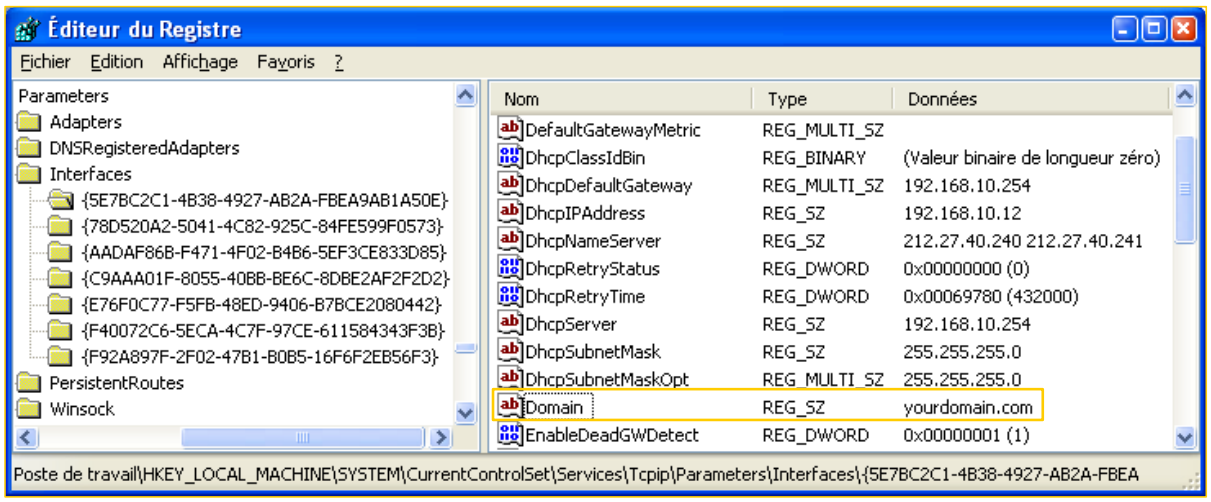

Figure 10 : Propriétés d'une connexion réseau dans la BDR

Je vous déconseille fortement d'utiliser la BDR (Base De Registre) pour effectuer ce genre de modification. Il est plus sûr d'utiliser des outils plus appropriés comme WMI dans des langages de Scripting ou dans des langages plus évolués comme C++/C#/VB/Delphi etc. voire utiliser direment les API Windows si vous maîtrisez ce genre de programmation. Dans tous les cas, on a toujours un code retour et éventuellement une description plus appropriée à l'erreur rencontrée. D'une manière générale, utiliser la BDR est non seulement risqué, mais aussi le passage d'une version d'OS à une autre ne garantie pas que les noms des clés seront conservés.

Conclusion

Le présent document ne constitue en aucun une référence, des documents électroniques ou physiques, plus élaborés et plus détaillés sur le sujet peuvent être consultés. J'ai plutôt essayé dans ce document de mettre la lumière sur l'essentiel du DNS et m'adresser à des lecteurs ayant déjà une base minimale sur le sujet. Le site de Microsoft, notamment la partie MSDN, constitue une excellente référence pour le DNS aussi bien côté serveur que côté client. Plusieurs sujets adhérents au DNS n'ont pas été évoqués dans ce document comme le design et l'infrastructure DNS, les autres implémentations du DNS que celle de Microsoft (Unix, Linux, Novell Netware etc.), le déploiement, la maintenance etc.