

Metro Hartford Information Services

Network Implementation, Operation, and Maintenance Standards

Title:	DIRECTORY SERVICE DESIGN & IMPLEMENTATION	STD#0002
Author:	Stephen Shipman	8/21/02
Reviewer(s):	Chris Collins	
Revision History:		Released

1.0 Scope & Notes

1.1 Scope

This document details the design and implementation standards for MHIS's Microsoft Active Directory (MS-AD) environment. Where appropriate, the design principles and operational practices that shaped the HPS Novell Directory Services (NDS) implementation have been brought forward into this new directory architecture.

2.0 Directory Architecture

2.1 Model

2.1.1 Geographic vs. Functional Models

There are two general models for organizing the Organizational Units (OUs) within an X.500-type directory: functional and geographic. The MHIS MS-AD tree structure is organized based on functional, not geographic divisions. For example: all elementary schools will exist in an Elementary container, all middle schools in a Middle container, all high schools in a High_Schools container. On the municipal side the Fire Department will exist in a Fire container, Police Department in the Police container, etc.

2.1.2 Direction in Implementing the Model

The HPS NDS structure had a mixed model which tended toward functional divisions: the combined directory architecture will be based more consistently on functional divisions.

2.2 Structure

2.2.1 Root Structure Outline

The forest name for MHIS's MS-AD implementation is *Hartford.metro*. Within the forest are two Domains: *Hartfordschools.org* and *Hartford.gov*. These two Domains contain Organizational Units which represent the functional divisions of the City of Hartford and the

Hartford Public Schools. Some of the Organizational Units within the structure are outlined in 2.1.1, and in Figure 1 below. This structure is expandable: it can accommodate additional organizations such as the Hartford Public Library without renaming of existing domains.

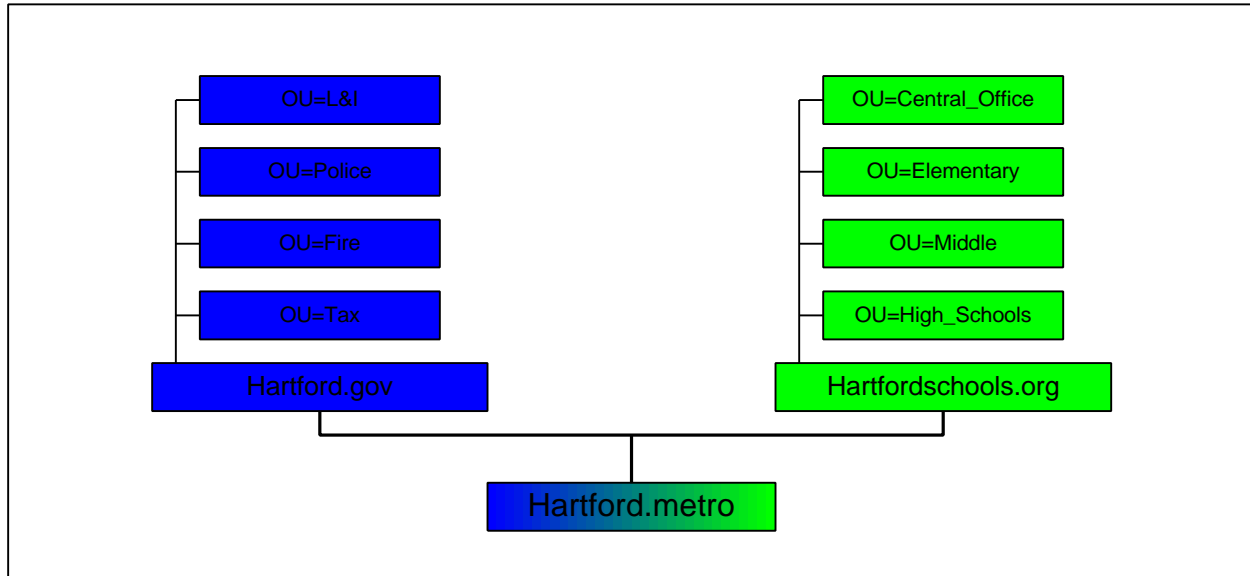


Figure 1 MHIS MSDS Tree and Forrest Structure

2.2.2 Organizational Structure Below the Domain Level

At the sub-domain level we will establish Organizational Units to divide users and groups up among their functional areas. We will implement one level of functional division. For example, the Hartford Fire Department will be established with an OU (Fire). There will be no further subdivision below the Fire OU. For example, there will be no *Company4* or *Training* OUs. The rule here is *one director, one OU*. For each top level city department and quasi-governmental organization there shall be one OU.

