

Oracle® DIVAnet

Installation, Configuration, and Operations Guide

Release 2.2

E81070-02

August 2017

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Primary Author: Lou Bonaventura

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	ix
1 Introduction	
System Overview	1-1
API Support	1-1
User Interface (DIVAnetUI)	1-2
Command Line Interface (DIVAnetAdmin)	1-2
Access Rules	1-2
Other DIVArchive Components	1-2
2 Installation Planning	
Understanding Site Connectivity	2-1
DIVAnet Services	2-2
Enabling Sites for Remote Access	2-2
DIVAnet ManagerAdapter Service	2-3
DIVArchive	2-3
Objects and Instances	2-3
Source and Destinations	2-4
Setting Up Transfer Source/Destinations	2-4
Media (Storage Mediums) and Storage Plans	2-5
Drop Folder Monitor (DFM)	2-5
Configuring Local Client Access	2-6
DIVAnet ClientAdapter Service	2-6
DIVAnet DbSync Service	2-7
Display-Only Sites	2-7
DIVAnet Database	2-7
Object Cleanup	2-7
DIVAnet Version Compatibility	2-7
3 Installing DIVAnet	
Before Installing	3-1
Installing DIVAnet Software	3-2

DIVAnet Windows Installation.....	3-2
Oracle Linux DIVAnet Installation.....	3-2
Creating the DIVAnet Database.....	3-3
Creating the Tables	3-3
Create the DIVAnet Database Schema.....	3-3
Deleting the DIVAnet Database Schema	3-3
Oracle Connection String.....	3-4
Adding Sites.....	3-4
Configuring the DIVAnet Configuration Files	3-4
Installing and Starting DIVAnet Services.....	3-5
Installing DIVAnet Windows Services	3-5
Installing DIVAnet Linux Services	3-6
Managing DIVAnet Linux Services.....	3-6
Viewing Logs	3-6
Checking the Configuration.....	3-6
Syncing the DIVAnet Database	3-7
Upgrading DIVAnet	3-7

4 Configuring DIVAnet Services

Configuring the ClientAdapter Service.....	4-1
Configuring the DIVArchive Sitename.....	4-1
Configuring Client API Ports	4-1
Configuring Client Web Connections.....	4-2
Configuring Workflow Profiles	4-2
Profiles and API Ports	4-2
Retries and Timeouts.....	4-3
Valid Messages.....	4-3
Site-to-Site Mappings	4-3
Reloading Workflow Profiles.....	4-4
Modifying the ClientAdapter Configuration File	4-4
Top Level Parameters.....	4-4
API Ports Section.....	4-5
DIVArchive Managers Section.....	4-6
DIVAnet Database Section.....	4-7
Workflow Profile Section.....	4-7
Site-to-Site Transfer Mappings (Workflow Profile).....	4-9
Preferred Source/Destination Mappings	4-11
Configuring the ManagerAdapter Service	4-11
Sync Filtering By Category	4-12
Configuring Category Prefix Substitution.....	4-13
Modifying the ManagerAdapter Configuration File	4-14
Configuring the DbSync Service	4-16
Modifying the DbSync Configuration File	4-16
DivaManager Section.....	4-17
DIVAnet Database	4-17
Configuring Access Rules.....	4-17
Methods of Running Access Rules	4-17

Archive Example	4-18
Copy Example.....	4-18
Include and Exclude Rules	4-19
Attribute Types.....	4-19
Rules for DIVAnet Requests (ClientAdapter).....	4-19
Connect Operations	4-19
SubType (for Delete).....	4-20
Rules for DIVArchive Requests (ManagerAdapter)	4-20
Rule Matching.....	4-20
Rulesets and Defaults	4-21
Another Example	4-21
Ruleset Parameters.....	4-21
Include/Exclude Rule Parameters.....	4-22
Adding Script Variables to a Configuration File	4-24

5 Using the DIVAnet User Interface

Prerequisites	5-1
Starting the DIVAnetUI	5-1
Connecting to DIVAnet	5-2
Viewing Assets	5-2
Asset Details Panel.....	5-2
Searching Assets.....	5-3
Cutting and Pasting Asset Details	5-3
Viewing Requests	5-4
Viewing Request Progress (Live Updates).....	5-4
Viewing Request Details	5-4
Canceling Requests	5-4
Filtering Requests.....	5-4
Cutting and Pasting Request Details.....	5-5
Copying an Object	5-5
Deleting an Object	5-5
Restoring an Object	5-6
Modifying the Configuration File	5-6

6 DIVAnet Administration

Using the DIVAnetAdmin Utility	6-1
Request Processing and Queuing	6-3
DIVAnet Request Queuing.....	6-3
DIVAnet Request Throttling.....	6-3
DIVAnet Request Retries	6-3

7 API Support

DIVAnet API Version Compatibility.....	7-1
Operations on Archived Content	7-1
Archive Requests.....	7-3
Restore Requests.....	7-3

Restore Workflow	7-4
Sources and Destinations.....	7-4
Retries	7-5
Limitations	7-5
Oracle Partial File Restore Requests.....	7-5
Copy Requests	7-5
Site-to-Site Transfer Methods.....	7-6
Local Copies.....	7-7
Selected By DIVAnet (media of any)	7-7
Retries	7-7
Delete Requests.....	7-7
Site Delete (media of any).....	7-8
Last Instance	7-8
Retries and Cancels.....	7-9
Other Supported Commands	7-9
Unsupported Requests	7-11
Return Codes	7-11

8 Troubleshooting

A Sample Configuration Files

ClientAdapter Sample Configuration	A-1
ManagerAdapter Sample Configuration	A-2
DbSync Sample Configuration	A-3

Glossary

List of Tables

2-1	DIVAnet Services	2-2
3-1	Linux DIVAnet Services.....	3-6
4-1	ClientAdapter Top Level Parameters	4-4
4-2	API Port Parameters	4-5
4-3	DivaManagers Configuration Parameters.....	4-6
4-4	DIVAnetDatabase Parameters.....	4-7
4-5	WorkflowProfile Parameters.....	4-7
4-6	SitetoSiteTransfer Workflow Profile Parameters.....	4-10
4-7	Top Level ManagerAdapter Parameters	4-14
4-8	ManagerDatabase Parameters.....	4-15
4-9	ObjectFilter Parameters.....	4-15
4-10	DbSync Top Level Parameters	4-16
4-11	DivaManagers Parameters for DbSync.....	4-17
4-12	DIVAnetDatabase Parameters.....	4-17
4-13	Ruleset Parameters	4-22
4-14	Rule Operational Parameters (Include or Exclude)	4-23
4-15	Rule Request Parameters (Include or Exclude)	4-23
4-16	Rule Object Parameters (Include or Exclude)	4-24
6-1	DIVAnetAdmin Options.....	6-1
7-1	Supported DIVArchive Content Requests	7-2
7-2	DIVAnet Restore Methods.....	7-4
7-3	DIVAnet Copy Derived Parameters.....	7-6
7-4	Site-to-Site Copy Methods	7-6
7-5	Delete Types	7-8
7-6	Other Supported DIVArchive Commands Not Related to Content.....	7-9
8-1	DIVAnet Common Errors.....	8-1
8-2	DIVAnetUI Common Errors	8-2

Preface

This document describes how to install, configure, and administer Oracle DIVAnet 2.2.

Note: Before DIVAnet 2.0, DIVAnet was installed, revised, and released with Oracle DIVArchive. DIVAnet is now released and revised independently of DIVArchive. DIVAnet 2.2 is not directly compatible with Legacy DIVAnet 1.0 (in DIVArchive 6.x and 7.x), and is configured differently. It is therefore not a drop in replacement.

Audience

This book is intended for installation, administration, and operations personnel.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For additional information, see the Oracle DIVAnet Security Guide for this release in the Oracle DIVAnet documentation library, and the Oracle DIVArchive documentation set for your installed DIVArchive release located at

<https://docs.oracle.com/en/storage/#csm>.

Introduction

- System Overview
- API Support
- User Interface (DIVAnetUI)
- Command Line Interface (DIVAnetAdmin)
- Access Rules
- Other DIVArchive Components

System Overview

Oracle DIVAnet provides a unified view of archived digital assets across multiple, distributed DIVArchive systems and DIVAcloud. It facilitates the moving of content back and forth among DIVArchive sites, and from customer Source/Destination servers and disks. It performs its tasks for the purposes of disaster recovery, content distribution, access control, performance, and content availability.

By connecting to multiple DIVArchive sites, DIVAnet creates a virtual archive system that spans geographical locations. You can leverage disaster recovery sites to provide additional asset availability when outages occur, and improve archive and restore performance for some assets when systems run normally. You can use DIVA installations from other parts of your organization as strategic backup sites, content sharing partners, and participants in global business media workflows. DIVAnet eases the burden of keeping track of content that could exist in local DIVA sites, remote DIVA sites, and the cloud.

Applications that use the DIVArchive API can use it to connect directly to DIVAnet. This allows applications to store and access digital assets in different geographical locations. DIVAnet synchronizes asset information from each DIVA site, so that it always has an up-to-date asset inventory. DIVAnet uses this information to enforce naming uniqueness, and to select the best site for various requests (for example Restore and Copy requests). DIVAnet also provides access rules to limit the operations that users are allowed to perform.

API Support

DIVAnet allows applications written using the DIVA API to control multiple DIVArchive sites as if the sites were one large archive system. The DIVArchive API allows client applications to connect to an archive system and store content, copy content, delete content, or restore content (see the *Oracle DIVArchive C++ API Programmer's Guide* in the *Oracle DIVArchive Additional Features Documentation* library).

for more on the DIVArchive API). The API also supports retrieving of information about archived objects. DIVAnet implements support for a large subset of the DIVA API - that which is commonly used by popular applications such as Media Asset Management (MAM) applications.

For more information, see [Chapter 7](#).

User Interface (DIVAnetUI)

DIVAnetUI is an application that allows the user to monitor DIVAnet requests, and view, copy, restore, and delete DIVAnet Assets (DIVA archived objects) across multiple DIVArchive sites. All DIVAnet level requests can be monitored, whether issued through the API or through the UI itself. You can also view asset information for all configured DIVArchive sites, whether the asset was archived through DIVAnet or not. DIVAnetUI provides flexible ways of querying both request information and asset information.

DIVAnet 2.2 does not support the DIVArchive Control GUI (even in Direct Mode) - use the DIVAnetUI to monitor DIVAnet, and the DIVArchive Control GUI to monitor DIVArchive.

For more information, see [Chapter 5](#).

Command Line Interface (DIVAnetAdmin)

The DIVAnetAdmin command-line tool can perform various DIVAnet administrative functions including:

- Monitoring the status of database synchronization
- Setting site-specific parameters and limits
- Viewing API connections
- Reloading the workflow profiles and access rules
- Changing the log level of DIVAnet services.

For more information, see [Chapter 6](#).

Access Rules

DIVAnet allows rules to be created that govern which users and which Workflow Profiles are permitted to perform certain operations. For instance, you can create rules that limit which Source/Destinations can receive restored content, or limit which objects can be copied to sites.

For more information, see [Chapter 4](#).

Other DIVArchive Components

Some intersite transfer tasks are not specifically performed by DIVAnet. The Drop Folder Monitor (DFM) service monitors content that is being written to disk in a folder. It then archives that content into a DIVArchive system after the content has been completely transferred. DFM can be used as part of an intersite copy workflow, and can also be used with DIVAnet. See the *Oracle DIVArchive Drop Folder Monitor (DFM) User's Guide* in the *Oracle DIVArchive Additional Features Documentation* library for more information on DFM.

You can configure Oracle DIVArchive Storage Plan Manager (SPM) to automatically perform an operation when content arrives at a DIVArchive site. One of these operations is to restore the new content to a drop folder, which can in turn archive that content into another DIVA system. This allows a kind of replication of the content. See the *Oracle DIVArchive Storage Plan Manager (SPM) User's Guide* in the *Oracle DIVArchive Additional Features Documentation* library for more information on SPM.

Finally, DIVArchive itself provides the ability to transfer content from Source/Destinations to DIVA systems and back again, and copy and delete archived content.

DIVArchive, SPM, DFM, and DIVAnet work in unison to satisfy distributed media workflows.

Installation Planning

DIVANet is a distributed application that is commonly configured at multiple DIVA sites. This chapter describes the concepts necessary to determine which DIVANet services to install, and where. There are three major steps:

1. You must understand which sites must be connected to realize the desired workflows for a specific site. See [Understanding Site Connectivity](#).
2. You must enable remote access for each site in the system (or local access). See [Enabling Sites for Remote Access](#).
3. You must configure local client access on sites that have client applications that will connect to and use DIVANet workflows locally. See [Configuring Local Client Access](#).

Understanding Site Connectivity

A DIVANet site is defined as exactly one DIVArchive installation (which could exist in the cloud), and one or more DIVANet services. Each site is assigned a unique sitename. Each DIVANet service belongs to a particular site, indicated by the *LocalSitename* parameter in the DIVANet configuration files. You can configure Multiple DIVANet sites each with or without local client access. DIVANet sites can communicate with each other and replicate each other's information.

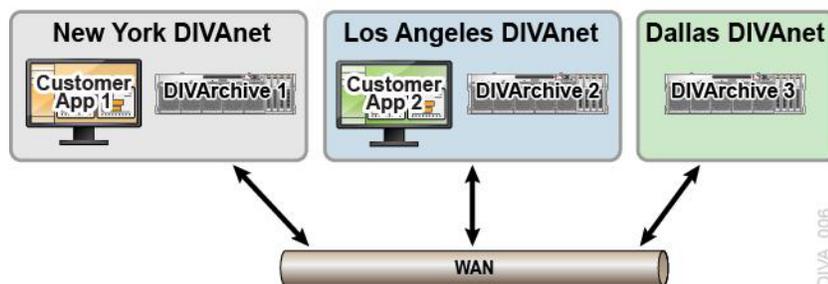
The most basic type of DIVANet connectivity is the use of DIVANet as a simple DIVArchive proxy for a single DIVArchive system. In this configuration DIVANet Direct Mode is used. You can configure Access Rules to permit or reject operations on a DIVA API connection. This mode does not provide a federated view of multiple sites, and cannot be used (for example) to copy between sites. For more information on setting up DIVANet Direct Mode, see [Configuring Client API Ports](#).

To truly address multiple DIVA sites as one large archive system, DIVANet sites must be connected together using DIVANet services. The remaining sections of this chapter describe how to configure DIVANet to achieve a federated view of archived content.

DIVANet can connect to remote sites to retrieve asset information, monitor the remote site's status, and send requests to the site (for example, a restore request), to satisfy DIVANet level requests. This rich interaction allows DIVANet to operate as one large archive system.

Note: Some DIVANet deployments will not require each site to be connected to all other sites in the network.

The following figure shows an example of a typical DIVAnet deployment with three sites: New York, Los Angeles, and Dallas. In this example, applications in New York can see and copy assets from Los Angeles and Dallas (and those existing in New York). In addition, applications in Los Angeles can see and copy assets from New York and Dallas. No customer applications are running at the Dallas site.



To realize this deployment, you will first configure a site for remote access. Dallas is perfect to demonstrate this scenario, as it does not have local clients to serve. You will then look at how to configure a site for client access, examining New York and Los Angeles, and how they interact.

DIVAnet Services

A DIVAnet service is either a Windows or Linux Service installed on a server responsible for carrying out computing tasks in a DIVAnet deployment. [Table 2–1](#) shows a summary of the available DIVAnet services.

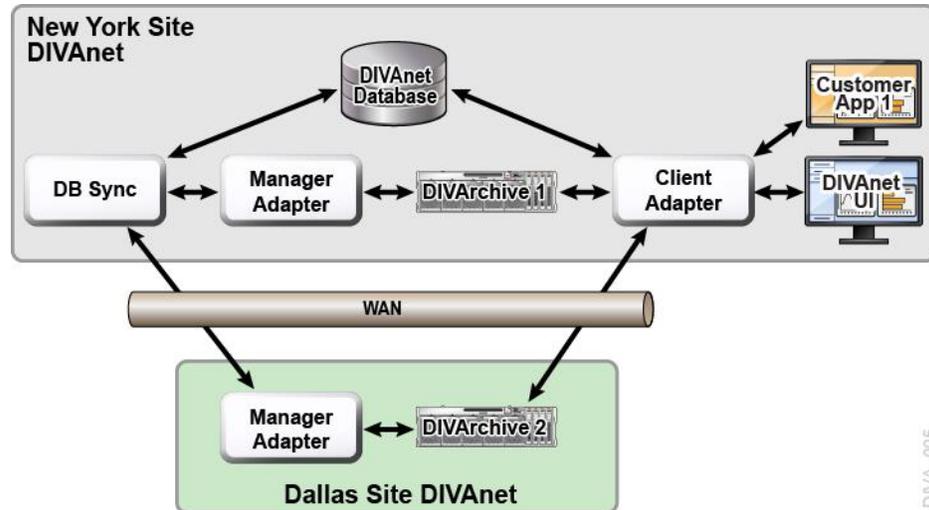
Table 2–1 *DIVAnet Services*

Service	Description
Client Adapter	The DIVAnet ClientAdapter service accepts requests from the DIVA API and web clients, and interacts with DIVArchive sites and the DIVAnet database to satisfy those requests. Configured when implementing local client (application) access. This can also be used in a minimal proxy-only DIVAnet deployment (DIVAnet Direct Mode, described in Configuring Client API Ports). For more information, see DIVAnet ClientAdapter Service .
Manager Adapter	The ManagerAdapter service serves as a bridge between DIVAnet and the Oracle DIVArchive Manager. Provides remote access for a DIVA site. Configured for all DIVAnet sites, especially those that will have asset information synchronized. For more information, see DIVAnet ManagerAdapter Service .
DB Sync	The DbSync service is responsible for synchronizing asset information from multiple DIVArchive sites, and storing the information in the DIVAnet database. Configured when implementing local client (application) access. For more information, see DIVAnet DbSync Service .

