Migrating from MetaFrame 1.8 to MetaFrame XP

By

Citrix Consulting Services
Citrix Systems, Inc.
Notice

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED “AS IS” WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CITRIX SYSTEMS, INC. (“CITRIX”), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

This publication contains information protected by copyright. Except for internal distribution, no part of this publication may be photocopied or reproduced in any form without prior written consent from Citrix.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its own.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Copyright © 2001 Citrix Systems, Inc., 6400 NW 6th Way, Ft. Lauderdale, Florida 33309 U.S.A. All rights reserved.

<table>
<thead>
<tr>
<th>Version History</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 27, 2001</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

INTRODUCTION ................................................................................................................... ..................................................... 1

MIGRATION METHODOLOGY ........................................................................................................ 2

  ANALYSIS ............................................................................................................................... .................................................. 2
  DESIGN ........................................................................................................................................ 5
  IMPLEMENTATION .................................................................................................................. 6
  READINESS ................................................................................................................................ 6
  PRODUCTION ROLLOUT .......................................................................................................... 7

MIGRATION STRATEGIES ........................................................................................................... .......................................... 8

  STRATEGY #1: MIGRATE EXISTING MetaFrame 1.8 SERVER FARM ............................................ 8
  STRATEGY #2: CREATE A NEW MetaFrame XP SERVER FARM .................................................. 8
  STRATEGY #3: CREATE A NEW MetaFrame XP SERVER FARM WITH NEW SERVER INSTALLATIONS .................................................. 9

METAFRAME XP MIXED MODE ........................................................................................................ 10

  MIXED MODE ARCHITECTURE .............................................................................................. 10
  MIXED MODE CONSIDERATIONS AND RECOMMENDATIONS ............................................. 13

MIGRATION SCENARIOS, PROCESSES, AND BEST PRACTICES .................................................. 21

  SCENARIO #1 ............................................................................................................................. ................................................ 21
  SCENARIO #2 ............................................................................................................................. ................................................ 27
  SCENARIO #3 ............................................................................................................................. ................................................ 34

APPENDIX A – NFUSE MULTI-FARM CONSOLIDATION ................................................................. 42

  OVERVIEW .................................................................................................................................. 42
  CREATING A SINGLE POINT OF ACCESS .................................................................................. 42
  CUSTOMIZING THE MULTIPLE SERVER FARM SITE ................................................................. 42
  NFuse Java Objects, Substitution Tags, and Session Fields ...................................................... 44
## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>MetaFrame XP Mixed Mode Architecture</td>
<td>11</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>MetaFrame XP Native Mode Architecture</td>
<td>12</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>Interoperability Tab</td>
<td>17</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>QServer Command</td>
<td>18</td>
</tr>
<tr>
<td>FIGURE 5</td>
<td>ICA Gateways Tab</td>
<td>18</td>
</tr>
<tr>
<td>FIGURE 6</td>
<td>Scenario 1 Existing Infrastructure</td>
<td>22</td>
</tr>
<tr>
<td>FIGURE 7</td>
<td>ICA Client TCP/IP Setting</td>
<td>23</td>
</tr>
<tr>
<td>FIGURE 8</td>
<td>Scenario 1 Planned Native MetaFrame XP Architecture</td>
<td>24</td>
</tr>
<tr>
<td>FIGURE 9</td>
<td>Scenario 2 Existing Infrastructure</td>
<td>27</td>
</tr>
<tr>
<td>FIGURE 10</td>
<td>Scenario 2 Planned Native MetaFrame XP Architecture</td>
<td>29</td>
</tr>
<tr>
<td>FIGURE 11</td>
<td>Scenario 2 Migration Architecture</td>
<td>31</td>
</tr>
<tr>
<td>FIGURE 12</td>
<td>Scenario 3 Existing Infrastructure</td>
<td>34</td>
</tr>
<tr>
<td>FIGURE 13</td>
<td>ICA Client TCP/IP + HTTP Setting</td>
<td>36</td>
</tr>
<tr>
<td>FIGURE 14</td>
<td>Scenario 3 Planned Native MetaFrame XP Architecture</td>
<td>37</td>
</tr>
<tr>
<td>FIGURE 15</td>
<td>Scenario 3 Migration Architecture</td>
<td>38</td>
</tr>
<tr>
<td>FIGURE 16</td>
<td>NFuse Multi-Farm Consolidation Process Flow</td>
<td>43</td>
</tr>
</tbody>
</table>
**Introduction**

This white paper presents a structured methodology from Citrix Consulting Services (CCS) that can be used as a model for planning your migration from MetaFrame 1.8 to MetaFrame XP. This white paper presents the CCS methodology model and then applies the model to the migration process.

This white paper presents sample migration strategies, scenarios, and best practices for transitioning to a native MetaFrame XP architecture comprised of only MetaFrame XP servers. A native MetaFrame XP architecture, based on the Independent Management Architecture (IMA), includes many benefits that are described in detail in the *Citrix MetaFrame XP Application Server for Windows Administrator’s Guide* (hereafter referred to as the MetaFrame XP Administrator’s Guide). While MetaFrame XP does provide interoperability with previous versions of MetaFrame, the full benefits of MetaFrame XP are realized only when operating in a native MetaFrame XP environment.

The CCS methodology features tested and proven design and implementation techniques for large-scale deployments of server-based computing utilizing Citrix MetaFrame. The methodology ensures project success through early detection and mitigation of risks, rigorous project planning, and effective execution of the methodology and project plans. This methodology is the result of years of experience performing successful large-scale MetaFrame implementations.

**Note:** This white paper requires familiarity with the concepts and configurations found in the *MetaFrame XP Administrator’s Guide* and *Advanced Concepts for MetaFrame XP*. References to these documents are made throughout this white paper to avoid repetition of information.

Both of these documents can be downloaded from the Citrix Web site at [http://www.citrix.com](http://www.citrix.com). Click the Support link and then select the Product Documentation tab. Choose “MetaFrame XP Application Server for Windows” and then click “go.”
Migration Methodology

The CCS migration methodology is made up of the following five main phases:

- Analysis
- Design
- Implementation
- Readiness
- Production Rollout

In addition to the five main phases, a management checkpoint is included at the end of each phase to review deliverables and assess overall project status. Project management is also required throughout each phase. The CCS migration methodology is depicted in the following diagram.

The five phases of the CCS methodology are explained below.

Analysis

The analysis phase is broken down into the following four segments:

- Project Scope
- Project Plan
- Infrastructure Assessment
- Proof of Concept

The following deliverables are created during the analysis phase:

- Project Plan
- Infrastructure Assessment

In addition, if a Proof of Concept is conducted, the results of the Proof of Concept are also published.
Project Scope

The scope of the MetaFrame XP migration is identified in the project scope segment of the analysis phase. What is included and not included in the migration is detailed as part of the scope definition. When the project scope is defined, project/phase success criteria are also developed to help gauge the progress of the migration.

Project Plan

When the project scope is completely defined, it is necessary to detail the tasks to be completed during the MetaFrame XP migration along with estimated times to completion and project milestones (project milestones generally coincide with the management checkpoints). The tasks that are necessary for a successful MetaFrame XP migration are outlined in the Project Plan (the form of this deliverable is typically a Microsoft Project file or a Gantt chart).

Infrastructure Assessment

The existing MetaFrame 1.8 environment is reviewed during the infrastructure assessment segment. The infrastructure assessment reviews several areas that are critical to successfully migrating from MetaFrame 1.8 to MetaFrame XP. The infrastructure assessment:

- Determines if anything in the existing environment presents risks to the migration
- Identifies considerations that should be taken
- Identifies existing infrastructure components that can be leveraged as part of the migration
- Gathers requirements that will feed the migration design process

The infrastructure assessment reviews the following areas of the MetaFrame 1.8 environment:

- **Farm Architecture**: This area includes a review of the existing farm architecture, including the number of farms, the sizes of the farms, and how the farms are grouped. Migration to MetaFrame XP may require a re-evaluation of farm design and implementation.
- **ICA Browser**: This area includes a review of the existing ICA Browser configuration. The number of master ICA Browsers, backup ICA Browsers, and member servers are noted.
- **Servers**: This area identifies the number of servers in the environment and their specifications. The capability of the hardware is evaluated based on anticipated user load and growth. The servers’ CPU, memory, physical disk, and network interface are validated during the infrastructure assessment.
- **Load Balancing**: The current load balancing structure is analyzed to validate that the structure of the load-balanced groups is optimized to take advantage of MetaFrame’s load balancing capabilities. The load-balanced groups should be built with consideration to application usage, application dependencies, application compatibilities, or application upgrade frequency.
- **Network Structure**: This area defines the network structure configuration. Network structure analysis includes the physical design of the network and how the MetaFrame servers communicate with other servers. Login script design, policy configuration, security/lockdown, and connection methods (LAN, WAN, or dial-up) are also analyzed during this phase.
- **Users**: This area defines the number of users who currently access the farm and the anticipated load that they will place on the farm. It is most useful to group users into categories such as power users, medium users, and light users. The findings of this analysis can dictate farm and load-balanced group definition.
Applications: This area reviews information about published applications. The published applications are analyzed based on their type (32-bit or 16-bit), version number, and memory requirements.