2.2.3 MHIS and Similar Entities

The container for MHIS, as an organization with roots on both side of the BoE / Municipality division will be in Hartford.gov. The default location for entities that are not clearly school system only organizations shall be Hartford.gov

2.2.4 Discussion

There is a concept in programming which is helpful here: when dealing with instances of objects a system should allow no instances of a thing, exactly one, or an infinite number of that thing. The core idea is that arbitrary limitations on the number of instances of a *thing* are not helpful and should be avoided. This idea is applied here in the directory structure where 1) we do not arbitrarily limit the number of OUs in an organization's domain, and 2) we specify that for each top level department there shall be only one OU.

Under the structure described in 2.2.2 the OU=Central_Office in the HartfordSchools.org domain should properly be broken into its roughly thirteen component departments. At this time, it has already been implemented as a single Central_Office OU. For the time being, we will leave this structure intact. We will evaluate the function of the structure specified in 2.2.2 as we

implement it in Hartford.gov and decide based on that evaluation whether to revisit Central_Office and fracture it into true departmental OUs.

Unlike Novell Directory Services, placement within the MS-AD structure as we have implemented it, is more of an administrative distinction than a functional one. With the flat email address structure specified in 4.1, a user's placement within their organization's domain has no bearing on their messaging environment. Authentication is *context-less* and is not affected by placement in one OU or another: the OU structure is generally transparent to the end user, thus the end-user experience cannot be used to argue for one OU structure or another.

2.3 To be a Domain or Not to be a Domain

2.3.1 Preferred Practice

Wherever possible organizations should be fitted in to the Hartford.gov domain or the HartfordSchools.org domain, instead of creating new peer domains.

2.3.2 Discussion

There are a number of conditions under which an organization may not be nestled under Hartford.gov or HartfordSchools.org: cases where the organization is a quasi-governmental body and there is strong political incentive to maintain the appearance of separation; cases where the organization has an existing IT department and has both the capability and desire to support their own MS-AD domain, either as a stand-alone MS-AD forest or a peer domain in the Hartford.Metro forest.

In these cases we should evaluate placement of the organization with an eye to the following criteria:

- Are they self supporting, or do they rely on MHIS for support?
- Who would bear the financial impact of maintaining additional domain controllers for a new peer domain in the Hartford.Metro forest?
- Do they have an established strong Internet domain identity that would make migration into Hartford.gov or HartfordSchools.org unduly burdensome?

Let us examine three cases to illustrate these criteria.

1. Hartford Economic development -- Econ. Dev. relies on MHIS for technical support, would probably not be interested in bearing an economic cost for maintaining their own domain controllers, and does not have such an extensive staff that migrating their email services into Hartford.gov would be overly burdensome. Thus, Economic Development should be fitted into Hartford.gov as an OU (econ_dev\hartford.gov).
2. The Hartford Parking Authority presents a similar case. They rely on MHIS for network services, and some additional support. Also, while they might have the economic wherewithal to pay for domain controllers for a HartfordParking.com MS-AD domain, they lack the technical staff to support them. Their Internet presence under the

HartfordParking.com Internet domain is small and thus not a problem to migrate. Thus, they too should be fitted into Hartford.gov as an OU (HPA\Hartford.gov).

3. Hartford Public Library exemplifies the case for establishing a peer domain within Hartford.metro or a free-standing forest of their own. Beyond basic network services and Internet access, HPL is self-supporting. They have both the wherewithal and staff to pay for and maintain MS-AD domain controllers, and they have a long established Internet domain presence as HPL.Lib.CT.US. Thus, they should be created as a peer domain within Hartford.metro if they wish to access enterprise services such as GIS, or as their own free-standing forest if they desire more complete separation.

2.4 Server Placement

2.4.1 Proximity as the Rule

Servers should be placed in the domain where the majority of their users are located. For example, a server with a Tax department application should be placed within Hartford.gov. Under the server naming convention (Standards Doc #0003) a Windows 2000 server in this example would be something like hg-2k-tax -- with the "tax" portion of the server name being the discretionary part.