Enabling Sites for Remote Access

Enabling a DIVArchive site for remote access by other DIVAnet systems involves setting up a ManagerAdapter service on the site and configuring DIVArchive for remote access.

The following figure shows an example of two sites: a New York site with a full DIVAnet configuration (remote access and local client access) and a Dallas site that is configured just for remote access. The Dallas site only has one DIVAnet service running: the ManagerAdapter service. DIVArchive has been configured so it can interface well with other sites.



DIVAnet ManagerAdapter Service

The ManagerAdapter service serves as a bridge between DIVAnet and the DIVArchive Manager. It must be configured to provide remote access by other DIVAnet systems.

For security and performance reasons, Oracle recommends that the ManagerAdapter be installed on the same system as the DIVArchive Manager. Likewise, it is often the case that the ClientAdapter and DIVAnet database run together on a different server entirely.

The ManagerAdapter is configured using a simple configuration file. For more information, see [Chapter 4](#).

DIVArchive

Much of the configuration necessary to realize DIVAnet workflows is performed at each DIVArchive site. This section details some concepts needed to understand how DIVAnet interacts with DIVA, and the importance of the DIVA configuration. For the details on how to configure DIVArchive, refer to the *Oracle DIVArchive Installation and Configuration Guide*.

Objects and Instances

In a DIVArchive system, archived objects are uniquely identified by two parameters: an Object Name and an Object Category. The Category is part of the formal name of the object, a kind of namespace. For instance, an object with a name of *CLIP01* and category of *MOVIES* is a different object than one with a name of *CLIP01* and a category of *COMMERCIALS*.

DIVAnet uses the object name and category to associate objects on various sites.

Note: If an object on one site has the same object name and category as on another site, DIVAnet will consider both to be the same object.

When assets are archived using DIVAnet, DIVAnet will (by default) reject new assets that have the same name (and category) as assets already archived on other sites. However, archives issued directly to a DIVArchive system are not checked this way. Archiving without using DIVAnet may result in an object on site B that has different content than the corresponding object on site A. This in turn can lead to DIVAnet restoring the wrong content.

In DIVArchive, each archived object can contain many instances: one instance for each physical copy of the object on tape or on disk. There is an instance order number for each instance. The numbering starts with zero and increments by one for each instance in the object. So, you can uniquely refer to an instance on a DIVA system by providing the object name, category, and instance order number.

DIVAnet assigns its own set of instance order numbers that are derived from the DIVArchive instance order number. It does this so that for each object, the DIVAnet instance order numbers are unique across all DIVAnet sites.

Source and Destinations

A DIVArchive Source/Destination contains information necessary to communicate with a customer server or disk external to DIVArchive. Customers transfer content to and from DIVArchive through these servers and disks.

DIVAnet has an important convention concerning Source/Destination names.

Note: If a Source/Destination on one site has the same name as one on another site, DIVAnet concludes that they refer to the same physical server and (or) disk.

This convention is important in setting up a DIVAnet system (see [Restore Workflow](#) for more information). If the Source/Destinations are addressable through the API, and they point to the same physical server, disk, and path, you should give them the same name.

Setting Up Transfer Source/Destinations

To use DIVAnet to transfer content from one site to another, configure at least one Source/Destination to be accessible from both sites. This common Source/Destination will be used by DIVAnet to copy objects from one site to the other. The Source/Destination configurations on both sites should have the following characteristics:

- **Same Name** — On all sites, you should configure the same name for Source/Destinations that refer to the same physical server, disk, and directory.

DIVAnet's Site-To-Site Mappings can handle Source/Destinations that point to the same place but are not necessarily named the same. See [Site-to-Site Mappings](#) for more information.
- **Same Place** — The two Source/Destination entries must point to the exact same location (path) on the disk on a server. The types of transfer (for example, `FTP_STANDARD`, `DISK`) may differ on each site, and may even have different root paths in the configuration. For example, a Source/Destination named `NY_SHOWS` might be of type `DISK` on the New York site, but of type `FTP` on the Los Angeles site.
- **No Transcode or Renaming** — For Source/Destinations used in intersite copies, do not configure the Source/Destination for transcoding upon restore. This will lead to incorrect content being archived to DIVA sites.

- **Delete on Source** — For each Source/Destination that will be used in copy commands, set the **-allow_delete_on_source** option in the DIVArchive Source/Destination configuration. This allows content to be deleted from the site after it has been transferred to DIVA. You provide this option in the options field of DIVA's Source/Destination configuration panel.
- **AXF and Checksums** — You can enable end-to-end checksum comparisons on intersite copies (copy operations from one site to another) by enabling **AXF Genuine Checksums** in DIVArchive. In the DIVArchive Configuration Utility, select the Source/Destination that you use for copies, and then select the **AXF Genuine Checksum** option. After this is performed, you can set the **-axf** option in the DIVAnet Site-To-Site mapping *AdditionalOptions* parameter. This allows checksum information to be embedded in the AXF wrapper on the source site, and checked again at the target site.

Do not be confused by the *Site* parameter in the DIVArchive Configuration Utility's Source/Destination panel. The sitename here is used only by DIVA, and does not correspond to a DIVAnet site (see the *Oracle DIVArchive Installation and Configuration Guide* for more information).

Caution: Modifying the names of DIVArchive configuration parameters (such as *Source/Destinations*, *Media Names*, and *Storage Plans*) when connected to DIVAnet can cause errors.

Media (Storage Mediums) and Storage Plans

When DIVAnet copies objects from one DIVA system to another, care must be taken in assigning the archive *Media Name* and *Storage Plan Name* of the copy on the target site. Use a good naming policy regarding media values on each DIVA system.

DIVAnet records the DIVA Media Names when it synchronizes each object instance. You can configure DIVAnet to automatically assign the Media/Storage Plan on a Copy operation - see [Selected By DIVAnet \(media of any\)](#) for more information. One of the ways to configure this feature is to archive to the target site with the same Storage Plan Name as the source object. For this to work, the proper storage plans need to be configured in the target DIVA. Alternatively, you can use DIVA Media Mappings to transform the Storage Plan Name into a media or another storage plan, all at the target DIVA site.

Drop Folder Monitor (DFM)

DFM monitors folders for new content, and then archive the new content into DIVArchive. By restoring to a particular drop folder, DFM can pick up the content and archive it to a different DIVA system.

DIVAnet can implement copy workflows without DFM, but in some cases, it is needed or desirable. To copy without DFM in the mix, the DIVAnet RestoreAndArchive Transfer Method can be used. However, there are some situations where it is appropriate to use DFM. Good candidates for using DFM might include autonomous sites that want to perform their own cleanup of unsuccessfully transferred content, or systems where third party WAN accelerators are used. To use DFM for transfers, use the DIVAnet RestoreAndMonitor Site-To-Site Transfer Method. See [Site-to-Site Transfer Mappings \(Workflow Profile\)](#) for more information.

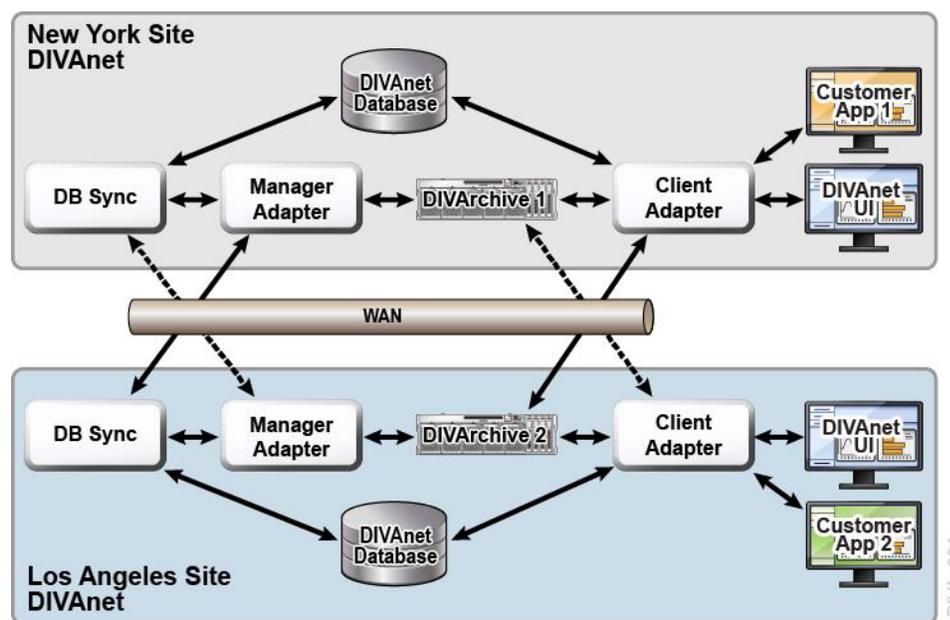
Configuring Local Client Access

Configuring for local Client Access involves configuring:

- The local DIVArchive for remote access (see [Enabling Sites for Remote Access](#))
- The ClientAdapter service
- A DbSync service
- A DIVAnet database

Configuring all DIVAnet services enables the site for full DIVAnet workflow processing.

In the following figure, both the New York and Los Angeles sites are configured for full DIVAnet workflow processing. Applications in Los Angeles connect directly to the ClientAdapter in LA. By doing so, they can retrieve content from New York if needed. The local DIVAnet database provides a global view of assets across sites, even if connectivity is lost from one site to another. If sufficient permissions are granted, DIVAnetUI users in LA can copy content from New York to LA, and even delete content in New York.



Though it is technically possible to configure Customer App 2 to remotely connect to New York's ClientAdapter, this configuration often provides better availability, security, and auditing. Performance and scalability is often increased as well, especially with unreliable or slow WAN links.

DIVAnet ClientAdapter Service

Application clients that want to use the DIVA API, or want to use the DIVAnet GUI, connect to the DIVAnet ClientAdapter service. This DIVAnet service accepts web and socket connections from applications and processes the requests. A ClientAdapter is configured on each site that has applications that are *local* to the site where DIVArchive and DIVAnet are installed. The ClientAdapter communicates with local and remote sites through the ManagerAdapter service. ClientAdapters can also connect directly to DIVArchive Managers using Socket Mode.

The ClientAdapter service is configured using one (or two) configuration files (see [Chapter 4](#) for more information).

DIVAnet DbSync Service

The DbSync service is responsible for synchronizing asset information from multiple DIVArchive sites, and storing the information in the DIVAnet database. DbSync communicates remotely with ManagerAdapter services on multiple sites to synchronize archived object information. DbSync is typically deployed along with the ClientAdapter. Both the DbSync service and ClientAdapter require direct access to the DIVAnet database.

The DbSync service is configured using a simple configuration file (see [Chapter 4](#) for more information).

Display-Only Sites

You can configure a site as display-only, meaning that asset information from that site will be synchronized, but no requests (or any other messaging) will be sent to the site. You configure the site (for example, site *diva4*) in the DbSync configuration file, but not in the ClientAdapter configuration. Site *diva4* will be effectively display-only. Asset information from this site will be queryable in the UI and in informational API calls, but requests sent to the site (using DIVAnet) will be rejected.

DIVAnet Database

Configuring DIVAnet local client access also involves setting up a DIVAnet database.

Object Cleanup

DIVAnet will sometimes satisfy a restore operation by temporarily copying an object from a remote site to the local site before restoring. This way, future restores of the content will be much faster. DIVAnet does not automatically delete the disk instance after the restore. Instead, it leaves the content in case others will want to restore it.

DIVArchive contains two tools that can automatically clean up content when a given disk/array becomes full:

- Oracle DIVArchive Storage Plan Manager (SPM) has a feature that can automatically clean up disk instances for an individual DIVA site.
- DIVArchive Local Delete can perform a similar function, but can optionally verify that an object also exists on other DIVA sites.

Because DIVArchive is configured to create a nearline disk instance by default, object cleanup will also possibly need to occur on a DIVA site configured for DIVAnet remote access only.

DIVAnet Version Compatibility

DIVAnet 2.2 will interoperate with DIVArchive 7.3.1 or later. Some features incorporated into future releases of DIVArchive may not be accessible to DIVAnet without upgrading DIVAnet to a later release.

The DIVAnet 2.2 ClientAdapter and DbSync services will interoperate with a DIVAnet 2.1 ManagerAdapter. However, a 2.1 DIVAnet UI instance will not interoperate with DIVAnet 2.2.

Installing DIVAnet

- Before Installing
- Installing DIVAnet Software
- Creating the DIVAnet Database
- Configuring the DIVAnet Configuration Files
- Installing and Starting DIVAnet Services
- Viewing Logs
- Checking the Configuration
- Syncing the DIVAnet Database
- Upgrading DIVAnet

Before Installing

- DIVAnet is a distributed application, meaning that it is often installed on multiple systems. Refer to [Chapter 2](#) to understand what systems and servers DIVAnet should be installed on, and what particular DIVAnet services should be installed.
- Each DIVArchive site connected to DIVAnet must be running DIVArchive 7.3.1 or later.
- DIVAnet can be installed on 64-bit Windows platforms, or 64-bit Oracle Linux platforms (release 7 or later). DIVAnet can be installed either in the same home directory as DIVArchive, or in its own directory.
- On Windows computers the DIVAnet installer requires administrator privileges. On Linux computers the DIVAnet installer requires the user named `diva`.
- By default, the DIVAnet installer handles the installation of the DIVAnet services files, the DIVAnetUI, and a compatible 64-bit Java 8 Java Runtime Environment (JRE). DIVAnet uses the installed JRE for execution. Installing the DIVAnet database requires an Oracle Database 11g release 2 or later (Oracle 12c recommended). Oracle SQLPlus (Oracle 11 or later client software) must be installed directly on the DIVAnet system.
- If you are reinstalling DIVAnet, make sure the DIVAnet components are not running.
- If a DIVAnet database is installed, you must run the `upgradeDb` script (see [Upgrading DIVAnet](#)) to upgrade the database to the latest release.

Installing DIVAnet Software

DIVAnet can be installed on either Windows or Oracle Linux 7 operating systems. The following subsections describe the installation on each platform.

DIVAnet Windows Installation

Use the following procedure to install DIVAnet in a Windows environment:

1. Launch the DIVAnet installer executable, `DIVAnet_V2.2.version.exe` using an account with administrator privileges.
2. Select the desired components, and then click **Next**.
 - Select *Java* if you have not previously installed in the target directory.
 - If you only want to install the DIVAnet user interface, deselect *DIVAnet Services*.
3. Specify a target installation directory. This will be the DIVAnet home directory. Then, click **Install**.

Note: You can use the DIVArchive installation directory as the DIVAnet home directory. However, Oracle recommends installing DIVAnet in its own directory.

Oracle Linux DIVAnet Installation

If you run on Linux, you must create a user named `diva` and a user named `divanetsvc`. The `divanetsvc` user must have permissions to write, read, and execute files owned by `diva`. The `diva` user must have `sudo` privileges for the Linux `systemctl` command. You must run the installation as the `diva` user, an admin user, or `root`. You must create the `diva` user before installing the software. An option is provided in the installer to create the `diva` and `divanetsvc` users before installation.

Use the following procedure to install DIVAnet in a Linux environment:

1. Open a terminal session and navigate to the parent directory for the DIVAnet release.
2. Place the DIVAnet installation package (for example, `DIVAnet-2.2-RELEASE.sh`) in the directory. The installer enables DIVAnet to be installed in a child subdirectory of this parent directory.
3. Start the DIVAnet installation using the following command:

```
sh ./DIVAnet-2.2-RELEASE.sh
```

The install wizard provides an option to create the prerequisite `diva` and `divanetsvc` Linux users before installing the software. These users do not need to be recreated if they already exist and are assigned the correct permissions.

The installation will not continue if the users do not exist. If the users already exist, or after you create them, select the option to install the DIVAnet software.

4. When prompted for the installation directory, press **Enter** to use the default subdirectory named `DIVAnet`, or specify a custom subdirectory name and then press **Enter** to create the installation directory.

The DIVAnet install process begins, and all files and folders are extracted into the specified target directory.

The following sections describe additional steps you may need to take to complete the installation. See [Chapter 2](#) for more information.

Creating the DIVAnet Database

To create the DIVAnet database, you first create the DIVAnet Oracle database user and tables, and then configure the DIVArchive sites.

Creating the Tables

This step involves creating the DIVAnet database tables that will exist on a running Oracle server. This step is required only for a full DIVAnet install (see [Chapter 2](#)).

Create the DIVAnet Database Schema

To run the SQL creation scripts, you must have SQLPlus (Oracle 11g or later client software) installed on the system you are installing DIVAnet on. Insure that the path to the sqlplus utility is in the `PATH` environment variable on your system.