Application Delivery: This area reviews how applications are currently delivered. It also includes a review of ICA Client configuration.

Licensing: This area reviews the existing licensing strategy and how any existing gateways are configured for license pooling.

Following the completion of the infrastructure assessment, a management checkpoint is conducted to review the results and findings of the infrastructure assessment. The results of the infrastructure assessment are used to develop requirements for the design phase. The infrastructure assessment determines whether to perform a Proof of Concept, to continue on to the design phase, or to address additional issues.

Proof of Concept

A Proof of Concept (POC) proves to the organization’s business managers, users, and information technology teams that an application can be successfully delivered using the MetaFrame XP architecture. It also allows an organization to identify hardware, software, and business process issues prior to system implementation. The POC demonstrates the key technologies involved and provides an environment for experimentation and evaluation. The design and implementation of a POC, while very detailed and organized, does not serve as a replacement for a complete system analysis and design. The POC uses a controlled environment to simulate the organization’s target technical environment and, utilizing representative business functions, demonstrates MetaFrame XP’s benefits to the organization.

Objectives must be clearly defined prior to undertaking a POC. These objectives most often fall into the following categories:

- **Corporate Objectives** – High-level goals that focus on reducing overall system cost, seeking a method to improve system redundancy, and reducing information technology related incidents that reduce employee effectiveness. These requirements are linked to a search for an enterprise solution that can be leveraged throughout the company.

- **Management Objectives** – Mid-level goals that focus on addressing specific deployment/integration issues, including:
  - Reducing desktop support activities.
  - Correcting specific application issues, such as slow WAN performance
  - Deploying applications to users faster, with fewer resources
  - Enhancing network security and streamlining user deployment/configuration

- **User Objectives** – Low-level goals that focus on the quality of service provided by the desktop or Windows-based terminal configuration, including:
  - Faster application performance
  - Enhanced application stability
  - Faster issue resolution
  - Better access to new applications

When the POC is complete another management checkpoint is performed to review the results of the POC. The results of the checkpoint determine whether to continue on to the design phase.
Design

The design phase includes the following segments:

- Native MetaFrame XP architecture
- Identify migration strategy
- Migration architecture

The following deliverables are created during the design phase:

- Native MetaFrame XP Architecture Design
- Migration Architecture Design

Native MetaFrame XP Architecture

To determine the migration strategy, the native MetaFrame XP architecture must be planned. The Native MetaFrame XP Architecture Design covers the following areas:

- **Farm Architecture**: This area defines the architecture of the new MetaFrame XP farm.
- **Load Management**: This area defines the configuration of Load Management evaluators. MetaFrame XP manages load across farms and at the application level. The Load Management design takes into consideration the optimal settings for the load-managed farm.
- **Data Store**: This area defines the configuration of the MetaFrame XP data store. Based on the infrastructure size, the data store hardware configuration may need to be designed. This includes operating system choice and relational database selection (Oracle, SQL).
- **Data Collector**: This area defines the configuration of the MetaFrame XP data collectors.
- **Zones**: This area defines the MetaFrame XP zone configuration.
- **Applications**: This area defines whether existing application installations should be used in the native MetaFrame XP architecture.
- **Application Delivery**: This area defines how applications will be delivered to clients. It includes ICA Client configuration.
- **Licensing**: This area defines the licensing model to be used.

Identify Migration Strategy

The migration strategy describes the transition process from the current MetaFrame architecture to the planned native MetaFrame XP architecture, as identified in the Native MetaFrame XP Architecture Design. The migration strategy includes a description of how servers, applications, and users will be migrated from MetaFrame 1.8 to MetaFrame XP. For more information on migration strategies, refer to the “Migration Strategies” section of this document.
Migration Architecture

The migration architecture must be defined once a migration strategy has been identified. This migration architecture will include the infrastructure design that supports the chosen migration strategy. The Migration Architecture Design includes the following areas:

- **Farm Architecture**: This area defines the MetaFrame XP farm architecture during the migration period.
- **Load Balancing/Management**: This area defines how load balancing and Load Management evaluators will be configured.
- **Data Store**: This area defines the configuration of the MetaFrame XP data store.
- **Master ICA Browser**: This area defines the master ICA Browser configuration.
- **Data Collector**: This area defines the configuration of the MetaFrame XP data collectors.
- **Zones**: This area defines the MetaFrame XP zone configuration.
- **Applications**: This area defines the configuration of published applications during the migration.
- **Application Delivery**: This area defines how applications will be delivered to users during the migration period. It includes ICA Client configuration.
- **Licensing**: This area defines how licensing will be addressed during the migration.

Implementation

The implementation phase includes the development and testing of any components or scripts that were identified and designed during the design phase.

Examples of implementation phase deliverables include:

- Unattended Installation scripts used to build a new server, including the operating system, MetaFrame, and user applications and configurations
- Logon/logoff scripts that are used to customize the user’s environment
- Components needed to integrate published applications into a Web portal using NFuse
- Customized NFuse Web pages

Readiness

The readiness phase consists of the following two main segments:

- **Testing**
- **Pilot**

The testing and pilot segments are used to verify that the native MetaFrame XP architecture and the migration architecture will scale to support production users.
Change Management

When migrating from MetaFrame 1.8 to MetaFrame XP, it is important to have a stringent change control process in place. An unmanaged MetaFrame environment creates instability and is difficult to maintain and support. Organizations have many changes driven by the business and information technology teams that impact the MetaFrame environment both directly and indirectly. All changes must be tracked to allow for proper risk management of the MetaFrame environment.

A formal change control process requires administrators to document every change and prove that it has been through an aggressive testing environment. By following a change control process the MetaFrame environment becomes more stable and efficient.

It is highly recommended that for each change the following process be followed:

Testing --> Pilot --> Production Rollout

This process requires that the change be implemented first in a test environment. The change is then migrated to a pilot environment, and finally to the full production environment. Before each migration, the change is fully tested in the environment to ensure that it functions correctly.

Testing

Testing is used to identify and fix potential issues with the planned native MetaFrame XP architecture or the migration architecture prior to production rollout. Testing includes a full system test and scalability testing.

System Test

The system test tests both the native MetaFrame XP architecture and the migration architecture to ensure that all functionality and quality requirements have been met. System testing occurs on multiple levels as each component of the overall environment is tested alone and in conjunction with the other components (as they would be used in a production environment). The test environment should resemble the production environment as closely as possible, including server hardware, server configuration, and network configuration.

Scalability Testing

Scalability testing includes testing for both server and network scalability.

Pilot

A pilot implementation allows you to “test drive” a Citrix solution in a live production environment. The pilot implementation is the proving ground for network designs, application compatibility, and other requirements identified during the analysis and design phases. The pilot consists of an actual production environment with live users. The pilot provides the baseline environment on which additional enhancements can be tested.

Production Rollout

The production rollout phase consists of the installation and configuration of the non-pilot portion of the production environment. This includes the rollout of the migration architecture that will evolve into the native MetaFrame XP architecture.
Migration Strategies

This paper presents the following three strategies for migrating from MetaFrame 1.8 to MetaFrame XP. Each of these migration strategies is explained in detail in this section.

- **Strategy #1: Migrate an existing MetaFrame 1.8 server farm.** The existing MetaFrame 1.8 servers are migrated to MetaFrame XP in a phased approach, using MetaFrame XP "mixed" mode of operation.
- **Strategy #2: Create a new MetaFrame XP server farm.** Two farms, a MetaFrame 1.8 farm and a MetaFrame XP farm, coexist but do not interoperate in mixed mode.
- **Strategy #3: Create a new MetaFrame XP server farm with new server installations.**

**Strategy #1: Migrate Existing MetaFrame 1.8 Server Farm**

This first migration strategy migrates an existing MetaFrame 1.8 server farm to MetaFrame XP. The existing MetaFrame 1.8 servers are upgraded and migrated to MetaFrame XP in a phased approach. MetaFrame XP mixed mode operation is used to allow both the new MetaFrame XP farm and the existing MetaFrame 1.8 farm to interoperate together until the migration is complete.

Reasons to choose this migration strategy include the following:

- The company has a large MetaFrame 1.8 farm, making it difficult to quickly migrate to a new MetaFrame XP farm.
- The size of the server farm requires that existing MetaFrame 1.8 servers interoperate with the new MetaFrame XP servers to handle the production load without requiring users to connect to two separate farms.
- The company wants to maintain existing application deployments by migrating published applications to MetaFrame XP.
- The company wants to use existing server installations during the migration.