2.4.2 Enterprise Shared Servers

Servers whose function is truly enterprise-spanning should be placed in Hartford.Metro. For example, our DHCP and DNS servers are in Hartford.Metro (hm-2k-dhcp and hm-2k-dns1 & hm-2k-dns2 respectively). If in doubt place the server in the best-fit organizational domain: placement in Hartford.Metro should be reserved for systems that serve clearly identifiable enterprise-wide functions. Best-fit can also be determined by asking who sponsored the application or service that caused the server to be built in the first place.

2.4.3 Discussion

Section 2.4.2 indicates that only servers with true enterprise-wide functions should be placed in Hartford.metro. The standard should be interpreted with a bias toward *infrastructure* systems. You will note that the two examples given are DNS and DHCP. User application servers should probably not go into Hartford.metro under any circumstances.

3.0 Login Scripts and Drive Mappings

3.1 Login Scripts

3.1.1 Script Location

In MS-AD login scripts can exist at the Domain, Container, and User levels. While there is an attraction to maintaining monolithic Domain login scripts, we will implement login scripts at the Container level, and reserve the use of User login scripts to meet specific, specialized needs that are difficult or inappropriate for a Container login script.

3.1.2 Existing Scripts

The initial public implementation of MS-AD within HartfordSchools.org – in support of the SASI application – has one login script at the domain level. This script will be migrated into OU-based login scripts as the client PCs are migrated off of Windows 95/98 and on to Windows 2000/XP.

3.2 Home Directories

MHIS will use Container login scripts to map user home directories. MS-AD provides a facility to “automagically” map user home directories based on a property of the User object: we will not use this facility.

3.3 Drive Mappings

The following drive letters are assigned for use in the MS-AD structure. All but P: are in use in the NDS structure and will require migration.

- I: Container-based shared files volume
- K: Kardex – maps to one server for *all* users
- H: User home directories (mnemonic ‘h’ for “home”)
- N: SASI (mnemonic ‘n’ for “NCS”)
- P: Common file “parking lot” – maps to one server for *all* users
- R: Budget prep folder in central_office container
- S: SmartStream – maps to one server for *all* users

Additional drive letters will be identified for City functions in later revisions of this document.

4.0 Messaging Architecture

4.1 Directory Structure and Messaging

The messaging structure will follow the top-level domain structure but not the lower level container structure. This means that if there is a user within the Hartford Police Department named Trevor Harding -- hardt001 under the convention established in Standards Doc. #0004 -- their email address would not reflect their location within the directory structure beyond Hartford.gov. The user would be CN=hardt001.OU=Police within the Hartford.gov domain, while their email address would be hardt001@Hartford.gov. Similarly, the author would be ships001@Hartford.gov if his account were created new today.

4.2 Strategy for "Prestige" Internet Domains

Should an organization within the MS-AD structure have a "prestige" Internet domain name -- such as HartfordPolice.com and HartfordParking.com -- the canonical email addresses for members of those departments shall be within their MS-AD domain. To wit, the hypothetical

police officer described in 4.1 would be `hardt001@hartford.gov`, not `hardt001@hartfordpolice.com`. We will maintain a small number of addresses for these domains as secondary email addresses for users within `Hartford.gov` or `HartfordSchools.org`. Examining `HartfordPolice.com`, such addresses might be: `admin`, `postmaster`, `webmaster`, `chief`, `officerfriendly` and so on.

4.3 Discussion

Consistency would dictate that the user's email address follow their Domain - Container naming; that the user `slivb001.Central_Office.HartfordSchools.org` should have the email address `slivb001@Central_Office.HartfordSchools.org`. We are not implementing this method. Following the container model to its logical end would result in `slivb001@Central_Office.HartfordSchools.org`, however this would require annoying email address changes and extra administrative work if the user moved from one area to another. If `slivb001` takes on a job in a high school her address would have to change to `slivb001@High_School.HartfordSchools.org` under a strict interpretation of the model. In order to simplify both our lives as administrators and the lives of our user community we have chosen to implement the flat email addressing model described in 4.1: the directory structure should be transparent unless its being visible is helpful to the users or administrators.