1. Open a command line window or terminal shell. Change directory to the DIVAnet home directory (the directory where DIVAnet is installed).
2. Change directory to `Program/Database/divanet/Install`.
3. Run the following command:

Windows:

```
create_divanet_user.bat {sys_password} {user_name} {user_password}
{oracle_tns_name}
```

Linux:

```
create_divanet_user.sh {sys_password} {user_name} {user_password}
{oracle_tns_name}
```

In the previous command, *syspassword* is the Oracle system password, *user_name* is the Oracle user name/schema name that will contain the DIVAnet tables, and *oracle_tns_name* is the Oracle TNS Name for the DIVAnet database you want to use for installation. If there is no TNS Name defined for the database, you can refer to the following section to create a TNS Name real-time.

Deleting the DIVAnet Database Schema

To delete the DIVAnet database schema, execute the following steps.

Caution: This will delete **all** data in the DIVAnet database.

1. Change directory to `Program/Database/divanet/Install`.
2. Run the following command:

Windows:

```
delete_divanet_user.bat {sys_password} {user_name} {oracle_conn_string}
```

Linux:

```
delete_divanet_user.sh {sys_password} {user_name} {oracle_conn_string}
```

In the previous command, *syspassword* is the Oracle system password, *user_name* is the Oracle user name or schema name that will contain the DIVAnet tables, and *oracle_conn_string* is either the Oracle TNS Name for the DIVAnet database you want to use, or the Oracle Connection String (see the following section) for installation. If there is no TNS Name defined for the database, refer to the following section to create an Oracle connection string for use in the add and delete scripts.

Oracle Connection String

If you do not have a TNS Name defined for your database, you can supply an Oracle connection string. The format for this string is the following:

```
netaddress:port/remote_servicename
```

Where:

- *netaddress*: Network Address where the database runs
- *port*: Port where clients can connect
- *remote_servicename*: The Oracle remote service name. This is different from the service name used locally. The default for the DIVA Oracle database installation package is `lib5.world`. An example full connection string is `183.34.34.67:1521/lib5.world`.

Adding Sites

After the DIVAnet tables are created, you must configure the DIVA sites in the DIVAnet database. Run the `addSites` script to configure the sites in the database as follows:

1. Change to the DIVAnet home directory, and then navigate to the `Program/divanet/bin` subfolder.
2. Run the `addSites` script:

Windows: `addSites.bat`

Linux: `addSites.sh`

This script will walk you through adding sitename records and configuring sitenames, locations, and comments. The network connections to the sites are configured in the `ClientAdapter` configuration file (see [Chapter 4](#)). You should create sitenames that are easy to read and reflect either the physical location, or the function of the site.

Note: The sitenames you provide in this step will be used when configuring both local and remote DIVAnet services. Ensure the desired sitenames are consistent and agreed upon before running the `addSites` script. If you make a mistake while running the command, you can logically delete the site from within the `addSites` script. After the sitenames are in the database, DIVAnet installation can proceed.

Configuring the DIVAnet Configuration Files

Refer to [Chapter 2](#) to determine which DIVAnet services need to be configured, and then refer to [Chapter 4](#) for details on how to configure each service.

Underneath the DIVAnet home directory, in the `Program/conf/divanet/templates` subdirectory, there are template configuration files for each of the DIVAnet services that need to be configured. Each template file has a `.ini` extension. You should take each file and copy it to the parent directory (in this case, `Program/conf/divanet`). When you copy each file, make sure to remove the `.ini` extension.

At the top of each `.ini` file, there is documentation of the parameters available. For each DIVAnet site, configure a `ManagerAdapterConfig.xml`. For a full DIVAnet installation, you also configure the `ClientAdapterConfig.xml` and `DBSyncConfig.xml` files.

Many of the parameters can be left at their default value. You should not have to change any of the configuration files in the `Program/conf/divanet/wrapper` folder. You may optionally add information to the `divaEnv.conf` file (see [Adding Script Variables to a Configuration File](#) for more information).

Note: If after reading this documentation you are unclear on what parameters should be modified from their defaults, contact Oracle Support.

Installing and Starting DIVAnet Services

See [Chapter 2](#) to determine what services need to be installed and configured.

After the configuration files have been set up properly, install and start the DIVAnet services. Perform each step for every service to be installed at each DIVAnet site.

Installing DIVAnet Windows Services

Use the following procedure to install DIVAnet services in the windows environment:

1. Change to the DIVAnet home directory (where DIVAnet is installed).
2. Navigate to the `Program/divanet/bin` folder.
3. Run the `InstallDivanetService.bat` script, providing the wrapper configuration file for the service being installed. For example, to install the ManagerAdapter Service (required at all sites where a Manager to be connected to DIVAnet is running), run the following command:

```
InstallDivanetService.bat
../..../conf/divanet/wrapper/ManagerAdapterWrapper.conf
```

- For the ClientAdapter Service, use the same command and same path, but substitute `ClientAdapterWrapper.conf` as the file name.
 - For the DbSync Service, use the file name `DBSyncWrapper.conf`.
4. In the Windows Services window, you should see one or more of the following DIVAnet services:
 - Divanet ManagerAdapter
 - Divanet ClientAdapter
 - Divanet DbSync

The services can be started in any order, but keep in mind that after the DbSync Service and the ManagerAdapter Services are both started, the database syncing process will start automatically. Start the ClientAdapter Service to allow new DIVAnet connections and API processing.

Installing DIVAnet Linux Services

Use the following procedure to install DIVAnet services in the Linux environment:

1. Open a terminal session and navigate to the `Program/divanet/bin` directory.
2. Run the `divanetservice` script to install the desired service (DbSync, ClientAdapter, or ManagerAdapter). For example, to install the ClientAdapter, enter the following command:

```
./divanetservice install ClientAdapter
```

Managing DIVAnet Linux Services

After the Linux DIVAnet services are created you use the following commands to manage them:

Table 3–1 Linux DIVAnet Services

Operation	Command
List All Services	<code>./divanetservice list</code>
Start a Service	<code>./divanetservice start {service_name}</code>
Stop a Service	<code>./divanetservice stop {service_name}</code>
Restart a Service	<code>./divanetservice restart {service_name}</code>
Install a Service	<code>./divanetservice install {service_name}</code>
Uninstall a Service	<code>./divanetservice uninstall {service_name}</code>

The `service_name` parameter can be DbSync, ClientAdapter, or ManagerAdapter. The DIVAnet services can also be managed using the standard Linux system V `systemctl` command syntax. You can enter `man systemctl` at the Linux command prompt for more information on the `systemctl` command.

Viewing Logs

The DIVAnet applications produce two main types of logs:

- **Trace Logs** — Located within the DIVAnet home directory, in the `Program/log/divanet` subfolder. Each of the DIVAnet services (ManagerAdapter, DbSync, and ClientAdapter) create their own subfolder underneath the `Program/log/divanet` folder, and generate their log files in that subdirectory.
- **Wrapper Service Logs** — Examine these first to make sure the applications are starting correctly. These logs are within the DIVAnet home directory in the `Program/log/divanet` subfolder. They will be named similar to the `Wrapper.conf` file, but with a `.log` file name extension (for example: `ManagerAdapterWrapper.log`).

Checking the Configuration

The most common errors in configuring DIVAnet are errors in specifying the sitenames. All DIVA sites in the DIVAnet network must use the same set of sitenames. The sitenames must be consistent in all of the configuration files, and in the DIVAnet database. The `LocalSitename` parameter must be correctly configured in all of the service configuration files.

You should also check the DIVArchive Source/Destination. For example, by default, if site *diva1* has a Source/Destination that is named the same as one on site *diva2*, DIVAnet considers them as pointing to the same logical Source/Destination. If the same Source/Destination name is configured on two different DIVArchive sites, DIVAnet will assume that these Source/Destinations refer to the same place.

Syncing the DIVAnet Database

When the DbSync service is started, it will attempt to synchronize archived object information by consulting the ManagerAdapter service. The ManagerAdapter service in turn pulls asset information from the DIVArchive site to which it is connected. Therefore, you should start the DbSync service and ManagerAdapter services only when you are ready to begin synchronization.

Note: If you stop the DbSync service, synchronization will start again where it left off when the service is restarted.

The *DivanetAdmin* command-line tool monitors the status of the database synchronization on a site-by-site basis. If DIVAnet is performing an initial synchronization of a particular site, it will show the percentage complete of that initial sync operation. If the site is up-to-date, DivanetAdmin will indicate that the site is *In Sync*.

DivanetAdmin can also be used to resynchronize a site. This involves deleting all archived object records previously synchronized from the site, and resynchronizing them from the source. This should only be used in rare circumstances, such as when the source DIVA database has been unexpectedly restored from an out-of-date database backup.

You can run the `gather_schema_stats` script after an initial synchronization, or resynchronization, to immediately improve performance. See the following section, *Upgrading DIVAnet*, for more information.

Upgrading DIVAnet

It is recommended that you back up the existing DIVAnet home directory before attempting to upgrade to a new version of DIVAnet. You must also stop any running DIVAnet services or programs on the server before attempting an upgrade. Use the following procedure to upgrade the DIVAnet software:

1. Follow the instructions in the [Installing DIVAnet Software](#) section. This will install the latest DIVAnet software into the DIVAnet home directory.
2. You must run the DIVAnet database upgrade utility if you are upgrading a system that contains a DIVAnet database. To run the utility, change directory to `Program/divanet/bin` and run the following command:

Windows: `./upgradedb.bat`

Linux: `./upgradedb.sh`

The script will prompt for input about the DIVAnet database (including database user name and the network address of where the database resides). The utility will add or modify the required database objects. The script will simply return success if the database has already been upgraded to the latest release.

3. If you are upgrading a system containing a DIVAnet Database to DIVAnet 2.2, you must run the `gather_schema_stats` script. This script installs a weekly job to gather statistics for the database (if needed), and then immediately gathers current statistics. The script requires the Oracle `sys` password. To run the script, change directory to `Program/Database/divanet/Install`, and execute the following:

In Windows:

```
./gather_schema_stats.bat {divanetDbUser} {sysDbPassword}  
{oracleConnectionStr}
```

In Linux:

```
./gather_schema_stats.sh {divanetDbUser} {sysDbPassword} {oracleConnectionStr}
```

For example, in Linux execute:

```
./gather_schema_stats.sh divanet sysDbPass3 127.0.0.1:1521/lib5.world
```

Configuring DIVAnet Services

Before configuring DIVAnet services, see [Chapter 2](#) for a description of the DIVAnet services and how they interact with DIVArchive and each other.

- [Configuring the ClientAdapter Service](#)
- [Configuring the ManagerAdapter Service](#)
- [Configuring the DbSync Service](#)
- [Configuring Access Rules](#)
- [Adding Script Variables to a Configuration File](#)

Configuring the ClientAdapter Service

Configuring the ClientAdapter service involves configuring how clients connect to DIVAnet. It also involves configuring how DIVAnet connects to DIVArchive.

Configuring the DIVArchive Sitename

A DIVAnet site is defined to be one DIVArchive System, and one or more of the DIVAnet services (ManagerAdapter, ClientAdapter, DbSync). Each site is assigned a unique sitename. The site names must be configured in the DIVAnet Database (through the `addSites` utility) before configuring the ClientAdapter. You should create sitemames that are easy to read and reflect either the physical location, or the function of the site. These sitemames will be used on each site.

Note: Site names are case sensitive.

Configuring Client API Ports

DIVAnet allows configuration of the socket ports that client API applications use to connect to DIVAnet. The ClientAdapter allows multiple API ports to be configured. Each API port is configured in one of two modes: Direct mode or MultiDiva mode. Each port can be assigned a Workflow Profile.

- **Direct Mode** — In Direct mode, an incoming API connection is routed directly to an individual DIVArchive system with no added processing. The particular site that DIVAnet routes to is configured in the ClientAdapter on a per-port basis. This functionality allows local clients to connect to a remote DIVArchive system as if it were local. The DIVA site to route messages to must be configured in the `DivaManagers` section of the ClientAdapter configuration file. Note also that a site can be designated as operational in Direct Mode only (see [Configuring the DIVArchive Sitename](#) for more information).

Requests in Direct mode will not be displayed in DIVAnetUI, and can only be monitored at the DIVA site where the requests are being routed.

Some DIVAnet workflows do not require a DIVAnet database, DIVAnet UI, or MultiDiva mode processing. To configure a ClientAdapter service for Direct Mode only, insure that only Direct Mode ports are defined, and that a `WebServicePort` is not defined. This effectively disables use of the DIVAnet UI. An example configuration can be found in the file found here:

`Program/conf/divanet/templates/ClientAdapterConfig.xml.ProxyOnly.ini`

- **MultiDiva Mode** — In MultiDiva mode, DIVAnet makes all DIVA sites appear to be one big archive system. In this mode, an incoming API request is routed directly to DIVAnet. DIVAnet satisfies the high-level request by consulting other DIVArchive systems as needed. This functionality allows (for example) copying of content from one site to another, restores without needing to know which particular site has the content, and retries on other sites when content on the first site is inaccessible. In addition, it provides a global view of content on all sites.

The progress of a request executed in MultiDiva mode can be monitored in the DIVAnetUI. DIVAnet will create its own request-level events that inform the user of:

- How DIVAnet will process the request
- Requests that are made of DIVA sites
- Any errors or warnings encountered along the way

Configuring Client Web Connections

The DIVAnetUI application and the DivanetAdmin tool both make web connections to the ClientAdapter. You can configure the port that is made available for these connections in the ClientAdapter configuration file.

Configuring Workflow Profiles

A Workflow Profile is a set of parameters that defines how incoming requests will be processed by DIVAnet. Workflow Profiles provide a way to logically group users and systems that use DIVAnet in a similar manner.

DIVAnet allows one or more profiles to be created within the ClientAdapter. These profiles contain parameters that are needed by a particular set of users, or set of applications. Workflow Profiles are assigned based on where the request was received (the port where it was received - see the following section).

In the Workflow Profile you can customize information such as the list of accepted valid messages, retry parameters, site-to-site copy methods, and other attributes.

Profiles and API Ports

In the API Ports section, each defined port can be assigned a Workflow Profile Name. Requests received on the API port are processed using the assigned Workflow Profile. If no name is defined, the profile **default** is assumed. Ports in MultiDiva mode can have Workflow Profile Names that refer to a Workflow Profile body section further down in the ClientAdapter configuration.

If the port is in Direct mode, the Workflow Profile Name is a label only - there is no Workflow Profile body for Direct mode connections. This name, however, can be useful in Access Rules (see the following section).

In both MultiDiva mode and Direct mode, if you do not assign a name, the connection uses a Workflow Profile named **default**. Each workflow profile body section is configured in the `ClientAdapterConfig.xml`. The following sections list the types of information configurable in each Workflow Profile.

Retries and Timeouts

Some commands within DIVAnet can be retried (for example, Restore, Copy, and Delete requests). The parameters for the retries, including how long to retry and the retry interval are configurable here. Many commands have their own retry settings (for example, the Copy command retry parameters are completely distinct from the Delete command retry parameters). Message timeouts and connection limits can also be configured in the Workflow Profile. The retry and timeout settings are specific to the Workflow Profile in which they are defined.

Valid Messages

In each Workflow Profile, a list of messages is configured. This represents the list of valid API messages that can be accepted by this Workflow Profile. For example, if the Archive message does not appear in the list, then Archive messages cannot be sent to DIVAnet through the API (at least for this Workflow Profile).

Site-to-Site Mappings

DIVAnet provides a flexible way to configure how Site-to-Site transfers are performed. In each Workflow Profile, you configure mapping parameters for each transfer *path* (one path would be transfers from site A to site B, another would be from site A to site C, and so on). This table is consulted when Copy, Restore, or Partial File Restore commands are received.

It is useful to define the Site-To-Site mappings in the **default** Workflow Profile, as other Workflow Profiles can load their mappings from the **default** Workflow Profile. This can reduce the number of mappings in the configuration.

Parameters — DIVAnet supports several different transfer types. [Site-to-Site Transfer Methods](#) describes the different types. **RestoreAndArchive** is the default transfer type. **RestoreAndMonitor** requires Drop Folder Monitor (DFM), or another application that performs a similar function. Other parameters include:

- *Source/Destination* — the common storage location for transfer (accessible at both the source and target sites).
- *Default Media* — the default archive media to use when archiving on the target site. Default Media can be used when either a temporary copy was initiated (such as by a restore), or a user launched a copy and indicated that DIVAnet should select the media.
- *Options* — the parameters to use in the restore, archive, and transfer requests.
- *FilePathRoot* — the parent directory to store the content

Directory Location — DIVAnet constructs a directory path to store the files that is relative to the Source/Destination chosen. This relative path is constructed as follows:

```
{FilePathRoot} / {Media} / {UniqueDirName} /
```

The *FilePathRoot* is specified in the Site-To-Site mappings. Media will be embedded in the path only if the *AppendMediaToPath* option is set to true in the mappings (the default is **false**). Finally, DIVAnet generates a unique directory name that is included as part of the path. This unique name has the sitename that initiated the request prepended to it.

Reloading Workflow Profiles

Parameters specified in the Workflow Profile can be changed and reloaded without restarting the ClientAdapter. Restarting the ClientAdapter service should be avoided, because it will stop all executing requests and close all client connections. The reload can be performed within the DivanetAdmin tool (see the following for more information on DivanetAdmin).

Note: The reload updates not only all of the Workflow Profiles, but also all of the Access Rules.

Modifying the ClientAdapter Configuration File

The following tables describe the parameters that can appear in a ClientAdapter configuration file. The file is in XML format. The default value column indicates the value that the parameter will have if the parameter is not specified in the configuration file. The column also indicates whether the parameter is required or optional.