MetaFrame XP mixed mode operation is designed to facilitate migration to MetaFrame XP; it is not designed to be a permanent solution. For specific considerations, issues, and recommendations regarding MetaFrame XP mixed mode operation refer to the "MetaFrame XP Mixed Mode" section of this document. For instructions on how to enable MetaFrame XP mixed mode during MetaFrame XP installation, see the *MetaFrame XP Administrator’s Guide*.

**Strategy #2: Create a new MetaFrame XP Server Farm**

This migration strategy consists of migrating MetaFrame 1.8 servers from an existing MetaFrame 1.8 farm to a new MetaFrame XP server farm operating in native mode. The MetaFrame XP farm does not interoperate with the existing MetaFrame 1.8 farm. Two separate farms, a MetaFrame 1.8 farm and a MetaFrame XP farm, co-exist until all MetaFrame 1.8 servers are migrated.

Reasons to choose this migration strategy include the following:

- The company wants to maintain existing application deployments by migrating published applications to MetaFrame XP.
- The company wants to use existing server installations during the migration.
Migrating from MetaFrame 1.8 to MetaFrame XP

- The company wants to segregate and migrate users to the new MetaFrame XP farm.
- The company wants to use MetaFrame XP IMA communication and the full MetaFrame XP feature set.

Strategy #3: Create a New MetaFrame XP Server Farm with New Server Installations

The third migration strategy is to create a new MetaFrame XP server farm with new server installations. This strategy mirrors the second migration strategy with the exception that servers receive clean system installations. In this scenario, existing or new servers can be utilized for the migration.

Reasons to choose this migration strategy include the following:

- The company wants to deploy new applications or re-configure existing applications.
- The company wants to implement new hardware when creating the new MetaFrame XP environment.
- The company wants to implement a new architecture for the following reasons:
  - The physical architecture of the existing implementation needs to be changed. For example, MetaFrame servers are currently deployed to multiple sites, and the company wants to physically consolidate the servers.
  - The company is planning other infrastructure implementations. For example, the company is moving from Windows NT 4.0 to Windows 2000.
- The company wants to use MetaFrame XP IMA communication and the full MetaFrame XP feature set.
- The company prefers a “clean” system migration.
- The company wants to segregate and migrate users to the new MetaFrame XP farm.
MetaFrame XP Mixed Mode

MetaFrame XP supports two modes of operation: native mode and mixed mode. MetaFrame XP native mode does not allow for interoperability with previous versions of MetaFrame. Organizations that have existing MetaFrame 1.8 farms may need to integrate new MetaFrame XP servers with the MetaFrame 1.8 farm for migration purposes. MetaFrame XP mixed mode allows MetaFrame XP server farms to interoperate with MetaFrame 1.8 server farms. Mixed mode allows a MetaFrame 1.8 farm and a MetaFrame XP farm to interact together so they appear to ICA Clients as one server farm.

This section discusses MetaFrame XP mixed mode operation, how it differs from MetaFrame XP native mode operation, and how it pertains to migration. This section also presents recommendations for operating in mixed mode.

Mixed Mode Architecture

Mixed mode must support the existing features of the MetaFrame 1.8 farm across both the existing MetaFrame 1.8 farm and the new MetaFrame XP farm. This is accomplished by running the Program Neighborhood and ICA Browser services on each MetaFrame XP server in the farm. These services are installed during MetaFrame XP installation. The functionality of these services also exists in a native MetaFrame XP environment as subsystems of the IMA service.

When MetaFrame XP mixed mode is enabled, the Program Neighborhood and ICA Browser services are started. The IMA-based Program Neighborhood and ICA Browser subsystems stop their listeners, effectively disabling them. The Program Neighborhood and ICA Browser services allow the MetaFrame XP servers to communicate and share information with MetaFrame 1.8 servers. MetaFrame XP servers running in mixed mode still communicate with other MetaFrame XP servers using the native Independent Management Architecture protocol.
The MetaFrame XP mixed mode architecture is depicted in the following diagram:

When MetaFrame XP native mode is enabled, the Program Neighborhood and ICA Browser services are stopped and the IMA-based Program Neighborhood and ICA Browser subsystems are enabled. MetaFrame XP servers can no longer communicate and share information with MetaFrame 1.8 servers.

Figure 1 MetaFrame XP Mixed Mode Architecture
The MetaFrame XP native mode architecture is shown below:

**Figure 2 MetaFrame XP Native Mode Architecture**

The operating mode is determined by the global “interoperability flag” set when MetaFrame XP is installed on the first server in a new farm. The operating mode is a farm-wide setting and can be changed at any time by modifying the farm properties using the Citrix Management Console. Once a farm is configured for a mode of operation, all existing servers in the farm are changed to that mode.

Each MetaFrame XP server queries the interoperability flag during boot up, and enables or disables the correct IMA sub-systems accordingly. This operation places the server in the configured mode. All new servers that are added to the farm are automatically put into this mode.

The MetaFrame XP servers continue to monitor the interoperability flag after boot up. The appropriate services are enabled or disabled if the operating mode changes. A reboot is not required to switch between operating modes.

Because it takes time for a change in the operating mode to propagate to all servers in the farm, ICA Clients may be unable to refresh applications in Program Neighborhood or browse for published applications until the change has been propagated (current ICA connections are not affected). It is recommended that the operating mode be changed when the fewest users are connected to published applications.

Native mode and mixed mode can be compared as follows:

- In mixed mode, the XML Service connects to the Program Neighborhood Service using Program Neighborhood Named Pipes. In native mode, the XML Service connects to the IMA Service using an IMA Remote Procedure Call (RPC).
The ICA Client uses the Program Neighborhood virtual channel to connect to the Program Neighborhood Service in mixed mode; and to the Program Neighborhood subsystem in native mode.

The ICA Client uses the ICA Browser protocol (UDP Port 1604) to connect to the ICA Browser service in mixed mode; and to the Browser subsystem in native mode.

In mixed mode, the Program Neighborhood and ICA Browser services exist and are enabled, while the Program Neighborhood and Browser subsystems are disabled. The Program Neighborhood and ICA Browser services interact with the local Windows registry.

In mixed mode, Citrix Server Administration (mfadmin.exe) makes RPC connections to all MetaFrame 1.8 and MetaFrame XP servers. It also connects to Termsrv via Winstation API (RPC). In native mode, Citrix Server Administration makes RPC connections only to MetaFrame 1.8 servers.

In mixed mode and in native mode, Published Application Manager (appcfg.exe) reads application information only from MetaFrame 1.8 servers. Published applications for MetaFrame XP are managed only through the Citrix Management Console.

The IMA service exists in both modes of operation. It communicates with other servers via the IMA protocol over TCP Port 2512. It also connects to Termsrv via Winstation API (RPC), the local host cache via ODBC, and the data store via ODBC. The IMA service interacts with the local Windows registry only in mixed mode.

Note: There are more differences between native mode and mixed mode, including differences in load balancing, license pooling, and gateways, that should be considered before implementing mixed mode in a production environment. See Advanced Concepts for MetaFrame XP for a detailed description of these differences.

Mixed Mode Considerations and Recommendations

The following sections describe considerations to take and recommendations to follow when implementing a MetaFrame XP mixed mode farm.

Farm Name

The MetaFrame XP server farm must have the same name as the MetaFrame 1.8 server farm. When you install MetaFrame XP on the first server in the farm, you name the server farm at the same time you create the data store.

Note: ICA Clients see the MetaFrame XP and MetaFrame 1.8 farms operating in mixed mode as a single farm. However, they are two separate farms.

Management Utilities

MetaFrame XP farms and MetaFrame 1.8 farms are managed by separate utilities. You can manage a MetaFrame 1.8 farm using MetaFrame 1.8 utilities including Citrix Server Administration (mfadmin.exe) and Published Application Manager (appcfg.exe). You should use the updated versions of these tools that are installed on each MetaFrame XP server. Running previous versions of these tools from the existing MetaFrame 1.8 servers is not recommended (refer to the “Published Application Migration” section of this document for additional information).

Use the Published Application Manager utility to configure and modify published applications for MetaFrame 1.8 servers. Use the Citrix Server Administration utility to configure options on MetaFrame 1.8 and MetaFrame XP servers. However, the settings on MetaFrame XP servers take effect only when the server farm is operating in mixed mode.
Use the Citrix Management Console to manage a MetaFrame XP farm. Published Application Manager cannot be used to manage applications migrated to MetaFrame XP servers.

**Shadowing**

In MetaFrame XP native mode operation, the Shadow Taskbar displays MetaFrame XP servers in a server farm. In MetaFrame XP mixed mode operation, the taskbar displays servers in both the MetaFrame 1.8 farm and the MetaFrame XP farm.

**License Pooling in Mixed Mode**

For information and examples about license pooling and how license gateways are affected by mixed mode, refer to the “Native and Mixed Mode Details” section of *Advanced Concepts for MetaFrame XP*.

**Load Balancing in Mixed Mode**

Although MetaFrame XP servers interoperate with MetaFrame 1.8 servers when operating in mixed mode, the new MetaFrame XP Load Management functionality is available only on MetaFrame XP servers running in native mode.

Successfully load balancing applications between MetaFrame 1.8 servers and MetaFrame XP servers requires the following:

- One of the MetaFrame XP servers must be the master ICA Browser. If a MetaFrame XP server is not the master ICA Browser, the two farms cannot be load balanced.

- Only the Default Load Evaluator should be attached to all the MetaFrame XP servers.

It is possible to add different server load evaluators to the MetaFrame XP servers in mixed mode, although it is very difficult to load balance the MetaFrame XP servers with MetaFrame 1.8 servers in a logical way. Therefore, load balancing in mixed mode is supported only when the Default Load Evaluator is used.

- Applications to be load balanced must first be created in Published Application Manager in the MetaFrame 1.8 farm, and then created with the exact same name in the MetaFrame XP farm using the Citrix Management Console.

  If applications are not published with an identical name, the applications are considered to be different applications to the master ICA Browser and will not load balance across MetaFrame 1.8 and MetaFrame XP servers.

  Applications must be published on the MetaFrame 1.8 servers before being published on the MetaFrame XP servers. If applications are published on the MetaFrame XP servers first, attempting to publish the applications on the MetaFrame 1.8 servers returns an error stating that the application could not be created because the application name already exists.

The *qserver /app* command is used to report load values for MetaFrame 1.8 farms and MetaFrame XP farms operating in mixed mode. The *qfarm /app* and *qfarm /load* commands are used to report load values for MetaFrame XP farms operating in native mode. While both the *qserver* and *qfarm* commands can be used to report load values for a MetaFrame XP farm operating in mixed mode, they are based on different timing mechanisms and update at different time intervals. Consequently, these values may return different results. In mixed mode, ICA Clients send application name resolution requests to the master ICA Browser. The master ICA Browser resolves the application to the proper server using the load values seen by *qserver /app*, not *qfarm*. Therefore, only the *qserver /app* command should be used to check load values in a mixed mode MetaFrame XP farm.
A MetaFrame XP farm operating in mixed mode follows a unique load balancing algorithm. When a connection is launched to a load-balanced application, the following process occurs:

- The ICA Client makes a request to the master ICA Browser to access the least loaded server on which the application is published.
- The master ICA Browser evaluates the loads on all MetaFrame XP servers on which the application is published, and chooses the least loaded server. This load is based solely on the number of user connections to the server.
- The master ICA Browser evaluates the loads on all MetaFrame 1.8 servers on which the application is published, and chooses the least loaded server.
- The master ICA Browser ignores both load values from the least loaded MetaFrame XP server and the least loaded MetaFrame 1.8 server. The master ICA Browser instead determines the number of active connections running on the two servers, and routes the client to the server with the least number of active connections. If both servers have the same number of active connections, the master ICA Browser will always route the ICA Client to the MetaFrame XP server.

The load balancing algorithm used in MetaFrame XP mixed mode is depicted in the diagram below.

In Step 1, the master ICA Browser compares the server load values of the MetaFrame XP servers (600, 200), and then compares the server load values of the MetaFrame 1.8 servers (300, 180). In this scenario, MF XP Server 2 has the least load of the MetaFrame XP servers, and MF 1.8 Server 2 has the least load of the MetaFrame 1.8 servers.
In Step 2, the master ICA Browser ignores the server loads of the final two servers and looks at the number of active connections to each server. In this scenario, MF XP Server 2 has two connections and MF 1.8 Server 2 has three connections. Thus, MF XP Server 2 will field the ICA Client request for the application session.

**ICA Browser Elections**

When MetaFrame XP is configured for mixed mode operation, MetaFrame 1.8 farms and MetaFrame XP farms appear unified because ICA Browsers in both farms pool information. A MetaFrame XP server becomes the master ICA Browser of both farms. The new master ICA Browser holds information about the published applications available on each server.

When the first MetaFrame XP server is brought online with an existing MetaFrame 1.8 farm, either by upgrade or new installation, the new MetaFrame XP server becomes the master ICA Browser on the subnet if mixed mode is chosen when MetaFrame XP is installed. This is an important migration consideration because of how it could affect the environment. It is recommended that the existing master ICA Browser be upgraded first, and then the backup ICA Browsers. The newly migrated server becomes the master ICA Browser for both the MetaFrame 1.8 farm and the MetaFrame XP farm. This same server also becomes the zone data collector for the MetaFrame XP farm.

If mixed mode operation is enabled when there is a MetaFrame 1.8 farm and a native mode MetaFrame XP farm on the same subnet, one of the MetaFrame XP servers will become the master ICA Browser. This is because the MetaFrame XP ICA Browser has a higher version level than the existing MetaFrame 1.8 ICA Browser.

The first MetaFrame XP server in the farm will host the data store if the option **Use a local database for the data store** is selected. The data store will be created as an Access database. If running in mixed mode operation, this server will also become the master ICA Browser.

If there is a need to host the data store on a server other than the master ICA Browser, perform the following steps to separate the data store from the master ICA Browser. These steps ensure that the previous master ICA Browser will become the new master ICA Browser while in mixed mode operation.

1. Install or upgrade the server that will host the data store. During this installation, do not choose **Operate in mixed mode with MetaFrame 1.8 servers**.

2. Upgrade the master ICA Browser and backup ICA Browser servers.

3. Open the Citrix Management Console and modify the farm properties to enable mixed mode operation.
Select the server that was the master ICA Browser prior to upgrading. Right-click the server and select **Properties**. The figure below displays the **Interoperability** tab.

![Interoperability Tab](image)

**Figure 3 Interoperability Tab**

Choose the following options on the Interoperability tab:

a. Under the section **During Master ICA Browser Election** select **Always attempt to become the master ICA browser**.

b. If backup ICA Browsers were upgraded, the number of backup ICA Browsers should be indicated in the **If Elected Master ICA Browser** section next to **Number of backup master ICA browsers**.

c. The properties of the data store server should be set to **Do not attempt to become the master ICA browser**.

4. Click **OK** to close the **Properties** dialog box for the master ICA Browser.

For additional information regarding the election of the master ICA Browser during mixed mode operation, refer to the “Election of the master ICA Browser” section in Chapter 3 of the *MetaFrame XP Administrator’s Guide*. 
ICA Gateways

In a MetaFrame 1.8 implementation, ICA Gateways may be in place to enable farms to span network subnets. ICA Gateways are also used in MetaFrame XP mixed mode to enable communication between master ICA Browsers on each subnet. This is indicated by a G listed next to each master ICA Browser in the results of a qserver command. The figure below displays a sample output:

![Figure 4 Qserver Command](image)

When ICA Gateways are created there is an initiating server (the server where the gateway was configured) and a receiving server (the remote server for the gateway as configured on the initiating server). These servers are listed in Citrix Server Administration (madmin.exe) on the ICA Gateways tab when All Listed Servers is selected, and also by running the qserver /gateway command. The figure below shows the ICA Gateways tab in the Citrix Server Administration utility.

![Figure 5 ICA Gateways Tab](image)

On the ICA Gateways tab in the figure above, the Local Server column displays the name of the initiating server and the Remote Server Address column displays the IP address of the receiving server.

When a MetaFrame 1.8 server that initiated an ICA Gateway is migrated to MetaFrame XP in mixed mode, this server is no longer displayed in the results of the qserver /gateway command. However, the actual ICA Gateway continues to exist between the master ICA Browsers on each subnet.
Client Connectivity

In MetaFrame XP mixed mode operation, ICA Clients communicate with a single master ICA Browser for both the MetaFrame 1.8 farm and the MetaFrame XP farm by connecting to servers in either farm. When launching published applications, ICA Clients contact the master ICA Browser for the address of a server that can run the application. ICA Clients also contact the master ICA Browser to find new application sets and servers that host published applications for custom connections. The single master ICA Browser allows ICA Clients to connect to servers in either the MetaFrame 1.8 or MetaFrame XP farm.

The operating mode of the MetaFrame XP farm determines how it responds to ICA Client connections requests, whether through “auto-locate” broadcasts or specified server locations. ICA Client communication also functions differently when MetaFrame 1.8 servers reside on the same network as MetaFrame XP servers, regardless of the MetaFrame XP operating mode. For a more detailed explanation of ICA Browsing and ICA Client configuration refer to Chapter 3 of the MetaFrame XP Administrator’s Guide, and Chapter 7 (“ICA Client Configuration”) of Advanced Concepts for MetaFrame XP.