To create a new configuration file:

1. Navigate to the DIVAnet home directory (where DIVAnet is installed).
2. Navigate to the Program/conf/divanet/templates folder.
3. Copy the ClientAdapterConfig.xml.ini to the parent directory, but leave out the .ini extension (../ClientAdapterConfig.xml).
4. Using the tables below as a reference, edit the parameters in the ClientAdapterConfig.xml file to configure the ClientAdapter.

See [Appendix A](#) for a sample ClientAdapter configuration file.

Top Level Parameters

Table 4–1 ClientAdapter Top Level Parameters

Parameter	Description	Default Value
LocalSitename	The sitename of the Local DIVAnet site.	None (required)
LogDirectory	The folder where the log files will be produced.	The log/divanet/ClientAdapter directory
LogLevel	The detail level at which trace file logging takes place (ERROR, WARN, INFO, DEBUG, TRACE).	INFO
SyncTimeoutSecs	The number of seconds to wait for an object to synchronize.	60
WorkerThreads	The number of worker threads in DIVAnet thread pools. Used for tuning large configurations. If you are unsure, do not set this value.	25
AbortAllOnStartup	Stop all incomplete DIVAnet requests when DIVAnet starts - even if the request has completed at the DIVArchive level. New requests are not affected.	false
MaxClientConnections	The maximum total number of API connections allowed.	200

Table 4–1 (Cont.) ClientAdapter Top Level Parameters

Parameter	Description	Default Value
GlobalDivanetRequestLimit	The maximum number of pending or running DIVAnet requests that can be accepted into the system. When this limit is reached, DIVAnet will start rejecting new requests.	5000
InternalPollingRateMillis	The base rate at which sites are polled for information. Change this parameter only if slow networks and (or) systems are encountered.	4000
WebServicePort	The port used to send management messages to the ClientAdapter.	None (optional)
SSLWebServicePort	True if SSL should be enforced for web service requests.	true
WebDefaultWorkflowProfile	Workflow profile to be used for web requests (including DivanetUI).	The default profile
AccessRulesFilename	The access rules file name. This file name is relative to the directory where the client adapter configuration file exists.	None (if not configured, no access rules will be applied).

API Ports Section

Inside of the `ApiPorts` tag, multiple `ApiPort` definitions can appear. [Table 4–2](#) shows the parameters that can appear as part of an `ApiPort` definition.

Table 4–2 API Port Parameters

Parameter	Description	Default Value
ListenPort	This is the port socket to listen on.	None (required)
RoutingMode	This identifies how to route requests (Direct or MultiDiva). <ul style="list-style-type: none"> ▪ Direct — Route to one Manager only (the <code>Sitename</code> parameter is required in this case). ▪ MultiDiva — Route using DIVAnet Workflow Commands. Requests submitted will receive a unique Request ID by DIVAnet. 	MultiDiva
Sitename	The site to route to if using Direct Mode. Sites are defined in the <code>DivaManagers</code> section (covered in the following table). This parameter is only required in Direct Mode. If defined, it must match one of the sitenames defined in the <code>DivaManagers</code> section.	None (required in Direct Mode)
LocalAddress	The local address used to send to this Manager (typically which network card to use). Do not specify this if you're not sure.	None (optional)

Table 4–2 (Cont.) API Port Parameters

Parameter	Description	Default Value
WorkflowProfile	The Workflow Profile Name to use for requests traveling through this port (see the Workflow Profile section). If it is not supplied, the default Workflow Profile will be used.	default (the default profile in MultiDiva Mode).

DIVArchive Managers Section

Inside of the `DivaManagers` tag, multiple `DivaManager` definitions can appear. Table 4–3 shows the parameters that can appear as part of an `DivaManager` definition.

Table 4–3 DivaManagers Configuration Parameters

Parameter	Description	Default Value
Sitename	The name of the site where the Manager is installed. The sitename listed must match what was configured in the DIVAnet database and what is configured the <code>ManagerAdapter.xml</code> file.	None (required)
ConnectionType	This identifies how to connect to the Manager (valid values: Socket , WebService).	Socket
Address	The Network Address (IP or internet name) of the Manager	localhost
Port	The port where clients connect to the Manager.	None (required)
LocalAddress	The local address used to send to this Manager (typically which network card to use). Do not specify this if you're not sure.	None (optional)
LocalPort	The local port used. Do not specify this if you're not sure.	0
BaseURL	The url of the ManagerAdapter if the <code>ConnectionType</code> is set to WebService ; required .	None (optional)
TotalThrottleThreshold	DIVAnet will wait until the total number of Manager requests falls below this limit before sending more requests. This is used when the <code>ConnectionType</code> is set to Socket . When the Manager exceeds the number of running requests regardless of source (for example, DIVAnet, SPM, local API connection) DIVAnet will not send any more requests until the number of running requests on the Manager is below this threshold value.	400

Table 4–3 (Cont.) DivaManagers Configuration Parameters

Parameter	Description	Default Value
SubmittedThrottleThreshold	DIVAnet will wait until the number of requests DIVAnet itself has running on a Manager falls below this limit before sending more requests. When the Manager exceeds the number of running requests sent solely from DIVAnet itself, DIVAnet will not send any more requests until the number of running requests on the Manager sent only from DIVAnet is below this threshold value. Zero is a valid value, causing all messages to be queued internally within DIVAnet. The value -1 indicates there is no limit.	100

DIVAnet Database Section

Table 4–4 shows the parameters that can appear within the DivanetDatabase section.

Table 4–4 DIVAnetDatabase Parameters

Parameter	Description	Default Value
Address	The IP Address of the database.	localhost
Port	The port used to access the database.	1521
User	The user name of the schema.	None (required)
Password	The password of the schema.	None (required)
DbSiteId	The Oracle SID.	lib5
DbServiceName	The Oracle Service Name. Can be supplied instead of DbSiteId.	None (optional)

Workflow Profile Section

Table 4–5 shows the parameters that can appear within a WorkflowProfile section.

Table 4–5 WorkflowProfile Parameters

Parameter	Description	Default Value
Name	The name of the Workflow Profile	default
AllowDirectRemoteRestores	Allow direct transfers (to Source/Destinations) from remote DIVAs. Set this to false to always create a local copy of content before restoring.	true
MessageTimeoutMillis	The default timeout for messages sent to Managers.	15000 (15 seconds)
TotalRequestTimeoutHours	How long to keep requests before timing them out (in hours).	72

Table 4–5 (Cont.) WorkflowProfile Parameters

Parameter	Description	Default Value
PreventArchiveIfInDirectory	Prevent new Archive requests if the object exists on any site. If set to true, and the object exists on any site, requests to archive that object will be rejected (even if the object is not present on the site you are archiving to).	true
DeleteRetryIntervalMins	The interval between retries for the Delete Workflow	5 minutes
DeleteRetryLimitMins	The total number of minutes for which Delete Workflow retries will continue. Delete retries will be attempted every DeleteRetryIntervalMins, for DeleteRetryLimitMins, or until the delete is successful.	0 minutes (do not retry)
DeleteWaitTimeoutMins	The total number of minutes to wait for a delete operation to complete. Set this value low for real-time deletes, and higher for batch operations.	2 minutes
IntersiteCopyRetryIntervalMins	How long to wait before retrying a copy request. This parameter applies only for Copy requests.	5 minutes
IntersiteCopyRetryLimitMins	Keep retrying copies until this total time limit is reached (or they succeed). This parameter applies only for Copy requests.	0 minutes (do not retry)
RestoreRetryAttempts	In the case of failure, the maximum number of retries that should be attempted (commonly, retrying with content that exists on another site).	3 times
RestoreRetryIntervalMins	The interval between retries when DIVAnet performs a retry with the same site.	5 minutes
SiteDownRequeueWaitMins	(Advanced) Maximum time a site is down before routing queued requests to another site. Deletes are also governed by the DeleteWaitTimeoutMins parameter.	20
BackupArchiveSite	If the local site is down for an extended period (configured in SiteDownRequeueWaitMins), the site to use for archives instead of the local site. If the archive is submitted and fails on the local site, the archive will not be retried on the backup site.	None (no backup site)
ForceGlobalDeleteToSite	Converts a global delete request to a site delete at the specified site.	None (optional)
Messages	One or more Message Lists, each having a list of valid messages for the Workflow Profile.	NA

Table 4–5 (Cont.) WorkflowProfile Parameters

Parameter	Description	Default Value
Message	<p>One or more valid Message Names:</p> <ul style="list-style-type: none"> ■ AllInfo ■ Archive ■ Cancel ■ CloseObjectsList (legacy) ■ Copy ■ Delete ■ DeleteFile ■ GetArchiveSystemInfo ■ GetArrayList ■ GetFilesAndFolders ■ GetGroupsList ■ GetObjectsByFilename ■ GetObjectDetailsList ■ GetObjectInfo ■ GetObjectsList (legacy) ■ GetRequestInfo ■ GetSourceDestinationList ■ GetStoragePlanList ■ InitObjectsList (legacy) ■ PartialRestore ■ Restore <p>At least one Message tag must be specified.</p> <p>AllInfo allows all informational request messages to be sent.</p>	None (one required)
UseDefaultMappings	True if the Workflow Profile should include all of the mappings defined in the default Workflow Profile.	False

Site-to-Site Transfer Mappings (Workflow Profile)

The Mappings tag contains multiple Site-To-Site Transfer mappings. A Site-To-Site Mapping defines how an object is copied from one site to another. Each mapping contains FromSitename and ToSitename parameters. Each mapping defines how copies are performed from FromSitename to ToSitename.

Each mapping contains a Type parameter that indicates the method used to perform the transfers (see [Restore Workflow](#) for more information). The other parameters are defaults that are used in the process of executing the intersite copy.

DIVAnet will use the FromSrcDest as a temporary storage area, and will optionally push to the target DIVA using the ToSrcDest. When storing the content DIVAnet provides a unique folder name that is appended to the FilePathRoot. After storing the content in the FromSrcDest, DIVAnet will (based on the Type parameter) either:

- Archive the content into the target site
- Wait until the content is archived successfully at the target site

- Complete with no further action

Note: To avoid specifying the same mappings multiple times in the configuration, you can set the Workflow Profile parameter `UseDefaultMappings`. The Workflow Profile will use mappings from the **default** Workflow Profile.

Table 4–6 *SitetoSiteTransfer Workflow Profile Parameters*

Parameter	Description	Default Value
FromSitename	Origin sitename where objects are copied from. The value entered must match one of the sitenames defined in the DivaManagers section.	None (required)
ToSitename	Destination sitename where objects are copied to. The value entered must match one of the sitenames defined in the DivaManagers section.	None (required)
Type	The type of transfer: <ul style="list-style-type: none"> ■ Restore — Perform Restore and mark as transferred. ■ RestoreAndArchive — Restore then Archive to the Destination DIVArchive. ■ RestoreAndMonitor — Restore, then monitor destination (useful for DFM drop folders). 	RestoreAndArchive
FromSrcDest	The Source/Destination to use in the Restore step of the Copy. Note: Oracle recommends not using the default.	MISSING_MAPPING_TO + FromSitename
ToSrcDest	The Source/Destination to use in the Archive step of the Copy.	MISSING_MAPPING_TO + ToSitename
TempDefaultMedia	The target media to assign when making a temporary or transient copy of the object (performed on some Restore operations). The value is also used (depending on configuration) when API or UI users want DIVAnet to decide which media to use (the any keyword is used as the media). When RestoreAndMonitor is used with this variable, decide if the <code>AppendMediaToPath</code> option is needed.	None (required for RestoreAndArchive)
FilePathRoot	A path segment relative to the Source/Destination root. Prefixed to the unique folder name generated by DIVAnet.	Remote
AdditionalOptions	DIVA Options to use in the Restore/Archive operations.	-axf -rm -delete_fpr -allow_delete_on_source

Table 4–6 (Cont.) sitetoSiteTransfer Workflow Profile Parameters

Parameter	Description	Default Value
AssignDefaultMediaOption	<p>The strategy to use when API or UI users decide to let DIVAnet choose which media to use for copies. This is invoked when the any keyword is used as the media.</p> <p>StoragePlan — Use the Storage Plan name of the source object as the default media.</p> <p>StoragePlanAndSitename — Prepend Source Sitename to the Storage Plan (separated by an underscore).</p> <p>TempMedia — Use the value of TempDefaultMedia as the media.</p>	TempMedia
AppendMediaToPath	<p>True if the target media should be appended, as a subdirectory, after the FilePathRoot (and before the unique folder name). You would end up with the following:</p> <p>FilePathRoot / ToMedia / UniqueFolderName</p> <p>This option is useful when using the RestoreAndMonitor type with DFM, as DFM can parse the media name passed in this way.</p>	false
Weighting	<p>Rate this transfer path against other transfer paths based on performance, preference. Used in choosing sites for Copy and Restore. Valid range is 0-40. Be careful in using values above 20, as you will start to override other factors such as disk versus tape, site status, and so on. Misuse of this option can cause performance issues on Restore operations, and can contribute to WAN network congestion.</p>	Default is 10, local gets increased by 10.

Preferred Source/Destination Mappings

When DIVAnet calculates which site to use for Restore operations, DIVAnet typically prefers the local site, provided the Source/Destination is reachable using the local site. However, there are some cases where another site may be preferable.

The Mappings tag can contain a SrcDest tag. Within the SrcDest tag, a Name tag defines a Source/Destination name. The PreferredSitename tag indicates the preferred site to use when the Source/Destination is requested in a Restore operation. Multiple SrcDest sections can be present.

Configuring the ManagerAdapter Service

The ManagerAdapterConfig.xml configuration file contains the configuration for the ManagerAdapter Service. Use the following steps along with the description of each parameter in the ManagerAdapter configuration (each of the following tables has this information) to configure the ManagerAdapter Service.

Sync Filtering By Category

DIVAnet can place a filter on object information that is pulled by the DIVAnet DbSync service. This filter enables a site to select the subset of object records to be synchronized with downstream DIVAnet systems. The filtering is configured in, and performed by, the ManagerAdapter service.

Note: Object Filters and Category Prefix Substitution are advanced features that require careful testing to implement in production workflows. Do not add or change object filters indiscriminately.

For example, DIVAnet system in New York is configured to use and store assets from a site in Los Angeles. The Los Angeles site administrator wants to ensure that users in New York will only see objects that match one of three categories, AVID, POST1, and POST2. Implementing the following filter through the Los Angeles site's ManagerAdapter configuration file accomplishes this:

```
<LocalSitename>LosAngeles</LocalSitename>
<ObjFilter>
  <RequestingSitename>NewYork</RequestingSitename>
  <Category>AVID</Category>
  <Category>POST1</Category>
  <Category>POST2</Category>
</ObjFilter>
```

With this filter, objects matching the specified categories will be synchronized to the DIVAnet database in New York. Objects with other categories will not be synchronized. From the perspective of a New York site user, the only object records that exist on Los Angeles are the objects matching the configured category filter. Multiple ObjFilter tags may appear in the ManagerAdapter, each with a set of categories specific to a requesting site.

Note: Category Object Filters do not automatically prevent the ManagerAdapter from accepting requests for objects not matching the filter. ManagerAdapter Access Rules prevent operations on objects that do not have certain categories.

To prevent operations on objects not matching the filter, create the following access rule in the ManagerAdapter Access Rules file:

```
<Include>
  <SourceSitename>NewYork</SourceSitename>
  <Operation>*</Operation>
  <ReqObjectCategory>AVID</ReqObjectCategory>
  <ReqObjectCategory>POST1</ReqObjectCategory>
  <ReqObjectCategory>POST2</ReqObjectCategory>
</Include>
```

This rule will only permit requests from New York for objects that match one of three categories, AVID, POST1, and POST2. Other categories are denied if no other Include Rules are specified. If you define access rules in the ManagerAdapter, ensure that the ClientAdapter is configured to communicate with the site in *WebService* mode.

Configuring Category Prefix Substitution

There is a potential problem when using sync filtering by category. Continuing with the example, if New York creates an object with a category that is not present in the filter, and copies this object to Los Angeles, a naming conflict could occur. An object with that name could already exist on Los Angeles because the DIVAnet system in New York does not know about those objects. One solution is providing Access Rules in New York that limit the potential categories that can be archived.

You can achieve a more flexible solution using category prefix substitution. This feature not only filters the entries that are synced, but adds a category prefix to each incoming request. This provides a namespace-like feature for archived objects at a site.

In some DIVAnet workflows, a single site must accept objects copied from multiple sites. This can make it difficult to establish a set of unique categories for all objects in the system. Use category prefix substitution to address this situation. The following ManagerAdapter object filter will ensure that only objects in Los Angeles with categories beginning with NY001. are synchronized to New York's DIVAnet database.

```
<LocalSitename>LosAngeles</LocalSitename>
<ObjFilter>
  <RequestingSitename>NewYork</RequestingSitename>
  <CategoryPrefix>NY001.</CategoryPrefix>
</ObjFilter>
```

After applying the filter, but before the object reaches the destination (New York), the prefix is stripped – the remaining characters serve as the category in the New York DIVAnet database. For example, if the category of an object in Los Angeles is NY001.POST1, the resulting category sent to New York will be POST1. Similarly, any time the New York DIVAnet ClientAdapter sends commands to Los Angeles, the prefix is added back.