Network Traffic

MetaFrame XP mixed mode operation requires two types of network traffic. Both MetaFrame 1.8 servers and MetaFrame XP servers communicate via UDP Port 1604 for MetaFrame 1.8 server communication. In addition, IMA TCP Port 2512 traffic exists between all MetaFrame XP servers for MetaFrame XP server communication.

Because operating in MetaFrame XP mixed mode results in increased network traffic and can affect network scalability, you should carefully consider whether to operate in this mode of operation.

Zones, Data Collectors, and Data Store

All standard MetaFrame XP components exist in mixed mode operation. MetaFrame XP zones continue to be established during server installation or through the Citrix Management Console. A data collector exists for each zone and a data store exists for each MetaFrame XP farm.

Published Application Migration

Applications published under MetaFrame 1.8 prior to migrating will upgrade successfully to MetaFrame XP. This is the case for both mixed mode and native mode operation. When MetaFrame 1.8 servers that host these applications are migrated to MetaFrame XP, the server is automatically added to the list of servers that host the application. This holds true only if the application is not modified after the first server is migrated to MetaFrame XP.

The following complications can arise if you add new applications or modify existing applications after the first server is migrated to MetaFrame XP (while the farm is still in mixed mode operation).

- If applications that are to be load balanced across both MetaFrame 1.8 and MetaFrame XP servers are published first using the Citrix Management Console, they cannot then be published using Published Application Manager (appcfg.exe). When trying to publish an application to MetaFrame 1.8 servers using Published Application Manager after publishing the same application to MetaFrame XP servers, the following error message is displayed (where appname is the name of the application being published):

  “The application appname is already in use. It does not appear in the application list because it has been configured for use on server(s) that you are not presently viewing. Please enter a different name.”

  When the application is published to MetaFrame XP servers using the Citrix Management Console, the MetaFrame XP servers add this application to their local registry under
HKLM\System\CurrentControlSet\Contro\ Citrix\Managed Applications. Published Application Manager (appcfg.exe) reads application information remotely from the registry of every other server in the farm. Because Published Application Manager is aware of the application that was published on the MetaFrame XP servers (using the Citrix Management Console), it does not allow a new published application to be created with the same name.

To eliminate this issue, publish applications on MetaFrame 1.8 servers using Published Application Manager from a MetaFrame XP server, and then publish the applications on the MetaFrame XP servers using the Citrix Management Console.

- When a migrated published application is load balanced across both MetaFrame 1.8 and MetaFrame XP servers, the application icon appears when the MetaFrame 1.8 version of Published Application Manager is opened. However if the Published Application Manager window is refreshed, the application icons disappear. The applications reappear when Published Application Manager is closed and re-opened. This occurs because refreshing Published Application Manager causes it to only read the registry of MetaFrame XP servers only. Because the registry of the MetaFrame XP server does not list all the servers for each published application, the icons disappear. The MetaFrame XP server lists only applications that are published locally. This issue does not occur when using the MetaFrame XP version of Published Application Manager (appcfg.exe).

- Published applications that have been migrated continue to display the newly migrated MetaFrame XP servers in each application’s list of configured servers within Published Application Manager. If removed from the list, the migrated servers cannot be re-added because MetaFrame XP servers can only be managed from the Citrix Management Console.

- Published applications modified using the version of Published Application Manager that is installed with MetaFrame XP cannot later be modified from the version of Published Application Manager installed on all MetaFrame 1.8 servers.

It is not recommended to add or modify published applications while in mixed mode operation. If it is necessary to add published applications to be load balanced across MetaFrame 1.8 and MetaFrame XP farms or to modify existing published applications, the following actions are recommended:

- When a published application has been added or modified, do not migrate published applications during subsequent MetaFrame 1.8 to MetaFrame XP migrations.

- After the MetaFrame 1.8 server has been migrated to the MetaFrame XP farm without migrating the published applications, use the Citrix Management Console to add this server to the list of configured servers for each application to be published on this server.

For additional details about published application migration and potential issues that may be encountered, refer to Chapter 5 of Advanced Concepts for MetaFrame XP.

**Application Migration Log**

When migrating MetaFrame 1.8 servers and their applications to MetaFrame XP, the migration log file should be referenced after the migration. The log file contains application migration results detailing whether each application migrated successfully. The log file is located in the %SystemRoot%\system32 directory.

**License Migration**

There are several different options for upgrading MetaFrame 1.8 licenses to MetaFrame XP licenses. For additional information, refer to the “Subscription Advantage” section of the *MetaFrame XP Frequently Asked Questions* document at [http://www.citrix.com/products/metaframexp/docs/MetaFrame_XP_Launch-FAQ-Customer.pdf](http://www.citrix.com/products/metaframexp/docs/MetaFrame_XP_Launch-FAQ-Customer.pdf).
Migration Scenarios, Processes, and Best Practices

This section provides three migration scenario examples using the methodology defined in the Migration Methodology section of this document to demonstrate the process and best practices for each type of migration. These scenarios use the second and third migration strategies. The first migration strategy is not recommended because of considerations discussed in the Mixed Mode Considerations and Recommendations section of this document.

The scenarios do not include a project plan as part of the analysis phase. Specific tasks and estimated timelines for those tasks can vary greatly from company to company. The following factors can affect the project plan:

- Timelines
- Resources
- Budget
- Scope
- Hardware and software availability

A migration project plan should provide estimates for all the required tasks. To determine the tasks required, customize the methodology to the requirements for the specific migration project. The migration scenarios describe hypothetical examples. Therefore, specific detailed project plans are not included as the scenarios cover multiple possible requirements and environments.

The scenarios do not include a Proof of Concept. It is assumed that the MetaFrame XP technical concept has already been proven to business owners.

Scenario #1

The first scenario describes an example of migrating from one MetaFrame 1.8 farm on a single subnet to a MetaFrame XP farm. The existing farm currently uses MetaFrame 1.8 with Feature Release 1 and Load Balancing Services. In this scenario new server hardware will be used for all MetaFrame XP servers.

Analysis

The analysis phase of the migration methodology includes a project scope definition and an infrastructure assessment.

Project Scope

Project management has defined the success criteria to include the following:

- Successful implementation of the MetaFrame XP environment
- Minimal impact to the user community

The main requirement of the project is for all MetaFrame 1.8 applications to be migrated to MetaFrame XP with new server hardware.
Infrastructure Assessment

The following diagram depicts the existing infrastructure environment:

![Diagram of MetaFrame 1.8 Farm]

**Figure 6 Scenario 1 Existing Infrastructure**

The infrastructure environment is described as follows:

**Farm Architecture** – All servers reside in one MetaFrame farm. All servers in the farm are running MetaFrame 1.8 with Feature Release 1.

**Master ICA Browser** – A master ICA Browser exists within the farm. The master ICA Browser also hosts applications.

**Servers** – There are 12 servers in the farm, all located in a central data center.

**Load Balancing** – The Load Balancing parameters selected are the default settings.

**Network Structure** – Only one network subnet exists. All servers reside on this subnet.

**Users** – There are 500 users that connect to published applications within the farm. Users access the farm from multiple company sites.

**Applications** – There are 25 published applications.
**Application Delivery** – Users currently connect using the ICA Client. Each ICA Client’s networking properties for an application set is configured to use TCP/IP. Given these client settings, UDP broadcasts are sent between the ICA Client and the MetaFrame servers. This setting is shown below:

![ICA Client TCP/IP Setting](image)

**Figure 7 ICA Client TCP/IP Setting**

**Licensing** – Each server has a base MetaFrame 1.8 server license valid for 15 users (180 total user licenses). Additional user bump backs are included to support all 500 users (1 20-user bump pack, 6 50-user bump packs). A MetaFrame 1.8 Feature Release 1 license is also installed on each server.

**Design**

The design phase of the migration methodology determines the planned native MetaFrame XP architecture, the migration strategy, and the migration architecture.
Planned Native MetaFrame XP Architecture

Below is a diagram of the planned native MetaFrame XP architecture:

The planned native MetaFrame XP architecture is broken down as follows:

**Farm Architecture** – The native MetaFrame architecture will have only one farm.

**Load Management** – Default Load Management settings will be used.

**Data Store** – The data store will be an Access database located locally on a MetaFrame server.

**Data Collector** – A dedicated data collector will be used. No applications will reside on this server. This server will also serve as the data store.

**Zones** – Because the servers within the farm are not spread across low bandwidth connections, all servers will be contained within one zone.

**Servers** – Servers will remain in their existing locations.

**Applications** – Application installations will be re-configured and re-deployed. Therefore, no applications will be migrated. New applications will also be deployed.

**Application Delivery** – Users will continue to connect to the farm using the ICA Client. Each ICA Client’s networking properties for an application set will be configured to use TCP/IP + HTTP. When these settings are changed, there will be no UDP Browsing from the ICA Client to the server. The ICA Client will instead connect via
TCP/IP + HTTP to the XML Service running on the MetaFrame server. The latest version of the ICA Client will be deployed using the auto-client update feature.