This allows Los Angeles to store copies of all of the objects from New York without naming conflicts. This technique allows Los Angeles to serve as a Disaster Recovery site for multiple sites. New York does not need to change its naming policy. New York references objects the same way that it always has, therefore no renaming on New York is necessary. Only one category prefix is allowed for each requesting site.

For this to work, the DIVAnet ClientAdapter service must be configured to connect to the remote site in *WebService* mode. Because objects are essentially renamed when copied to the site with a prefix substitution enabled, objects that were copied to the site previously will not have the prefix, which can present an issue. One solution involves providing a list of additional categories that are not translated. Another solution involves having Oracle DIVA installation specialists rename a subset of objects in the filtered site (that is, adding the category prefix to the category of each affected object in the DIVA database). If you use Category Prefix Substitution, you (will likely need to) disable Category checking in the DIVArchive Actor configuration (contact Oracle Support for instructions).

Note: If you change an Object Filter, it will almost always be necessary for the downstream DIVAnet system to perform a resync of the site. That system can accomplish this by using the DIVAnetAdmin tool (refer to [Chapter 6](#)).

The two types of category filtering can be mixed together. The following second filter (to Dallas) performs category prefix substitution (using DAL01) on all categories except

the categories POST2 or POST3. Only object records that have the category prefix or have a category of POST2 or POST3 will be synchronized to Dallas.

```
<LocalSitename>LosAngeles</LocalSitename>
<ObjFilter>
  <RequestingSitename>NewYork</RequestingSitename>
  <CategoryPrefix>NY001.</CategoryPrefix>
</ObjFilter>

<ObjFilter>
  <RequestingSitename>Dallas</RequestingSitename>
  <CategoryPrefix>DAL01.</CategoryPrefix>
  <Category>POST2</Category>
  <Category>POST3</Category>
</ObjFilter>
```

If you use this hybrid approach, ensure that object names with categories appearing on the list (for example POST2) are not added again with the prefix (for example NY001.POST2). This policy can be enforced through Access Rules.

Modifying the ManagerAdapter Configuration File

The following tables describe the parameters that can appear in a ManagerAdapter configuration file. The file is in XML format. The default value column indicates the value that the parameter will have if the parameter is not specified in the configuration file. The column also indicates whether the parameter is required or optional.

To create a new configuration file:

1. Navigate to the DIVAnet home directory (where DIVAnet is installed).
2. Navigate to the Program/conf/divanet/templates folder, copy the ManagerAdapterConfig.xml.ini to the parent directory, but leave out the .ini extension (../ManagerAdapterConfig.xml).
3. Using the tables below as a reference, edit the parameters in the ManagerAdapterConfig.xml file to configure the ManagerAdapter.

A sample ManagerAdapter configuration file appears in [Appendix A](#).

Table 4–7 Top Level ManagerAdapter Parameters

Parameter	Description	Default Value
LocalSitename	The name of the local site. The sitename listed must match what has been configured in the DIVAnet databases (local and remote), and must match what was configured in both the ClientAdapterConfig.xml and DBSyncConfig.xml files. This configuration enables the ClientAdapter and DbSync to communicate with the ManagerAdapter.	None (required)
ManagerAddress	The network address (IP or internet name) of the DIVArchive Manager.	localhost
ManagerPort	The port where clients connect to the DIVArchive Manager.	None (required)
WebServicePort	The port used to receive web messages.	None (optional)

Table 4–7 (Cont.) Top Level ManagerAdapter Parameters

Parameter	Description	Default Value
SSLWebServicePort	True if SSL should be enforced on incoming web service connections.	true
AccessRulesFilename	The access rules file name. This file name is relative to the directory where the ManagerAdapter configuration file exists.	None (if not configured, no access rules will be applied)
WorkerThreads	The number of worker threads in DIVAnet thread pools. Used for tuning large configurations. If you are unsure do not set this value.	50
LogDirectory	The folder where the log files will be produced.	The log/divanet/ManagerAdapter folder.
LogLevel	The detail level at which trace file logging takes place (ERROR, WARN, INFO, DEBUG, TRACE).	INFO

Table 4–8 ManagerDatabase Parameters

Parameter	Description	Default Value
Address	The IP Address of the DIVArchive Manager Database.	localhost
Port	The port used to access the database.	1521
User	The user name of the schema.	None (required)
Password	The password of the schema.	None (required)
DbSiteId	The Oracle SID	lib5
DbServiceName	The Oracle Service Name. Can be supplied instead of DbSiteId.	None (optional)

An `ObjFilter` can be defined for each requesting sitename in the ManagerAdapter. The following table shows the valid parameters for the object filter:

Table 4–9 ObjectFilter Parameters

Parameter	Description	Default Value
RequestingSitename	The site name of the site requesting objects	None (required)
Category	Objects with the category provided will be synced to the requesting sitename. Multiple categories may appear.	None (optional if CategoryPrefix appears)
CategoryPrefix	The prefix will be prepended to every request received through the ManagerAdapter. Only objects with the category prefix will be synced to the requesting sitename.	None (optional if Category appears)

Configuring the DbSync Service

The `DBSyncConfig.xml` configuration file contains the configuration for the DbSync Service. Use the following steps along with the description of each parameter to configure the DbSync Service.

Insure that the DbSync service is running when using the ClientAdapter. If the DbSync service is not running, some requests that would ordinarily succeed may fail. For example, DIVAnet restores of newly archived objects may fail, even though new DIVAnet archive requests may succeed.

Modifying the DbSync Configuration File

The following tables describe the parameters that can appear in a DbSync configuration file. The file is in XML format. The default value column indicates the value that the parameter will have if the parameter is not specified in the configuration file. The column also indicates whether the parameter is required or optional.

To create a new configuration file:

1. Navigate to the DIVAnet home directory (where DIVAnet is installed).
2. Navigate to the `Program/conf/divanet/templates` folder, copy the `DBSyncConfig.xml.ini` to the parent directory, but leave out the `.ini` extension (`../DBSyncConfig.xml`).
3. Using the following Parameter Description Tables as a reference, edit the parameters in the `DBSyncConfig.xml` file to configure DbSync.

A sample DbSync configuration file appears in [Appendix A](#).

Table 4–10 DbSync Top Level Parameters

Parameter	Description	Default Value
LocalSitename	The sitename of the local DIVAnet site (where DbSync is running). The sitename listed must match what was configured in the DIVAnet database and what is configured in both the <code>ClientAdapterConfig.xml</code> and <code>ManagerAdapter.xml</code> . This configuration enables DbSync to communicate with the ManagerAdapter.	None (required)
LogDirectory	The folder where the log files will be produced.	The <code>log/divanet/Dbsync</code> folder.
LogLevel	The detail level at which trace file logging takes place (ERROR , WARN , INFO , DEBUG , TRACE).	INFO
InternalPollingRateMillis	The base rate at which sites are polled for information. Change this parameter only if slow networks and (or) systems are encountered.	2000
WebServicePort	The port used to send management messages to DbSync.	None (optional)
SSLWebServicePort	True if SSL should be enforced on incoming web service connections.	true

DivaManager Section

Inside of the `DivaManagers` tag, multiple `DivaManager` definitions can appear. [Table 4–11](#) shows the parameters that can appear as part of a `DivaManager` section.

Table 4–11 *DivaManagers Parameters for DbSync*

Parameter	Description	Default Value
BaseUrl	Url of the service on the DIVA Manager platform to use for syncing. By default, this corresponds to the network address of the remote <code>ManagerAdapter</code> , qualified by the <code>WebServicePort</code> used by the <code>ManagerAdapter</code> .	None (optional)
Sitename	The official name of the site to sync object information from. The sitename listed must match what was configured in the DIVAnet database and what is configured in both the <code>ClientAdapterConfig.xml</code> and <code>ManagerAdapter.xml</code> .	None (required)

DIVAnet Database

Configure the DIVAnet Database parameters as shown in [Table 4–12](#).

Table 4–12 *DIVAnetDatabase Parameters*

Parameter	Description	Default Value
Address	The IP Address of the database.	localhost
Port	The port used to access the database.	1521
User	The user name of the schema.	None (required)
Password	The password of the schema.	None (required)
DbSiteId	The Oracle SID (Site Identifier).	lib5
DbServiceName	The Oracle Service Name. Can be supplied instead of <code>DbSiteId</code> .	None (optional)

Configuring Access Rules

In DIVAnet, you use Access Rules to control access that client applications and users have to DIVAnet operations and resources. Access rules can be run in three ways:

- On DIVAnet requests within the `ClientAdapter` service (in MultiDiva Mode)
- On DIVArchive requests within the `ManagerAdapter` service
- On DIVArchive requests that enter the `ClientAdapter` through a Direct Mode port

To run Access Rules, you must define the `AccessRulesFilename` parameter in the `ClientAdapter` and (or) `ManagerAdapter` configuration file. You must supply the file name without the path - DIVAnet assumes that the file will be located in the same directory as the `ClientAdapter` configuration file.

Methods of Running Access Rules

Access rulesets defined in the `ClientAdapter` (MultiDiva Mode) enforce access control on DIVAnet requests (received locally). Access rulesets defined in the

ManagerAdapter configuration enforce access control on DIVArchive requests (submitted to satisfy a DIVAnet request). Two levels of access control allows service-level rules to be configured where requests originate, and site-specific rules to be enforced as exceptions to the service-level policies.

Access rulesets defined in the ClientAdapter (Direct Mode) enforce access control on either DIVAnet requests or DIVArchive requests, depending on whether the remote system is another DIVAnet instance, or a DIVArchive system. In this mode, additional operation types are available for use in rulesets. These operations correspond to DIVArchive-specific requests, and are detailed below.

Archive Example

Let's quickly jump into an example to get more familiar with the rules themselves. The following rule allows Archive operations for users connecting as *admin* or as *operator* from either Source/Destination *DATA_EXP_PDAT1* or *VID_FTP_3*, and, archiving to a media of either *HDFeatures* or *spm* (you will see that the order of the attributes is not significant), and being named with a category containing the word *POST*.

```
<Include>
  <Operation>Archive</Operation>
  <Username>admin</Username>
  <Username>operator</Username>
  <ReqMedia>spm</ReqMedia>
  <ReqObjectCategory>*POST*</ReqObjectCategory>
  <ReqSourceDest>DATA_EXP_PDAT1</ReqSourceDest>
  <ReqSourceDest>VID_FTP_3</ReqSourceDest>
  <ReqMedia>HDFeatures</ReqMedia>
</Include>
```

Copy Example

In the following two rules, guest users from the *GUI Workflow Profile* are not allowed to copy from *diva2* to *diva3*, or vice versa.

```
<Exclude>
  <WorkflowProfile>GUI</WorkflowProfile>
  <Username>guest</Username>
  <Operation>Copy</Operation>
  <SourceSitename>diva2</SourceSitename>
  <TargetSitename>diva3</TargetSitename>
</Exclude>
<Exclude>
  <Username>guest</Username>
  <WorkflowProfile>GUI</WorkflowProfile>
  <Operation>Copy</Operation>
  <SourceSitename>diva3</SourceSitename>
  <TargetSitename>diva2</TargetSitename>
</Exclude>
```

You used two rules here because you didn't want to explicitly restrict copy operations that occur within the same site. For instance, someone at site *diva2* may want to copy an object (using DIVAnet) to a new tape - in this case the source and target sitename are both *diva2*! If you had a single rule that contained all of the *SourceSitename* and *TargetSitename* attributes, you would exclude copies from *diva2* to *diva2* and from *diva3* to *diva3*!

You are still not done. The copy will not succeed unless you have at least one Include rule that matches.

```
<Include>
  <Operation>Copy</Operation>
  <WorkflowProfile>GUI</WorkflowProfile>
  <Username>guest</Username>
  <Operation>ApiConnect</Operation>
</Include>
```

In this case, a very general Include rule gives you what you want, the ability to copy everywhere except for *diva2* to *diva3* and vice versa. In fact, you didn't really need an Exclude rule at all. Sometimes, however, things are easier with Exclude rules. Keep in mind that if an Exclude rule matches an operation, that operation will be denied, even if one or more Include rules match.

Include and Exclude Rules

To summarize, there are two types of rules, Include and Exclude. Access is denied for all requests unless at least one Include rule matches the operation about to be performed. However, if any Exclude rule matches, the operation is automatically rejected, regardless of any Include rules that match.

Attribute Types

On requests such as Delete, Copy, Restore, Partial File Restore, Cancel, and Archive, DIVAnet runs the entire set of Access Rules set to see if the operation is permitted. It examines variables such as:

- **Originator attributes** — the Workflow Profile of the connection, the user name that sent the message, the IP address of the originator.
- **Request attributes** — the Source/Destination, the source/target sitenames, the requested media, comments, and so on. These are derived from the request itself. Many of these are prefixed with Req.
- **Object attributes** - the media(s) the object is stored on, the storage plan, the object size, and so on. These are derived from the object being processed by an operation. Many of these are prefixed with Obj.

The following rule combines all three attribute types. It allows a user *diva* to perform a site delete on New York only when the object exists on Los Angeles.

```
<Include>
  <Username>diva</Username>
  <Operation>Delete</Operation>
  <SubType>SiteDelete</SubType>
  <TargetSitename>NewYork</TargetSitename>
  <ObjOnSite>LosAngeles</ObjOnSite>
</Include>
```

Rules for DIVAnet Requests (ClientAdapter)

DIVAnet requests are generated when requests are received in MultiDiva Mode. Access rules can be created against these DIVAnet operations. The following are details of some attributes specific to DIVAnet requests.

Connect Operations

The ApiConnect and WebConnect operations are special operations that must be included to establish a connection to the ClientAdapter.

- **ApiConnect** — This operation governs the ability to connect to the ClientAdapter through an API client socket connection. This is required for DIVA API connections.
- **WebConnect** — This operation governs the ability for applications to connect through web connections (DIVAnetUI and DivanetAdmin). This is required for DIVAnetUI connections.

When these operations are matched against your rules, keep in mind that only originator attributes will be present to match against. For example, `TargetSitename` would not be matched when access rules are run during the ApiConnect operation, because the attribute is simply not present when a client connects.

SubType (for Delete)

The delete operation has a `SubType` field which represents a subtype of the operation. You can include the `SubType` field in rules along with the Delete operation, supplying multiple `SubType` parameters if you want. The values for Delete `SubType` are:

- **GlobalDelete** — Matches if the delete operation being performed is a global delete of an object on all sites. This also matches a site delete operation that just happens to be deleting all remaining objects in DIVAnet.
- **SiteDelete** — Matches if the delete operation is a delete of all instances on a particular site (the name of the site can be matched in rules using the `TargetSitename` parameter). In addition, an operation will have this `SubType` if the requester is deleting a single instance, but that is the last instance of the object on that site.
- **InstanceDelete** — This delete operation deletes a single instance on a site, and there are other instances on the site.

Specifying this parameter in rules is useful for enforcing the scope of the permitted delete operations.

Rules for DIVArchive Requests (ManagerAdapter)

DIVAnet allows access rules to be run on DIVArchive requests as well. Access rules defined in the ManagerAdapter specify which DIVArchive operations (sent to satisfy DIVAnet requests) are permitted. Only the operations valid for DIVAnet requests can be specified in the rulesets. In ManagerAdapter rulesets, the `WorkflowProfile`, `TargetSitename`, and `SubType` attributes are invalid.

Similar to the ClientAdapter rulesets, the WebConnect Operation must be granted for connections to be established to the ManagerAdapter. This also extends to DbSync operations. The `SourceSitename` attribute corresponds to the specific site that is making a request. The ApiConnect operation is not available in ManagerAdapter Access Rules.

Rule Matching

Matching a rule involves comparing attributes in the rule against the actual values in each request, in a matching object, or request originator. Wildcards are allowed (use an asterisk (*) for wildcards). Case is insensitive for the tag names, but most values are case sensitive. The `Operation` tag is required in each rule. You can provide an `Operation` tag containing an asterisk (*) to indicate that the rule applies for all operations. However, you should be careful when doing this because not all attributes are valid for all operations.

Distinct attributes in a rule (an Include or Exclude rule) are joined together using a logical AND in the matching process. However, a single attribute specified more than one time in a rule causes those attributes to be joined together using a logical OR in a single expression.

When matching a requested operation against an access rule, DIVAnet determines if the attribute in the rule is applicable for the operation being performed. If not, the attribute is not used in the comparison.

Rulesets and Defaults

Rules can be grouped into Rulesets. Each Include-Exclude rule must be contained inside of `Ruleset` tags. Each incoming DIVAnet operation is matched against all rulesets. Rulesets are useful because they can have attributes that serve as defaults for all rules contained inside. In the matching process, each Ruleset default attribute is folded in to each child rule - as if the attribute were specified directly within the rule. It is common to use `Workflow Profile` as a default attribute for Rulesets, as `Workflow Profile` is populated on every requested DIVAnet operation.

The following attributes can be defaulted within a ruleset:

- User Name
- Net Address
- WorkflowProfile (in ClientAdapter rules)

Another Example

The following example allows users connected with a `WorkflowProfile` of `GUI` to view requests and assets, and to perform deletes of individual instances on site `diva1`. Remember that a `SubType` of `InstanceDelete` will deny any delete that would remove the last instance of a particular object on a site.