**Licensing** – A Universal Server license for MetaFrame XPa is needed for all servers. This license covers MetaFrame XP and Load Management. In addition, connection licenses must be purchased for each user that connects to the farm. Ten 50 user bump packs will be required for 500 users.

**Migration Strategy**

A new MetaFrame XP server farm with new server installations (Strategy #3) will be created. This option is chosen for the following reasons:

- The company wants to re-configure existing applications as well as deploy new applications.
- The company desires to have clean system installations.
- The company plans to migrate all servers at the same time.

**Migration Architecture**

This company has chosen to migrate all 12 servers at once during one migration period. Therefore, a migration architecture is not needed. The migration will transition straight from the existing environment to the planned native MetaFrame XP environment.

**Implementation**

During the implementation phase, scripts are developed to support the migration and rollout of MetaFrame XP. These scripts include the following:

- An automated scripted server installation. This is created with the use of VBScript and Windows Scripting Host and includes partitioning the disks on the new system, installing the operating system, installing MetaFrame XP, applying service packs, hot fixes, and registry tweaks, and installing applications. The installation of applications includes both the re-configured applications and the new applications.
- A script to migrate users to different groups. This script is used to move users from groups that have permissions for MetaFrame 1.8 published applications to groups that have permissions for MetaFrame XP published applications.

**Readiness**

The readiness phase includes testing and a production pilot.

**Testing**

A testing environment is established for the migration. Testing includes two segments:

1. **System Tests**: This segment tests all components of both the native MetaFrame XP architecture and the migration architecture.

2. **Scalability Tests**: Scalability tests are conducted for both the native MetaFrame XP architecture and the migration architecture. Different performance counters are monitored to test server performance and network performance.
Pilot
A pilot is performed before rollout of the migration architecture. The pilot is conducted using a small subset of users from the production environment. The pilot is run for user acceptance testing as well as to obtain business signoff for the architecture change.

Rollout
The following high-level steps are performed for the production rollout:

Migration Preparation Steps
1. Determine a migration window of time to re-install the existing servers and migrate the servers and users to MetaFrame XP. This window should have the least effect on production users.
2. Create new user groups to be applied to the published applications on the MetaFrame XP servers.
3. Open internal TCP ports 2512 and 2513 for IMA and Citrix Management Console communication respectively.
4. Install and configure the remote data store. Refer to the MetaFrame XP Administrator's Guide for additional instructions.
5. Install the first new MetaFrame XP server for the production environment. The automated scripted server installation is used for this process. The first server houses the data store, becomes the new dedicated data collector, and establishes the new zone.
6. Add the MetaFrame XP Universal Server License through the Citrix Management Console (add MetaFrame XP user connection migration licenses as each server is upgraded).

Migration Rollout Steps
1. During the migration window, re-install all of the existing MetaFrame 1.8 servers using the automated scripted server installation. Each new server joins the MetaFrame XP farm and is added to the zone that was created.
2. Create all published applications on the new server installations and apply the new user groups to the published applications.
3. Configure the farm to initially support ICA Client broadcast messages.
4. Configure the client auto-update feature to update all ICA Client files with the latest version.
5. Run the scripts created for migrating designated users. This moves users from the groups that are applied to the MetaFrame 1.8 published applications to the new groups that are applied to the MetaFrame XP published applications. This process should not affect the existing MetaFrame 1.8 farms.
6. Once all users have received the updated ICA Client, configure the farm to not accept client UDP broadcasts. Configure all ICA Clients to use TCP/IP + HTTP for connections.

Scenario #2
The second scenario describes migrating from multiple MetaFrame 1.8 farms that span multiple subnets to a single MetaFrame XP farm. The existing farm currently uses MetaFrame 1.8 with Feature Release 1 and Load Balancing Services. In this scenario existing server hardware will be used with the exception of the MetaFrame XP data collector, which will be implemented using new hardware.
Analysis

The analysis phase of the migration methodology includes project scope definition and an infrastructure assessment.

Project Scope

Project management has defined the success criteria to include the following:
- Successful implementation of the MetaFrame XP environment
- Minimal impact to the user community.

The main requirement of the project is for all MetaFrame 1.8 servers and applications to be migrated to MetaFrame XP.

Infrastructure Assessment

The following diagram depicts the existing infrastructure environment:

![Diagram of existing infrastructure](image)

Figure 9 Scenario 2 Existing Infrastructure

The infrastructure environment is described as follows:

**Farm Architecture** – The servers reside in two MetaFrame farms. All servers in both farms are running MetaFrame 1.8 with Feature Release 1.

**Master ICA Browser** – Master ICA Browsers exist in both farms. The master ICA Browser in both farms also hosts applications.

**Servers** – There are 6 servers in each farm, for a total of 12 servers. The farms are located in two separate physical locations.

**Load Balancing** – Default Load Balancing parameters are applied to both farms.

**Network Structure** – Two network subnets exist. A farm resides on each subnet with the following distribution of servers:
Subnet 1: 6 servers
Subnet 2: 6 servers
The two subnets are connected by a T1 link.

**Users** – There are 500 users that access the farms from multiple company sites.

**Applications** – There are 25 published applications.

**Application Delivery** – Users from both farms currently connect with NFuse to access published applications. There is a separate NFuse server for each farm in this scenario.

**Licensing** – Each server has a base MetaFrame 1.8 server license valid for 15 users (180 total user licenses). Additional user bump backs are included to support all 500 users (1 20-user bump pack, 6 50-user bump packs). A MetaFrame 1.8 Feature Release 1 license is also installed on each server.

**Design**

The design phase of the migration methodology determines the planned native MetaFrame XP architecture, the migration strategy, and the migration architecture.
Planned Native MetaFrame XP Architecture

Below is a diagram of the planned native MetaFrame XP architecture:

![Diagram of Planned Native MetaFrame XP Architecture](image)

**Figure 10 Scenario 2 Planned Native MetaFrame XP Architecture**

The planned native MetaFrame XP architecture is broken down as follows:

**Farm Architecture** – The native MetaFrame XP architecture will have only one farm.

**Load Management** – Default Load Management settings will be used.

**Data Store** – The data store will be located on a remote database for redundancy and scalability.

**Data Collector** – A dedicated data collector will be used; no applications will reside on this server.

**Zones** – Because the servers within the farm are not spread across low bandwidth connections, all servers will be contained within one zone.

**Servers** – Servers will remain in their existing locations.

**Applications** – Application installations will be re-configured and re-deployed. Therefore, no applications will be migrated. New applications will also be deployed.

**Application Delivery** – Users will continue to connect to the farm and published applications via NFuse.

**Licensing** – A Universal Server license for MetaFrame XPa is needed for all servers. This license covers MetaFrame XP and Load Management. In addition, connection licenses are needed for each user that connects to the farm. Ten 50-user bump packs are required for 500 users.
**Migration Strategy**

A new MetaFrame XP server farm with new server installations (Strategy #3) will be created. This option is chosen for the following reasons:

- The company wants to re-configure existing applications as well as deploy new applications.
- The company is deploying new server hardware. The company’s existing hardware is coming off lease and needs to be replaced. The new hardware will be phased in during the migration as the old hardware is phased out.
- The company prefers a “clean” system migration as well.
- The company is implementing a new architecture by moving from Windows NT 4.0 Terminal Server Edition to Windows 2000 and Terminal Services.
Migration Architecture

A diagram of the migration architecture is shown below:

This migration architecture is defined as follows:

**Farm Architecture** – Three farms will co-exist. The two MetaFrame 1.8 farms will run parallel with a new MetaFrame XP farm (operating in native mode).

**Load Balancing/Management** – Default Load Balancing/Management settings will be used in both the MetaFrame 1.8 farms and the MetaFrame XP farm.

**Data Store** – The data store will be located on a remote database for redundancy and scalability. MetaFrame XP currently supports the use of Microsoft SQL Server or Oracle for a remote database.

**Master ICA Browser** – The existing master ICA Browsers will remain during migration. These servers will be the last to be migrated.

**Data Collector** – A dedicated data collector will be used for the native MetaFrame XP farm as well as during the migration. No applications will be hosted from this server. This server will be a new installation.

**Zones** – All MetaFrame XP servers will be contained within one zone.

**Servers** – Servers will remain in their existing locations.

**Applications** – Application mixes will remain on their existing servers. New applications will be deployed to the appropriate servers.

**Application Delivery** – NFuse will be used to provide multi-farm consolidation. Users will be able to access applications within either the MetaFrame 1.8 farms or the MetaFrame XP farm. Users will be assigned permissions to published applications in either farm.
Although multi-farm consolidation is being used, migrated users will not be accessing both farms during the migration. This method is used to ease the migration by providing a single point of access during the migration period.

**Licensing** – The MetaFrame XP Universal Server License will be entered through the Citrix Management Console on the first server to be migrated. User connection migration licenses will be added as each server is migrated.