The second part of the example prohibits all connections, either web or API connections, that originate from the subnet 172.53. It will do this for any operation, regardless of its `Workflow Profile`.

```
<Ruleset>
  <WorkflowProfile>GUI</WorkflowProfile>
  <Include>
    <Operation>WebConnect</Operation>
    <Operation>Delete</Operation>
    <SubType>InstanceDelete</SubType>
    <TargetSitename>diva1</TargetSitename>
  </Include>
</Ruleset>
<Ruleset>
  <NetAddress>172.53*</NetAddress>
  <Exclude>
    <Operation>ApiConnect</Operation>
    <Operation>WebConnect</Operation>
  </Exclude>
</Ruleset>
```

Ruleset Parameters

Table 4–13 shows the parameters that can appear under the `Ruleset` section.

Table 4–13 Ruleset Parameters

Parameter	Description	Default Value
WorkflowProfile	This attribute, defined in the ClientAdapter configuration, is the name of a workgroup or set of applications that access DIVAnet. This attribute is made part of every rule in the ruleset.	None (optional)
Username	The user name passed in the API or specified in the web request. This attribute is made part of every rule in the ruleset.	None (optional)
NetAddress	The network address (IP or internet name) of the remote connection. This may be the address of a gateway or router and not the address of the originator. This attribute is made part of every rule in the ruleset.	None (optional)
Exclude	A rule that denies permission if it matches the DIVAnet operation.	None (optional)
Include	A rule that allows permission if it matches the DIVAnet operation.	None (optional)

Include/Exclude Rule Parameters

You can specify rule attributes that appear in requests. For example, `ReqMedia` will match the media and (or) storage plan specified in a request (operation).

Similarly, you can specify rule attributes that match against the archived object that the request is processing. For example, if a certain object is specified in a delete operation, `ObjHasMedia` will match any media that is currently part of that archived object regardless of media passed in the request.

[Table 4–14](#) shows the parameters that can appear in an `Include` or `Exclude` rule section.

Table 4–14 Rule Operational Parameters (Include or Exclude)

Parameter	Description	Default Value
Operation	<p>The name of the DIVAnet operation to match:</p> <ul style="list-style-type: none"> ▪ Archive ▪ Copy ▪ Restore ▪ PartialRestore ▪ Delete ▪ DeleteFile ▪ Cancel ▪ ApiConnect ▪ WebConnect ▪ ChangeConfig <p>There are additional operations available for DIVArchive operations in Direct Mode:</p> <ul style="list-style-type: none"> ▪ CopyToNew ▪ InsertTape ▪ EjectTape ▪ AssociativeCopy ▪ TranscodeArchived ▪ TransferFiles ▪ ServerDelete ▪ ChangePriority 	None (optional)
Username	The user name of the connected API user and (or) service user.	None (optional)
WorkflowProfile	The ClientAdapter profile name.	None (optional)
NetAddress	The IP address of the API application and (or) user.	None (optional)
SourceSitename	<p>The source sitename of the operation. Some operations do not have a source sitename (for example, Archive has a Source/Destination as the source). If the request is retried on another site, this value will change, and the entire rule will be reevaluated.</p> <p>When this attribute appears in ManagerAdapter rules, this matches the site that submitted the request.</p>	None (optional)
TargetSitename	The target sitename of the operation. Some operations do not have a target sitename (for example, Restore has a Source/Destination as the target, not a site). If the request is retried on another site, this value will change, and the entire rule will be reevaluated.	None (optional)
SubType	Multi DIVA Mode operation type (Delete command has GlobalDelete, SiteDelete, and InstanceDelete).	None (optional)

Table 4–15 Rule Request Parameters (Include or Exclude)

Parameter	Description	Default Value
ReqObjectName	The name of the object that is being processed.	None (optional)

Table 4–15 (Cont.) Rule Request Parameters (Include or Exclude)

Parameter	Description	Default Value
ReqObjectCategory	The category of the object (in DIVArchive this is part of the formal name of the object).	None (optional)
ReqSourceDest	The Source/Destination specified in the request.	None (optional)
ReqComments	The comments field in the request.	None (optional)
ReqMedia	The media that was requested as part of the operation/request (sitename should not be appended). Keep in mind that <i>Storage Plan</i> can be passed as the requested media.	None (optional)
ReqOptions	The options field in the request.	None (optional)

Table 4–16 Rule Object Parameters (Include or Exclude)

Parameter	Description	Default Value
ObjOnSite	Matches if the object exists on the specified site.	None (optional)
ObjNotOnSite	Matches if the object is NOT on the specified site.	None (optional)
ObjHasMedia	Matches any media on any site.	None (optional)
ObjHasStoragePlan	Matches storage plan on any site (ObjStoragePlanSite can restrict to one site).	None (optional)
ObjStoragePlanSite	Restricts specified storage plan to a specific site.	None (optional)
ObjHasSizeGbLessThan	Matches if total object size in fractional GBs is less than the specified value.	None (optional)
ObjHasSizeGbGreaterThan	Matches if total object size in fractional GBs is greater than the specified value.	None (optional)

Adding Script Variables to a Configuration File

DIVAnet allows variable substitution in configuration files to make configuring multiple script files easier. When the pattern `${variable_name}` is encountered in an XML Value (variables are not valid in Tag Names), the value of the variable is substituted when the DIVAnet reads the script. The value can be taken from an environment variable or assigned directly within the script.

If the Variable Name in the script is named the same as an Environment Variable, the value of that variable will be substituted in the script. Alternatively, you can define variable values at the top of each script using the `Variable` tag. The syntax is: `<Variable name="LocalSitename" value="diva1"/>`. When DIVAnet reads a configuration script, it checks to see if any variables that are used within the script have been set at the top. If not, it then looks for an Environment Variable with the same name as the variable.

For ease of use, a script file named `divanetEnv.conf` has been created within the DIVAnet home directory, in the `Program\conf\divanet\wrapper` subfolder. Variables can be centrally defined in this configuration file and will be available to all of the

DIVAnet services. If the values are changed and the service is restarted, the changes will be reread. Refer to the `divanetEnv.conf` file for sample variables.

Using the DIVAnet User Interface

- Prerequisites
- Starting the DIVAnetUI
- Connecting to DIVAnet
- Viewing Assets
- Viewing Requests
- Copying an Object
- Deleting an Object
- Restoring an Object
- Modifying the Configuration File

Prerequisites

To use the DIVAnet application:

- You will need to have the ClientAdapter configured and running on the server side.
- If you want to perform intersite copies within DIVAnetUI, ensure that the ClientAdapter Site-To-Site mappings are configured.
- To view the most up-to-date asset information, you will want to configure and run the DbSync service.

Note: DIVAnet 2.x no longer supports the DIVArchive Control GUI — use the DIVAnetUI instead. DIVAnet 2.x does not forward or support Control GUI messages in either MultiDiva or Direct Mode.

Starting the DIVAnetUI

To start the DIVAnetUI application in Windows, click the provided shortcut link. This link is located in the DIVAnet home directory, in the `Program/divanet/bin` subfolder. This shortcut can be copied to the Windows Desktop or Windows Start menu.

Open a terminal window and call the `DIVAnetUI.sh` script to start the DIVAnetUI in Linux.

Refer to [Chapter 3](#) for details on how to install the DIVAnetUI.

Connecting to DIVAnet

After starting the DIVAnet application:

1. Click either the **Requests** or **Assets** tab to open a prompt for the DIVAnet server URL.
2. Enter the URL of the DIVAnet server, and then click **OK**.

By default, the URL corresponds to the host name of the ClientAdapter, followed by a colon, followed by the ClientAdapter WebServicePort. The value that you enter will be persisted in the DIVAnetUI properties file. Example:

```
https://127.0.0.1:9801
```

If a connection error occurs, a message will appear in the lower right corner of the screen. Clicking the **Change URL** button will prompt for a new URL.

Viewing Assets

To display a list of assets, click the **Assets** tab. The first 400 assets will be listed in the left pane. The asset name will be shown in blue, followed by the object category.

By default, assets are presented in Object Name order. Select an asset to view details displayed on the right pane.

Asset Details Panel

The Asset Details panel is divided into three sections: the Global Object Parameters at the top, the Sites Table in the middle, and the Asset Filenames table at the bottom. The following are descriptions of some parameters and sections on the Asset Details panel:

- **Object Parameters** — The parameters at the top of the Details panel include the total size of the object, the archive date, the number of copies (instances), and comments.
- **Last Verified Date** — This represents the last time that the object was checksum verified on tape, at any site. If no tape instances exist, only then will it return the last time the object was verified on disk. If the object was never verified on archived media, the field is blank.
- **Alerts** — Any object alerts will be displayed directly underneath the Comments field. For instance, if DIVAnet detects that two sites have objects with the same name, but different content, an alert will appear.
- **Active Requests** — On the Asset Details panel, the text in green (above the Sites Table) indicates any DIVAnet requests currently running on the selected object. It is useful to view this field before issuing copies or deletes. This value is refreshed only when you press the **Search** button (see [Searching Assets](#)).
- **Sites Table** — Each site that contains instances of the object is displayed in the site table. Each site can contain one or more object instances. Objects for each site have a Storage Plan and initial archive dates (which can be used in the Search panel).

A site icon resembling a house indicates that the site is the local site. A purple site icon indicates that the site's assets are display-only, and requests cannot be run on them. A site icon resembling a globe indicates that the site is not local.
- **Asset Filenames Table** — The file names that comprise the archived asset appear below the site table. The file names are organized into pages, and ordered according to the order that they were archived. The file names for complex archived objects cannot be displayed in this panel.

Searching Assets

DIVAnetUI contains powerful object search capabilities. Click the **Search** button on the left to open the Asset Search panel.

The Asset Search panel allows many ways to search for assets. Some parameters are selectable, meaning that you can select both an attribute and value. Each value entered poses an additional restriction on the assets returned. Wildcards are permitted in all search fields except the date and time fields.

The following is a list of some parameters used for searching assets:

- **Archive Date Search** — DIVAnet allows the user to search on the date that an object was archived to a DIVA site. A start and end date and time range can be specified that narrows the search to objects that have an archive date in range. Click the calendar icon to select a date, and then modify the time value in the text field to the desired time. An object may have multiple archive dates (one for each site), and this query searches for any that exist within the range.
- **Object Name / Category** — An asset name is composed of two fields, Object Name and Object Category. The Category is the second half of the formal name of an object.
- **Site Search** — Some the search terms are user-selectable, including site search parameters. You can search for objects that are either on a site, not on a site, or have a tape copy on a site. This is useful for determining, for example, which objects need to be copied to other sites.
- **Other Search Parameters** — You can also search on other parameters such as file name, object size, comments, and media.

You can select the same parameter (for example `media`) multiple times and provide different values. By default, these multiple values are joined with all other values in the query using a logical AND. For example, if you search on `Sitename=Site A` and `Sitename=Site B`, your search will return objects that are present on both *Site A* and *Site B*.

However, if you select the **OR Like Query Params** check box, the multiple values are joined together using a logical OR. For example, you can search for objects that have a media of `POST1 OR PLAYOUT` (or both).

- **Checksum Failures** — You can select to display Checksum Failures. This returns all assets that have encountered checksum failures as part of DIVArchive operations, such as **Restore** and **Copy**. The particular site that experienced the failure is marked with an alert icon in the site table. There are no query parameters for this search (use the default value of `true`).
- **Missed Deletes** — You can select to display Missed Deletes. This returns all assets where a delete was attempted, but could not be processed. The sites that experienced the missed delete are marked with a gray X in the site table. You provide a time duration in days, which is subtracted from the current date and used to filter the query to return only the most recent missed deletes.
- **Ordering** — You can sort the objects based on Object Name (the default), Category, Archive Date, and Object Size. You may see significantly slower query performance when using some optional sort methods.

Cutting and Pasting Asset Details

Many of the values in the Asset Details panel can be copied to the clipboard by double-clicking on the values. In the Asset Search panel, right-clicking on a text field

reveals a menu. This menu contains a paste option, which allows items from the clipboard to be pasted into the text field. Alternatively, you can use Ctrl-V to paste the information.

Viewing Requests

Click the **Requests** tab to display a table of DIVAnet requests in the left panel. This table is ordered by the date that each request was submitted. Each request in the table shows where content is moving from, and where it is moving to (if applicable). A label with an (SD) appended means that the content is moving from or to a Source/Destination. Otherwise, the content is moving from or to a DIVAnet site.

Viewing Request Progress (Live Updates)

Requests are enabled for live updates when DIVAnetUI starts. If the request that is queried is currently executing, the progress of the request will be automatically updated on the screen. Filters can be applied to reduce the number of requests that are dynamically updated on the screen. Live progress updates can be stopped by providing an end date and time in the Filter panel. See [Filtering Requests](#).

Viewing Request Details

You can view the details of the request by clicking a request in the table. The details include information such as the priority of the request, the quality of service, source and target sitenames, and media (if applicable). In addition, a list of request-level events are displayed. These events show the steps that DIVAnet takes to satisfy the request, and the errors encountered along the way.

Canceling Requests

Right-clicking a request in the request table will show the *Cancel* option. If a request is canceled, another request (a Cancel request) is generated that will show the status of that Cancel attempt.

Filtering Requests

DIVAnetUI contains powerful request filter capabilities. Click the **Filter** button on the left to open the Request Filter panel.

The Request Filter panel contains parameters used to filter the list of requests. Two of the parameters are user-selectable. Each value entered poses an additional restriction on the assets returned. Wildcards are permitted in all search fields except the date and time fields and *Request ID* field.

Note: Unlike the Asset Search panel, you cannot search on the same attribute more than once.

The following are some search terms that can be applied to the list of requests:

- **Request ID** — DIVAnet requests are uniquely identified by a Request ID. DIVAnet allows filtering on Request ID or the name of the object being processed.
- **Request Submission Date** — A start and end date and time range can be specified that returns only requests that have been submitted with the date range specified. Click the calendar icon to select a date. The current local (computer) time is

automatically populated in the text field. You can modify the time value in the text field to the desired time. If an end date and time is supplied, no status or progress updates will be provided on screen. Delete the end date and time to enable live updates again.

- **Request Type** — Filters can be created on the type of request. Each type can be individually enabled or disabled. In this way, filters can ensure (for example) that only Archives and Deletes appear in the list.
- **Request Status** — Filters can be created on the Request Status. In this way, filters can ensure (for example) that only requests that are either *Pending* or *Running* are displayed.
- **Other Search Parameters** — You can also filter on other parameters such as *Media*, *Source/Destination*, and *Comments*.

Cutting and Pasting Request Details

Many of the values in the Request Details panel can be copied to the clipboard by double-clicking on the values. In the Request Filter panel, right-clicking on a text field reveals a menu. This menu contains a Paste option, which allows items from the clipboard to be pasted into the text field. Alternatively you can use Ctrl-V to paste the information.

Copying an Object

DIVAnet can copy objects from one site to another, or it can copy the object to different archive media on the same site (by creating a new object instance).

1. Within the Asset Table, select the objects to copy.
2. Above the Asset Table, click the **Copy** button.
A dialog box will appear showing the selected objects.
3. Select a *Target Site*, *Media/Storage Plan*, and *Priority*.
 - If the object already exists on the site you are copying to, DIVAnet will create another instance with the archive media that you select. In this case, specifying a *Media/Storage Plan* will cause the copy request for that object to fail, as the object on that site already has a Storage Plan.
 - If you select *Selected By DIVAnet* for the *Media/Storage Plan* option, DIVAnet will choose an appropriate media for the copy. If the object already exists on the site you are copying to, DIVAnet will not create another instance on that site.
4. Click **Submit**.
5. Click either **Monitor Requests** or **Close**.

If you choose **Monitor Requests**, you will be taken to the Requests View where you can monitor the DIVAnet Copy requests for each object that you selected.

Deleting an Object

You can delete an object from a given site or from all sites.

1. Within the Asset Table, select the objects to delete.
2. Above the Asset Table, click **Delete**.

A dialog box will appear showing the selected objects.

3. Select a *Target Site* and *Priority*.
You can select a specific site or all sites.
4. Click **Submit**.
5. Click either **Monitor Requests** or **Close**.

If you choose **Monitor Requests**, you will be taken to the Requests View where you can monitor the DIVAnet Delete requests for each object that you selected.

Restoring an Object

Objects can be restored to a particular *Source/Destination* using the following procedure:

1. Within the Asset Table, select the objects to restore.
2. Above the Asset Table, click **Restore**. A dialog box appears showing the selected objects.
3. Enter a **Source/Destination** and select a **Priority**.
4. Optionally, enter restore options in the *Options* field, and (or) a directory path destination relative to the **Source/Destination**.
5. Click either **Monitor Requests** or **Close**.

If you choose **Monitor Requests**, you will be taken to the Requests View where you can monitor the DIVAnet Restore requests for each object selected.

Modifying the Configuration File

The standard DIVAnetUI startup script is configured to load a properties file upon startup. You can find this configuration file within the DIVAnet home directory (`Program/conf/divanet/divanetui.properties`).

Note: DIVAnetUI needs write permissions on the properties file, as it records each change to the URL directly in the properties file.

You can configure a default URL in this file. This URL should refer to the ClientAdapter's network address and `WebServicePort` (configured in `ClientAdapter.xml`). Local web servers and routers may change the address and (or) port needed in the URL.