**Implementation**

During the implementation phase, scripts are developed to support the migration and rollout of MetaFrame XP. These scripts include the following:

- An automated scripted server installation is created. This is created with the use of VBScript and Windows Scripting Host and includes partitioning the disks on the new system, installing the operating system, installing MetaFrame XP, applying service packs, hot fixes, and registry tweaks, and installing applications. The installation of applications includes both the re-configured applications as well as the new applications.

- A script is created to migrate users to different groups. This script is used to move users from groups that have permissions for MetaFrame 1.8 published applications to groups that have permissions for MetaFrame XP published applications.

Additionally, a custom NFuse Web site that performs multi-farm consolidation is built.

**Readiness**

The readiness phase includes testing and a production pilot.

**Testing**

A testing environment is established for the migration. Testing will include two segments:

- **System Tests**: All components of both the native MetaFrame XP architecture and the migration architecture are tested.

- **Scalability Tests**: Scalability tests are conducted mainly for both the native MetaFrame XP architecture and the migration architecture. Different performance counters are monitored to test server performance and network performance.

**Pilot**

A pilot is performed before rollout of the migration architecture. The pilot is conducted using a small subset of users from the production environment. The pilot is run for user acceptance testing as well as to obtain business signoff for the architecture change.
Rollout

The following high-level steps are performed for the production rollout:

Migration Preparation Steps

1. Determine a migration window time frame to migrate users to MetaFrame XP. This window should have the least effect on production users.
2. Determine sets of users to be moved to MetaFrame XP during each migration window.
3. Create new user groups to be applied to the published applications on the MetaFrame XP servers.
4. Configure the NFuse Web site to support multiple farms. This is based on the Migration Architecture Design. Refer to Appendix A – NFuse Multi-Farm Consolidation at the end of this white paper for information regarding this type of configuration.
5. Configure the NFuse site for automatic download of the latest ICA Client.
6. Open internal TCP ports 2512 and 2513 for IMA and Citrix Management Console communication respectively.
7. Install and configure the remote data store (refer to the MetaFrame XP Administrator's Guide for additional instructions).
8. Install the first new MetaFrame XP server for the production environment. The automated scripted server installation is leveraged for this process. The first server will act as the new dedicated data collector and will establish the new zone. This server will be located at one of the site locations.
9. Install the remaining new servers for both locations using the automated scripted server installation. Each new server will join the MetaFrame XP farm as well as be added to the zone that was created.
10. Create all published applications on the new server installations and apply the new user groups to the published applications.
11. Add the MetaFrame XP Universal Server License through the Citrix Management Console (add MetaFrame XP user connection migration licenses as each server is upgraded).

Migration Rollout Steps

1. During the first scheduled migration window migrate the first set of designated users.
2. Run the scripts created for migrating designated users to move users from the groups that are applied to the MetaFrame 1.8 published applications to the new groups that are applied to the MetaFrame XP published applications. This process should not affect the existing MetaFrame 1.8 farms.
3. After this migration, direct the users to all of their applications in the MetaFrame XP farm via the NFuse Web site.
4. During the next scheduled migration windows the remaining users can be migrated following the same above steps.
Scenario #3

The third scenario describes an example of a migration from one MetaFrame 1.8 farm that spans multiple subnets to a single MetaFrame XP farm. The existing farm currently uses MetaFrame 1.8 with Feature Release 1 and Load Balancing Services. In this scenario existing server hardware will be used with the exception of the MetaFrame XP data collector, which will be implemented using new hardware.

Analysis

The analysis phase of the migration methodology includes project scope definition and an infrastructure assessment.

Project Scope

Project management has defined the success criteria to include the following:

- Successful implementation of the MetaFrame XP environment
- Minimal impact to the user community.

The main requirement of the project is for all MetaFrame 1.8 servers and applications to be migrated to MetaFrame XP.

Infrastructure Assessment

The following diagram depicts the existing infrastructure environment:
The infrastructure environment is described as follows:

**Farm Architecture** – All servers reside in one MetaFrame farm. All servers in the farm are running MetaFrame 1.8 with Feature Release 1 and Load Balancing Services.

**Master ICA Browser** – This farm has a dedicated master ICA Browser. Using Citrix Server Administration, the server is set to **always attempt to become master ICA Browser**.

**Servers** – There are 50 servers in the farm, all located in a central data center.

**Load Balancing** – Default load balancing parameters are applied.

**Network Structure** – Two network subnets exist with the following distribution of servers:

- Subnet 1: 25 servers
- Subnet 2: 25 servers

The two subnets reside on the same physical Local Area Network (LAN). The connectivity speed between subnets is 100 Mbps. An ICA Gateway exists between the two subnets allowing the farm to span the two subnets. ICA Gateways are bi-directional, meaning that they are initiated on servers on both network subnets.

**Users** – There are 1000 users that connect to published applications within the farm. Users access the farm from multiple company sites.

**Applications** – There are 25 published applications. Each server contains different application sets. There is a group of servers used only for office productivity applications and another group of servers used only for line-of-business applications.

**Application Delivery** – Users currently connect using the ICA Client. Each ICA Client’s networking properties for its application set is configured to use TCP/IP + HTTP. Given these client settings, there is no UDP Browsing from the ICA Client to the server. The ICA Client instead connects via TCP/IP + HTTP to the XML Service running on the MetaFrame server.
This setting is shown below:

![ICA Client TCP/IP + HTTP Setting](image)

**Figure 13 ICA Client TCP/IP + HTTP Setting**

**Licensing** – Each server has a base MetaFrame 1.8 server license valid for 15 users (750 total user licenses). Additional user bump backs are included to support all 1000 users (1 50-user bump pack, 10 20-user bump packs). A MetaFrame 1.8 Feature Release 1 license is also installed on each server.
Design
The design phase of the migration methodology determines the planned native MetaFrame XP architecture, the migration strategy, and the migration architecture.

Planned Native MetaFrame XP Architecture
Below is a diagram of the planned native MetaFrame XP architecture:

The planned native MetaFrame XP architecture is broken down as follows:

**Farm Architecture** – The native MetaFrame XP architecture will have only one farm.

**Load Management** – Default Load Management settings will be used.

**Data Store** – The data store will be located on a remote database for redundancy and scalability. MetaFrame XP supports the use of Microsoft SQL Server or Oracle for a remote database.

**Data Collector** – A dedicated data collector will be used. No applications will reside on this server.

**Zones** – Because servers within the farm are not spread across low bandwidth connections, all servers will be contained within one zone.

**Servers** – Servers will remain in their existing locations.

**Applications** – Application installations will remain on their existing servers.

**Application Delivery** – Users will no longer connect to the farm using the Program Neighborhood client. Instead, users will connect to published applications via NFuse. Connections to the NFuse Web server will be via HTTP.

**Licensing** – A Universal Server license for MetaFrame XPa is needed for all servers. This license covers MetaFrame XP and Load Management. In addition, connection licenses are needed for each user that connects to the farm. Twenty 50-user bump packs are required for 1000 users.
Migration Strategy

Existing MetaFrame 1.8 servers will be migrated to a new MetaFrame XP server farm (Strategy #2). This option is chosen for the following reasons:

- The company wants to keep their existing application deployments and migrate published applications.
- The company wants to use existing server installations during the migration.
- The company wants to use MetaFrame XP IMA communication and the full MetaFrame XP feature set.

Migration Architecture

A diagram of this migration architecture is shown below:

This migration architecture is defined as follows:

Farm Architecture – Two farms will co-exist during migration. The MetaFrame 1.8 farm will run parallel with a new MetaFrame XP farm (operating in native mode).

Load Balancing/Management – Default Load Balancing/Management settings will be used in both the MetaFrame 1.8 farm and the MetaFrame XP farm.

Data Store – The data store will be located on a remote database for redundancy and scalability.

Master ICA Browser – A dedicated master ICA Browser will exist for the MetaFrame 1.8 farm during the migration. This will be the last server that is migrated.

Data Collector – A dedicated data collector will be used for the native MetaFrame XP farm as well as during the migration. No applications will be hosted from this server. This server will be a new installation.
**Zones** – All MetaFrame XP servers will be contained within one zone.

**Servers** – Servers will remain in their existing locations.

**Applications** – Applications will remain on their existing servers.

**Application Delivery** – NFuse will be used to provide multi-farm consolidation. Users will be able to access applications within either the MetaFrame 1.8 farm or the MetaFrame XP farm. Users will connect to the NFuse Web server via HTTP. Users will be assigned permissions to published applications in either farm.

Although multi-farm consolidation is being used, migrated users will not be accessing both farms during the migration. This method is used to ease the migration by providing a single point of access during the migration period.

**Licensing** – The MetaFrame XP Universal Server License will be entered through the Citrix Management Console on the first server to be migrated. User connection migration licenses will be added as each server is migrated.

**Implementation**

During the implementation phase, scripts are developed to support the migration and rollout of MetaFrame XP. These scripts include the following:

- An automated scripted upgrade installation of MetaFrame XP. This is created with the use of VBScript and Windows Scripting Host and includes the installation of MetaFrame XP and the migration of existing published applications.

- A script to migrate users to different groups. This script is used to move users from groups that have permissions for MetaFrame 1.8 published applications to groups that have permissions for MetaFrame XP published applications.

Additionally a custom NFuse Web site that performs multi-farm consolidation is built.

End-user training materials are developed that explain how to access applications.

**Readiness**

As part of the readiness phase, testing is performed and a production pilot is established.

**Testing**

A testing environment is established for the migration. Testing will include two segments:

- **System Tests**: All components of both the native MetaFrame XP architecture and the migration architecture are tested.

- **Scalability Tests**: Scalability tests are conducted mainly for both the native MetaFrame XP architecture and the migration architecture. Different performance counters are monitored to test server performance and network performance.

**Pilot**

A pilot is performed before rollout of the migration architecture. The pilot is conducted using a small subset of users from the production environment. The pilot is run for user acceptance testing as well as to obtain business signoff for the architecture change.
Rollout

The following high-level steps are performed for the production rollout:

Migration Preparation Steps

1. Determine migration windows for performing server upgrades. These windows should have the least effect on production users.
2. Determine sets of users to be moved to MetaFrame XP during each migration window.
3. Create new user groups to be applied to the published applications on the MetaFrame XP servers.
4. Designate groups of servers to be upgraded during each migration window. These servers should be capable of supporting all published applications that are currently available within the MetaFrame 1.8 farm. This will allow a complete transition of users to the new MetaFrame XP farm. Once the users are migrated they will no longer access applications from the MetaFrame 1.8 farm.
5. Users are trained on new application access using the training materials developed during the implementation phase.
6. Install a Web server for use with Citrix NFuse. Install NFuse components on that Web server.
7. Configure the NFuse Web site to support multiple farms. This is based on the Migration Architecture Design. Refer to Appendix A – NFuse Multi-Farm Consolidation at the end of this white paper for information regarding this type of configuration.
8. Configure the NFuse site for automatic download of the latest ICA Client.
9. Open internal TCP ports 2512 and 2513 for IMA and Citrix Management Console communication respectively.
10. Install and configure the remote data store (refer to the MetaFrame XP Administrator's Guide for additional instructions).
11. Install and configure the new MetaFrame XP server that will be the dedicated data collector. The automated server installation script that was developed during the design and implementation phases will be used. This server will be configured to use the remote data store.
12. Configure the zone that will be used in the farm.
13. Add the MetaFrame XP Universal Server License through the Citrix Management Console (add MetaFrame XP user connection migration licenses as each server is upgraded).

Migration Rollout Steps

1. During the first scheduled migration window, upgrade the first set of designated servers. The automated scripted upgrade installation is used to facilitate this process. Each server will join the MetaFrame XP farm and belong to the zone that was created. These servers will all be configured to use the remote data store.
2. During each upgrade, migrate existing published applications. The migration log file in %SystemRoot%\system32 directory should be checked for any errors.
3. After the upgrade installation is complete, remove the group permissions that are applied to the migrated published applications. Create a new group and assign the group to the migrated published applications.
4. Run the scripts created for migrating designated users to move users from the groups that are applied to the MetaFrame 1.8 published applications to the new groups that are applied to the MetaFrame XP published applications. This process should not affect the existing MetaFrame 1.8 farms.

5. After this migration, direct the users to all of their applications in the MetaFrame XP farm via the NFuse Web site.

6. During the next scheduled migration windows, the remaining servers and users can be migrated following the same above steps. The master ICA Browser should be the last server that is migrated.
Appendix A – NFuse Multi-Farm Consolidation

Overview

NFuse is an application portal that allows you to integrate and publish interactive applications into any standard Web browser. NFuse is a three-tiered solution that includes a Citrix server component, a Web server component, and a Citrix ICA Client component with a Web browser.

NFuse can be used to deliver applications from more than one server farm, including farms running on various platforms. By using NFuse as a single point of access for applications, you allow users to reach applications spread across heterogeneous environments. This is increasingly important in organizations where server farms are divided according to operating system. Users can access applications published in farms running MetaFrame XP, MetaFrame 1.8, or MetaFrame for Unix.

Creating a Single Point of Access

You can use one of NFuse’s sample Web sites to provide access to applications from more than one farm. The Multiple Sever Farm sample Web site provides all the necessary functionality for contacting multiple Citrix server farms of any flavor and displaying their published applications on the same Web page. The Multiple Sever Farm sample Web site is supported in all four NFuse Web site models: Active Server Pages and HTML for IIS versions (on Internet Information Server) and JavaServer Page and HTML for Servlets versions (on Unix Web servers). NFuse’s “straight-from-the-box” solution is a starting point for creating a single point of access for different types of server farms.

To configure the sample Web site to contact more than one server farm you must identify the servers to be contacted. The ASP and JSP sites use a plain text file to store the list of Citrix servers. This file, multiServer.txt, is parsed by the Web page, which then displays applications from each server that is selected by the user. Enter in this file the names of the Citrix servers (one from each farm) and the port on each server that is running the XML service. The HTML pages require that the Citrix servers and ports be entered in the HTML code. See the NFuse Administrator’s Guide for detailed information on this process.

Customizing the Multiple Server Farm Site

NFuse Web sites can be created and customized in a variety of ways. You can start with one of NFuse’s sample Web sites that best suits your organization’s needs or use the Web Site Wizard to generate a basic NFuse site. You can then add to the existing code to implement the multiple server farm feature.

If you do not want to start with the code included in the sample Web sites or generated by the Web Site Wizard, you should understand how NFuse works to determine how to implement the multiple server farm feature in your own code.
The figure below illustrates how NFuse works:

1. A user connects to the Internet or Intranet and directs his Web browser to the URL of an NFuse Web site. The page that is served reads a list of Citrix servers that has been provided by the administrator. Users can be allowed to select which servers to connect to, or the selection can be automated to include all servers on the list. A login screen is displayed after the selection has been made.

2. The account authority or authorities verify the user’s credentials. Sites can be set up for multiple or single logins, depending on how many account authorities must be contacted to ensure authentication into each farm.

3. As the user’s credentials are verified, a second list of servers is created. This list eliminates any servers to which the user is not or cannot be authenticated. The appropriate errors are displayed and sufficient login attempts are allowed to generate this list.

4. For each server the user has permission to access, the server name and port are added to the list along with the credentials that were used to authenticate them to that particular farm. If a user’s authentication fails on a given server because of bad credentials the user is redirected to the login page. If it fails for any other reason,
the server’s information is deleted from the CitrixServer and CitrixPort cookies. If the authentication succeeds, the credentials are stored in the Nfuse_User, Nfuse_Domain and Nfuse_Password cookies.

5. After the server list is generated, each server is contacted. Saved credentials are used to retrieve application sets for each farm. The applications are enumerated and presented to the user.

**NFuse Java Objects, Substitution Tags, and Session Fields**

NFuse substitution tags, session fields, and NFuse objects perform the actions required in the multi-farm consolidation process described above. The NFuse substitution tags and session fields are used in template HTML and ICA files, providing an interface to the NFuse Java objects. The NFuse Java objects can be used to modify NFuse properties and, when used with HTTP cookies and URLs, create cross-page state in NFuse Web sites.

The NFuse objects are Java objects that can be accessed from Web server scripts or custom-written Java servlets to perform NFuse-related tasks. The NFuse Java objects authenticate users to a Citrix server farm, retrieve per-user application sets, modify properties of individual applications before presenting them to users, and parse template HTML and ICA files. These HTML and ICA files display application sets to users and provide them with links to initiate ICA sessions. Handling these objects in Web server scripts allows access to multiple servers.

The following list describes the NFuse Java objects that are required to authenticate the user and retrieve the applications:


**AppEnumerator** – A class that Web pages use to access applications published in a server farm. AppEnumerator returns App objects and Program Neighborhood folders that can then be used to manipulate applications. The AppEnumerator is returned by a CitrixWireGateway object.

**ClearTextCredentials** – Encapsulates user authentication information for presentation to the server farm. User credentials include the user's user name and Windows NT domain in plain text and a password encrypted using basic encryption. Use this object to package a user's credentials before sending those credentials via a CitrixWireGateway object to the server farm for authentication.

**CitrixWireGateway** – Creates a communication link between the Web page requesting a user's application information and the server farm containing the information. This object is used to create a communication channel through which a user's credentials are sent to a server farm and the application information for that user is returned.

There are other NFuse objects that can be used to add additional NFuse features and functionality. See the *NFuse Administrator’s Guide* for detailed information on how to use the NFuse Java Objects.