In this file, you can configure the following parameters:

- **Logging Parameters** — DIVAnetUI writes a log file, and you can change the default logging level by changing the *logLevel* parameter.
- **Default Url** — This represents the url that connects to DIVAnet (the ClientAdapter service). Every time the url is changed in the application, it is updated here in the configuration file.
- **Max Objects to Query** — The maximum number of objects to retrieve in the Assets View.
- **Max Requests to Query** — The maximum number of requests to retrieve in the Requests View.

- **Username** — The user name (for example, *diva*) to pass to DIVAnet, which can be used in Access Rules.

DIVAnet Administration

This chapter discusses administration of DIVAnet.

Using the DIVAnetAdmin Utility

DIVAnetAdmin is a simple command-line tool that has options for administering DIVAnet. This utility can be run only on the server where the ClientAdapter and DbSync services are running.

To start the utility, open a command window, and then navigate from the DIVAnet home directory to the `Program/divanet/bin` subdirectory. Type one of the following commands to start the utility:

Windows:

```
DIVAnetAdmin.bat {ClientAdapterWebUrl} {DbSyncWebUrl}
```

Linux:

```
DIVAnetAdmin.sh {ClientAdapterWebUrl} {DbSyncWebUrl}
```

Note: If you do not supply the URL arguments, the utility will assume that the ClientAdapter and DbSync services are running locally, and will prompt for the ClientAdapter and DbSync service ports.

Table 6–1 shows the DIVAnetAdmin options, selectable through a menu system.

Table 6–1 *DIVAnetAdmin Options*

Category	Option	Description
Asset Database Synchronization	List Synced Sites	Generates the list of the sites in which asset information is being synchronized. It will also display the synchronization status of each site. If a site is in an initial sync state, the percentage complete will also be displayed.
Asset Database Synchronization	Resync a Site	Resynchronizes all object records from a particular site to the DIVAnet database. Caution: This option will delete all asset records in the DIVAnet database (for the selected site) before resyncing them.
Asset Database Synchronization	Resync an Object	Updates object records for a specific object at a specific site. Does not synchronize deletes.

Table 6–1 (Cont.) DIVAnetAdmin Options

Category	Option	Description
API Connections	List API Connection	Lists all Client DIVAnet API connections, including those in Direct Mode.
API connections	End API Connection	Provides an option to end a particular API connection, identified by ID.
Reload Workflow Profiles	Reload Workflow Profiles	This option will cause all Workflow Profiles to be reloaded. It will also reload any Access Rules that are defined. If Access Rules are defined, this option requires access to the ChangeConfig operation.
Site Parameters	Show Site Parameters	This option displays DIVA site parameters, including the number of queued requests to the site, number of requests (sent by DIVAnet) running on each site, date of last communication with the site, and the throttling parameters.
Site Parameters	Set Site Parameters	This option allows the user to set the DIVA throttling parameters: <ul style="list-style-type: none"> ■ The <i>SubmittedThrottleThreshold</i> controls how many requests DIVA should send to a site before waiting (for one of them to complete). ■ The <i>TotalThrottleThreshold</i> is similar, but is based on how many total requests are running on the DIVA site. DIVAnet will wait until the total DIVA requests drops below this number (before sending the next request). If Access Rules are defined, this option requires access to the ChangeConfig operation.
Runtime Stats	Show Runtime Stats: ClientAdapter	This option displays useful run time statistics for the ClientAdapter service.
Runtime Stats	Show Runtime Stats: DbSync	This option displays useful run time statistics for the DbSync service.
Log Level	Set Log Level: ClientAdapter	This option lists the log level of the ClientAdapter services, and provides the option to dynamically change the log level (valid values are ERROR, WARN, INFO, DEBUG, TRACE).
Log Level	Set Log Level: DbSync	This option lists the log level of the DbSync services, and provides the option to dynamically change the log level (valid values are ERROR, WARN, INFO, DEBUG, TRACE).

DIVAnet creates the following types of log files:

- **Trace Logs** — Located within the DIVAnet home directory in the `Program/log/divanet` subfolder. Each of the DIVAnet services (ManagerAdapter, DbSync, and ClientAdapter) create their own subfolder underneath the `Program/log/divanet` folder and generate their log files in that subdirectory.
- **Wrapper Service Logs** — These logs output messages related to startup and shutdown of the windows service. This is the place to look if the service is not starting correctly. These logs are within the DIVAnet home directory in the `Program/log/divanet` subfolder. They will be named similar to the `Wrapper.conf` file, but with a `.log` file name extension (for example, `ManagerAdapterWrapper.log`).

- **GUI Logs** — Errors that occur in DIVAnetUI will appear on the client computer, not on the server side. The logs appear in the `Program/log/divanet/gui` subfolder.

DIVAnet will package logs older than one hour into zip files, and retain two days worth of logs for review.

Request Processing and Queuing

The following sections discuss queuing and processing of DIVAnet requests.

DIVAnet Request Queuing

DIVAnet can accept more requests than downstream DIVArchive systems can process with request queues. DIVAnet enforces a maximum number of requests that can be queued and (or) running internally (the default is 5000), using the `GlobalDivanetRequestLimit` parameter. DIVAnet will start rejecting requests if this limit is reached. The DIVAnet ClientAdapter service maintains a queue for each site. You can configure DIVAnet to feed requests to sites slowly, or to send requests to sites all at once (where they will likely be queued by DIVArchive).

If a request is queued to a DIVArchive system, and the DIVA system is restarted, all running requests will be terminated. DIVAnet behaves differently - DIVAnet stores each request in a database. If the ClientAdapter service is restarted, DIVAnet reads the pending and running requests and restarts them. However, not all requests will be restarted this way. For example, delete requests will never be reissued after the service restarts. If recovering requests on startup is not desired, the ClientAdapter `AbortAllOnStartup` parameter can be set to terminate all requests in the queue at startup.

DIVAnet Request Throttling

By default, DIVAnet will send one hundred requests to a site before stopping and waiting for one of the requests to complete. The parameter that controls the limit is `SubmittedThrottleThreshold`, which is configurable in the ClientAdapter on a per site basis. It can also be changed at run-time through the DIVAnetAdmin utility. In addition, DIVAnet can throttle requests based on the total number of requests running on a DIVArchive site (using the `TotalThrottleThreshold` parameter). You can set either parameter to `-1` to indicate that no throttling should be performed. You can also set the value to `0`, indicating that no requests in the queue should be sent to DIVArchive. This can be useful in certain situations.

If a DIVAnet request is queued to a particular site for more than a few seconds, DIVAnetUI will display the position of the request in the queue. If another request is queued with a higher priority than the first, it will likely have a higher position in the queue. If the request fails and is retried, the request will be queued again (although it will get a priority boost from being in the system for a long time).

Delete requests associated with queuing and timeouts are treated differently than other requests. The `DeleteWaitTimeoutMins` parameter controls how long DIVAnet waits for a delete to complete before timing out the operation. The default is two minutes, but can be increased to be more in line with other commands.

DIVAnet Request Retries

Some requests can be retried automatically within DIVAnet. Copy, Restore, Partial File Restore, and Delete requests can be retried. Restore & Partial File Restore requests can

be retried a configurable number of times, where Copy and Delete requests are retried over a configurable time interval. Retries can be specified on a per Workflow Profile basis, and can be modified at run-time without restarting the ClientAdapter (see [Configuring Workflow Profiles](#) for more information).

If a site is down, and requests are queued to that site, DIVAnet will wait for `SiteDownRequeueWaitMins` minutes for the site to become operational. If that configured time period elapses, and retries are configured, DIVAnet will attempt to retry the request, possibly on another site. If retries are *not* configured the request will terminate.

This chapter highlights DIVAnet's DIVArchive API support, and provides information on how to supply DIVAnet specific information using the DIVArchive API. This chapter is intended to be used with either the DIVArchive C++ API, DIVArchive Java API, or the DIVArchive Web Service API documentation.

- [DIVAnet API Version Compatibility](#)
- [Operations on Archived Content](#)
- [Other Supported Commands](#)
- [Unsupported Requests](#)
- [Return Codes](#)

DIVAnet API Version Compatibility

DIVAnet supports a subset of the full DIVArchive API command set. Some DIVArchive API commands (such as `EjectTape`) will be rejected in DIVAnet MultiDiva Mode. DIVAnet 2.2 does support client connections from DIVArchive API clients releases 7.3 and earlier. Any new features added to the API after 7.3 will not supported by DIVAnet 2.2.

Invoking DIVA API calls against a DIVAnet server is largely the same as invoking calls against DIVArchive. However, there are some differences - DIVAnet sometimes accepts the well-known DIVA API parameters in a slightly different way. In addition, the content of fields returned by DIVAnet may be slightly different from DIVArchive, or have a different format. This section highlights those differences.

DIVAnet 2.2 requires that connected DIVArchive sites be installed with DIVArchive 7.3.1 or later. DIVArchive sites may then be upgraded independently without requiring an upgrade to DIVAnet.

Operations on Archived Content

This section highlights requests that involve the transfer (or deletion) of archived content, including `Archive`, `Restore`, `Delete`, and `Copy`. These commands can be invoked through the DIVArchive API. Some can also be invoked from the DIVAnetUI. Refer to the *Oracle DIVArchive C++ API Programmer's Guide* in the *Oracle DIVArchive Additional Features documentation* library for more information on what each command within the DIVArchive API does.

Note: In DIVAnet MultiDiva Mode, DIVAnet requests often require a few pieces of information that requests issued directly to DIVArchive do not.

DIVAnet requests often require additional information because of the supported functionality. For instance, you can use the Copy command in DIVAnet to copy content from one DIVA system to another. DIVAnet needs to know, at a minimum, what the target site is. However, the DIVA API CopyToGroup command does not contain a *target site* parameter. The sections below detail how to specify this additional information. For more information on configuring DIVAnet MultiDiva Mode, see [Configuring Client API Ports](#).

Table 7–1 identifies the DIVArchive API Content Requests that are supported by DIVAnet. Clients that issue these requests are provided a Request ID in return, which can periodically query the status of the request.

Table 7–1 Supported DIVArchive Content Requests

Request	Behavior in DIVAnet
Archive	Enables digital assets from a Source/Destination (such as FTP or CIFS file system) to be transferred to, and stored in, the local DIVArchive site, or optionally, another chosen site. For more information, see Archive Requests .
Restore Restore Instance	Enables a DIVAnet object to be restored (transferred) to a Source/Destination (such as FTP or CIFS). You can specify the site and (or) instance to restore, or let DIVAnet choose the best site. DIVAnet decides whether it should (1) restore an object from the local DIVArchive, (2) directly restore an object using a remote DIVArchive, or (3) pull an object from a remote DIVArchive and then transfer it to the chosen Source/Destination. A Restore Instance allows a specific instance at a specific site to be restored. If a failure occurs, you can configure DIVAnet to retry on other sites. For more information, see Restore Requests .
Partial Restore Partial Restore Instance	Partially restore an object in a manner similar to a full restore. If issuing a partial file restore on a remote system, that DIVA site must be configured for partial restore in a manner similar to the local site. For more information, see Oracle Partial File Restore Requests .
Copy (CopyToGroup)	Copy content from one DIVA site to another (an Intersite Copy), or create another instance of an object on new media at a DIVA site (equivalent to issuing a CopyToGroup on a single DIVA system). A specific instance can be copied to a target site. The CopyToNew message is not supported. You can configure DIVAnet to retry for a given time period. For more information, see Copy Requests .
Delete DeleteInstance	Delete content from all sites, a specific site, or delete a specific instance on a specific site. If objects are locked on sites to be deleted, you can configure DIVAnet to retry for a given time period. For more information, see Delete Requests .

Archive Requests

An Archive request allows the caller to archive content that exists on a particular Source/Destination (configured in DIVArchive). The DIVA API arranges for a transfer to occur from the Source/Destination to DIVArchive. This differs from a web-based cloud API where the content is transferred directly from the requester through HTTPS. By default, DIVAnet archives to the local site.

Archive requests issued to DIVAnet are similar to those directly to DIVArchive, but with a few additions to the *Target Sitename* - the DIVArchive site where you will archive the content. Normally, DIVAnet will archive to the local site. However, you can archive directly to another site in one of two ways:

- By providing the **-site** {sitename} option in the options field. An example would be **-site diva1**.
- By prefixing a destination sitename to the media parameter in the archive request. For example, *sitename1_TapeGroup1* indicates a destination site called *sitename1*, and a media of *TapeGroup1*.

DIVAnet does not support continuous retries for Archive commands, but does support a `BackupArchiveSite` option, which provides an alternate archive site in case the primary site is down.

Restore Requests

A Restore request allows the client to restore content that exists in the archive system. The content arrives at the particular Source/Destination chosen in the request. The DIVA API arranges a transfer from a DIVArchive site directly to a Source/Destination (such as FTP or CIFS disk). This differs from a web-based cloud API where the content is transferred directly to the requester through HTTPS.

When restoring content using DIVAnet, the caller need not know which particular DIVA system has the content. And if failures occur retrieving content from one DIVA site, another DIVA site can automatically be consulted to retrieve the content.

DIVAnet supports restoring to any Source/Destination on any DIVAnet site. DIVAnet will retrieve content from other sites as needed to satisfy the request, ultimately transferring the content to the target Source/Destination.

Restore requests issued to DIVAnet are similar to those issued directly to DIVArchive, but with a few additions. Normally, DIVAnet chooses the best site to restore from. However, DIVAnet enables a client to specify the specific site to use for the restore. There are several ways this can be done:

- **-site**: Normally, DIVAnet will choose the site from which to restore. However, you can attempt a restore from a particular site by providing the **-site** [*sitename*] option in the options field of the request. If in fact the content is not located at the chosen site, the operation will fail.
- **Instance Id**: If you need complete control over the source, you can provide an instance number in the restore request. This will allow you to choose the source site, and the DIVA instance to restore from (see the following section). This instance id can be obtained by performing a `getObjectInfo()` API call, or by viewing the object in DIVAnet UI.

In both of these cases, retries are disabled.

Restore Workflow

To satisfy a restore request, DIVAnet uses the restore methods in [Table 7-2](#). DIVAnet will dynamically select which restore workflow to use based on parameters such as the target Source/Destination and the source object. To decide which site to use for restores, DIVAnet asks a series of questions, including:

- Is the object available on the local DIVArchive System?
- Does the object have a disk instance?
- Is the Source/Destination accessible from the remote DIVArchive System?
- Is the Source/Destination accessible from the local site?
- Is DIVArchive running on the source or target sites?
- Is one site preferred over another in the configuration file?

Table 7-2 *DIVAnet Restore Methods*

Method	Description
Local	Used when an object exists on the local site. The local site is the DIVArchive System sitename to which you are sending messages. A local DIVArchive System is also considered part of the local site.
Direct Remote	DIVAnet can arrange to have a remote DIVA system perform a restore operation. It will do this only if the target Source/Destination is also configured in the remote DIVA system. The Source/Destination names must match, and they must both refer to the same server or disk (and path on that disk, if applicable). When available, DIVAnet prefers this method over performing a Restore using Intersite Copy.
Using Intersite Copy	If content is not local, and if a remote DIVA system cannot directly restore to the target Source/Destination, DIVAnet can arrange for the content to be delivered in two hops. First, the remote DIVA restores to a Source/Destination shared between the source and target sites. Then, the local DIVA will archive the object, and finally, restore to the target Source/Destination. This way future requests for the content will be retrieved much quicker. If it is desired to always perform remote restores by creating a nearline copy, set <code>AllowDirectRemoteRestores</code> to false in the Workflow Profile.
Using Intersite Transfer	In certain cases, when DIVAnet is not able to perform a direct remote restore, DIVAnet will deliver the content in two hops (as Restore using Intersite Copy) but will not actually archive the content locally. One such case involves Oracle Partial File Restore. First, DIVAnet will instruct the source DIVA to transfer the content to the Source/Destination accessible by both the source and target DIVA sites. Then, the local DIVA site will transfer the content to the target Source/Destination without archiving it.

Sources and Destinations

DIVAnet will allow you to restore to a Source/Destination that is available on any site. DIVAnet assumes that if a Source/Destination exists with the same name on two sites, both configurations point to the same physical `Server\Device\Path`. Users must be careful with the names that are assigned to Source/Destinations across the sites.

If a remote DIVA system cannot directly restore to the target Source/Destination, DIVAnet can arrange for the content to be delivered in two hops (see [Restore Using Intersite Transfer](#)). DIVAnet will arrange this only if the Source/Destination is the preferred site for the particular Source/Destination (see [Preferred Source/Destination Mappings](#)).

Retries

You can configure the Restore command to retry several times if the first restore fails. If the content to be restored exists on multiple sites, DIVAnet will automatically retry the restore with those sites. You can configure the maximum number of retries. In some cases, DIVAnet will decide to retry with the same site before moving on to other sites. In this case, DIVAnet will examine the `RestoreRetryIntervalMins` value to determine how long to wait before retrying with the same site.

Limitations

- DIVAnet 2.2 supports a subset of the DIVArchive API. Refer to this chapter for a list of the supported messages.
- DIVAnet 2.2 has limited support for Multi Restore. DIVAnet does not allow Multi Restores for remote Source/Destinations, and does not allow viewing or monitoring of multiple Source/Destinations through the UI. Multi Restore can be used if the object exists locally, but cannot be accessed (for example, it has been externalized locally).
- DIVAnet 2.2 does not support peer-to-peer site relationships, including load balancing.

Oracle Partial File Restore Requests

In addition to full restores of content, DIVAnet also supports Partial File Restores. DIVAnet determines the site where the content is located, and arranges to have the portion restored.

If an object exists on a remote DIVA system, and the target Source/Destination for the restore is not accessible by a remote DIVA system, DIVAnet transfers the content in two steps - first using the remote DIVA to get the content to the local DIVA (without transferring the entire object), and then using the local DIVA to restore the content to the target Source/Destination.

As with restore, you can specify the instance number or `-site` parameter to perform a restore from a specific site, or specify retries when the initial site fails. And also as with restore, restoring to multiple destinations within the same restore request is not supported.

Copy Requests

A Copy request creates a new instance of archived content from an existing instance. DIVAnet allows content to be copied from one DIVA site to another. The DIVArchive API `CopyToGroup` command (1) copies an object from one DIVA site to another, or (2) simply creates a new instance on a single site. For copies, DIVAnet needs to derive some parameters that are not available in the DIVA API. [Table 7-3](#) describes these parameters.

Table 7–3 DIVAnet Copy Derived Parameters

Derived Attribute	Description
Target Sitename	<p>Indicates which site the object should be copied to.</p> <p>Target sitename does not exist as an architected field in the DIVA API. You can convey this to DIVAnet in one of two ways:</p> <ul style="list-style-type: none"> ▪ -site Option: By providing the -site [sitename] option in the options field. An example would be -site sitename1. ▪ Sitename Media Prefix: By prefixing a destination sitename to the media parameter in a copy request (with an underscore (_) separating the two fields). For example, <i>sitename1_TapeGroup1</i> indicates a copy destination site called <i>sitename1</i>, and a media of <i>TapeGroup1</i>. <p>If you specify no sitename, the local site will be assumed. The -site keyword will work only with an API release 7.3 or greater.</p>
Media	<p>Media indicates the type of media to use to store the copied object. DIVAnet also enables providing a DIVA Storage Plan as a media name. Storage Plan will work only if the copy is an intersite copy.</p> <p>You can prefix the target sitename to the media to indicate the target sitename as well.</p> <p>If you are not sure which media to provide, you can provide a media of any to let the system choose which media to store to on the target site. For example, a media of <i>diva1_any</i> copies to site <i>diva1</i>, but DIVAnet chooses the media. The default that DIVAnet provides may not be appropriate for some use cases.</p> <p>If the object is already on the target site, and any is specified, the system will simply return success.</p>
Source Sitename (Instance Id)	<p>Normally, DIVAnet will choose the site to copy from. However, if you need complete control over the source, you can provide an instance number in the copy request. This implicitly enables choosing the source site, and which DIVA instance to copy from (see the following section). This id can be obtained by performing a <code>getObjectInfo()</code> API call on the object to be copied.</p>

Site-to-Site Transfer Methods

In a DIVAnet Copy request, if the source site is the same as the target site, DIVAnet can simply issue a `CopyToGroup` to the target DIVA site. For intersite copies, DIVAnet enables configuring the method used to perform these copies. For each source and target sitename pair (for example, site1 to site2), [Table 7–4](#) lists the available transfer methods.

In addition to the copy method, each source and target sitename pairing contains the actual Source/Destinations used for the common storage area. The default destination media (for **RestoreAndArchive**), options parameters (passed to `DIVArchive`), and other parameters are also configurable.

Table 7–4 Site-to-Site Copy Methods

Type	Description
RestoreAndArchive	<p>With this option, DIVAnet restores content from the source site, to a Source/Destination that is common to the source and target sites. Then, DIVAnet instructs the target <code>DIVArchive</code> System to archive the content now in the common storage area. This is an alternative to using Drop Folders.</p>

Table 7-4 (Cont.) Site-to-Site Copy Methods

Type	Description
RestoreAndMonitor	<p>Using this method, DIVAnet will perform the copy by first restoring the content to a specific destination. DIVAnet will then turn to monitoring the target DIVArchive system to determine when the content is successfully archived on the target site. The request will successfully complete only when the content is successfully archived on the target site.</p> <p>This of course relies on another process or program that will take the content and archive it into the target DIVA system. This option is useful in combination with DIVArchive Drop Folder Monitor software (DFM). Each DFM folder is configured to archive using a preselected media - meaning that when DFM is employed for copies, the media parameter is effectively ignored.</p>
Restore	<p>Using this method, DIVAnet will perform the copy by restoring to a specific destination, and then returning success. This method does not Confirm that the content has been successfully archived into the target site, and will likely cause failures if Restore Using Intersite Copy workflows are attempted.</p>

The DIVAnet Copy command will return success if an instance of the object already exists in the target site on the requested media. DIVArchive will terminate the request in this case.

Local Copies

If an object is requested to be copied to a site where the object already exists, but does not have the requested media, DIVAnet will create another instance of the object on that site using the media specified in the request. The exception to this is if **any** is supplied as the media. In this case, DIVAnet will not create another instance.

Selected By DIVAnet (media of any)

In DIVAnetUI, there is an option to have DIVAnet assign the target media on a Copy operation (use a media of `Selected By DIVAnet`). You can accomplish the same thing in a DIVA API request by specifying a media of **any** in the Copy request. DIVAnet will use its configuration to determine which media to use for the copy (see [Site-to-Site Mappings](#) for more information).

When **any** is passed, and the object already exists on the target site, DIVAnet will not create another instance of the object. No further action will be taken, and the request will succeed.

Retries

DIVAnet also supports periodic retries on copies. If enabled in the Workflow Profile, DIVAnet will retry copy operations that failed. In the Workflow Profile, you can configure how long DIVAnet will retry the request, and the wait period between retries. See [Configuring Workflow Profiles](#) for more information.

Delete Requests

A DIVArchive Delete request allows the caller to remove an archived object. A DIVAnet Delete request, by default, will delete the object from all DIVArchive sites. A DIVAnet DeleteInstance request can delete one or all instances from a single DIVA site. So in reality, DIVAnet can perform three kinds of deletes. [Table 7-5](#) describes the delete types and parameters they require.

Table 7-5 Delete Types

Type	Derived Attributes	Description
Global Delete	NA	Deletes all object instances on all sites. In the API, not specifying a media or instance id in the request will cause the specified object to be deleted from all sites.
Instance Delete	Target Sitename	Deletes a single object instance on a particular site. In the API, specify an instance id or a media to delete a specific object instance from a specific site. If you specify an instance id, you are targeting a specific instance on a specific site. Passing media or options parameters are not necessary. If you specify a media, DIVAnet needs to know the site you would like to delete from. You can specify the site in one of three ways: <ul style="list-style-type: none"> ▪ Media Sitename Prefix: You can prefix the sitename to the media. For example, <i>siteA_tapeGroup1</i> tells DIVAnet to delete the instance on <i>siteA</i> having a media of <i>tapeGroup1</i>. If you do not provide a sitename, the local site will be assumed. ▪ -site Option: Provide -site [sitename] in the options field, and supply a media. This is supported with DIVArchive API 7.3 or later. ▪ Media (with no site): You pass the media with no site. In this case, the local site will be assumed.
Site Delete	Target Sitename	Deletes all instances of an object residing on a particular site. This can be accomplished in one of three ways: <ul style="list-style-type: none"> ▪ Use DIVAnetUI: You can delete all instances on a particular site by issuing a Delete command from the DIVAnetUI. ▪ Sitename/Media of any: Issue DeleteInstance through the API by providing the sitename and a media of any in the media field (for example, <i>siteA_any</i>). ▪ -site Option: Specify a sitename by providing -site [sitename] in the options field. This is supported with DIVArchive API 7.3 or later.

Site Delete (media of any)

As mentioned in the previous table, if a sitename is supplied and a media of **any**, this will cause all instances of the object to be deleted from the selected site. This can also be accomplished by passing **-site [sitename]** in the options field.

Last Instance

DIVAnet will not allow an Instance Delete to succeed when attempting to delete the last instance of an archived object (the last instance that exists in the DIVAnet database, that is). In this case a Global Delete or Site Delete request will need to be issued. Note however that a Site Delete will allow the last instance or instances to be deleted! You can use Access Rules to protect against Instance Deletes or Site Deletes that are effectively Global Deletes (see [Configuring Access Rules](#) for more information).

Also, if a user issues a delete directly to DIVArchive, there may be a time window where DIVAnet cannot ensure that the last instance is preserved.

Retries and Cancels

If a Delete is received by DIVAnet, and DIVAnet is currently performing an Intersite Copy of the object, the DIVAnet request that spawned the copy will be canceled. The copy could be the result of a Copy command, or, it could be the result of a Restore command that performs a copy to satisfy the restore. Other types of DIVAnet requests will not be canceled.

If a DIVArchive request is being executed on behalf of a DIVAnet request, DIVArchive locks the object and prevents it from being deleted. So, if an object is locked before DIVAnet can send a delete message to DIVArchive, the delete request will fail.

DIVAnet supports periodic retries on deletes when deletes fail. If enabled in the Workflow Profile, DIVAnet will keep trying to delete on sites where (for example) instances/objects to be deleted are locked. You can configure within DIVAnet how long to retry.

Other Supported Commands

Table 7–6 identifies DIVArchive API commands that do not specifically process or transfer archived content. These commands obtain information about objects or requests and are not assigned a request ID.

Table 7–6 Other Supported DIVArchive Commands Not Related to Content

Request	Description	Behavior in DIVAnet
Cancel	Cancel a DIVAnet request.	The -site option does not apply for this command.
GetObjectInfo	Use the DIVAnet Database to get information about an archived object. DIVAnet returns all instances of the object on all DIVAnet sites. You provide the Object Name and Object Category (you can leave the Category blank, but if multiple objects have the same Object Name, the call will fail). DIVAnet uses the DIVAnet database to return information about the archived object. In the <code>GetObjectInfo()</code> response, DIVAnet prepends the sitename to the media parameter of each instance of an object.	The -site option is not supported for this command.
GetRequestInfo	Retrieve information regarding a DIVAnet request from the DIVAnet database. Note: The Additional Information parameter has limitations. The additional information is provided by DIVArchive and represents information from the last DIVA request processed. The information does not reflect the other sites in the DIVAnet network.	When DIVAnet receives a <code>GetRequestInfo()</code> call, it provides information about DIVAnet requests, not DIVArchive requests (although the DIVAnet requests often involve invoking one or more DIVArchive requests). The -site option does not apply for this command.
GetFilesAndFolders	Retrieve information directly from DIVArchive about the files and folders within a particular archived object.	Accepts the -site option to query a specific site, or no site to let DIVAnet choose (recommended).

Table 7–6 (Cont.) Other Supported DIVArchive Commands Not Related to Content

Request	Description	Behavior in DIVAnet
GetObjectDetailsList	<p>Retrieve information from DIVArchive regarding objects and object events. DIVAnet retrieves the object information directly from each DIVArchive system one site at a time, in a round-robin fashion, one batch per site.</p> <p>Each batch contains information from one DIVA site. If the same object exists on two sites, you will receive the object twice (once for each site), once in each batch.</p> <p>Note: The order of the returned entries is not guaranteed. If one DIVA site is down, GetObjectDetailsList() will return an error and processing of the command will stop.</p>	<p>You can retrieve information from one site by prefixing the sitename to the media field separated with an underscore (_). If you do not want to query on media, but still want to retrieve information from one site, you can supply just the sitename in the media field. This command does not accept the -site option.</p>
GetObjectsList	<p>Queries the DIVAnet Database to get a list of Object Name and Categories (InitObjectList, CloseObjectList).</p>	<p>There is limited support for this command within DIVAnet. Tape information queries are not supported; and there are limits on number of simultaneous queries. The -site option is not supported.</p>
GetArrayList	<p>Returns a List of Array Names from all sites including the disks that form each array, and returning the current disk capacities.</p>	<p>The -site parameter, passed in the options field, can return information for a specific site.</p>
GetGroupsList	<p>Returns a list of Tape Group Names from all sites. The sitename is prefixed to the Group Name.</p>	<p>Note: This command does not have an options field. As a result, the -site options is not supported.</p>
GetSourceDestinationList	<p>Returns a list of Source and Destination information from all sites.</p>	<p>In the returned list, the sitename of the Source or Destination is prefixed to the Source or Destination name separated by an underscore (_).</p> <p>The -site parameter, passed in the options field, can return information for a specific site.</p>

Table 7–6 (Cont.) Other Supported DIVArchive Commands Not Related to Content

Request	Description	Behavior in DIVAnet
GetArchiveSystemInfo	Retrieves the status of a single DIVArchive site (by default, the local site is returned). It does not return a global view of all sites.	The -site parameter, passed in the options field, selects which site to gather information from. For example, -site diva1 will route the <code>GetArchiveSystemInfo</code> request to the Manager designated with the sitename <i>diva1</i> and return Archive System Information for <i>diva1</i> only.
GetStoragePlanList	Retrieves the list of storage plans that are defined in all configured DIVArchive sites.	In the returned list, the sitename of the Storage Plan is prefixed to the Storage Plan name, separated by an underscore (_). The -site parameter, <code>passwd</code> in the options field, returns information for a specific site.
GetObjectsByFilename DeleteFile	The DIVArchive AMC product uses calls (for internal use only). The workflows supported in this release are limited to those where all desired DIVArchive AMC objects exist on the local site. Replicas of the content can exist and be restored from other sites.	DIVAnet invokes these commands on the local site. If the local site is down, <code>GetObjectsByFilename</code> is invoked on another site. Limited sequence support.

Unsupported Requests

For application compatibility reasons, these commands will always return success, even though DIVAnet will take no action to satisfy them.

- Change Priority
- Lock Object
- Unlock Object
- Link Objects
- Require Instance
- Release Instance

Return Codes

DIVAnet returns status codes that are similar to what DIVArchive returns. However, DIVAnet will sometimes accept requests that DIVArchive would fail immediately, because DIVAnet often does not have the information needed to perform checking until later during request processing.

In addition, DIVAnet will return the `ACCESS_DENIED` status for many commands. This status is not returned by DIVArchive. DIVAnet will reject requests that do not pass Access Rules checks, and rejects messages not configured in the `WorkflowProfile`. For compatibility purposes, API release 5.8 and earlier return `INVALID_PARAMETER` status instead of `ACCESS_DENIED`.

Troubleshooting

The following are common errors that can occur in a DIVAnet installation. For instance, although tag names in configuration files are not case sensitive, the values typically are (for example, sitenames), and can cause errors.

- [DIVAnet Common Errors](#)
- [DIVAnetUI Common Errors](#)

Table 8–1 *DIVAnet Common Errors*

Problem	Possible Resolution
A DIVAnet service does not start	<ul style="list-style-type: none"> ■ When installing a service, make sure to pass the name of the adapter's wrapper .conf file and not the name of the adapter's configuration file. ■ Make sure to place the configuration file in the parent directory of the templates folder, and remove the .ini extension. ■ If there are errors reading the configuration file, they will appear in the wrapper logs. These logs are placed in the DIVAnet home directory, in the Program/log/divanet folder, and will be named similarly to the Wrapper.conf file, but with a .log file name extension. ■ You must run the addSites script before attempting to start the ClientAdapter or DbSync. ■ Are the sitenames consistent between the database and the configuration files? If not, services may not start and (or) not function properly.
DIVAnet request is returning access denied	<ul style="list-style-type: none"> ■ Ensure that the correct Workflow Profile is configured for the port you are connecting on. Ensure that the request is permitted in the list of valid Messages section of the Workflow Profile. ■ If you are using Access Rules, ensure that ApiConnect appears in an Include rule. If you are using DIVAnetUI, ensure that WebConnect appears.

Table 8–1 (Cont.) DIVAnet Common Errors

Problem	Possible Resolution
Intersite copy operations fail with errors referring to Source and Destination names containing <i>MISSING_MAPPING_TO</i>	<ul style="list-style-type: none"> Ensure that the request is being assigned the Workflow Profile that you expect by checking the configuration. The ClientAdapter Site-To-Site Transfer configuration may be missing the from Source or to Destination.
No requests are being sent successfully to a DIVA site, even though they can be viewed in DIVAnet	<ul style="list-style-type: none"> Ensure that the <i>localSitename</i> in the ManagerAdapter configuration files correctly reflect the site that they are serving (and are not cut and pasted from somewhere else). Check that the <i>localSitename</i> is correct in the ClientAdapter and DbSync configurations. Check that the sitenames used are consistent with those defined in the DIVAnet database. Check that the network address and port of the DIVA Manager you want to send to is correct in the ClientAdapter configuration.
In Access Rules, after defining an Exclude rule, the operation is being rejected in all cases.	<ul style="list-style-type: none"> Remember that for any operation to succeed, there must exist one Include rule for the operation in question. Pay attention to any defaulted parameters in the Ruleset, such as <i>WorkflowProfile</i>, and change them accordingly.

Table 8–2 DIVAnetUI Common Errors

Problem	Possible Resolution
Cannot connect to DIVAnet	<ul style="list-style-type: none"> Ensure that you have specified the correct URL of DIVAnet. By default, this is the address of the computer that the ClientAdapter is running on, at the port defined in the <i>WebServicePort</i> parameter of the ClientAdapter configuration. Make sure to specify the port in the url (if applicable), and specify <i>https</i> if secure connections are enabled.
UI is no longer updating progress of existing requests, or not showing new requests.	<ul style="list-style-type: none"> Check the lower-right corner of the DIVAnet GUI to determine if a connection error has occurred. If this is the case, click the Requests or Assets buttons to re-spawn the connect dialog box. Perhaps you have provided an end time in your request query. This indicates to DIVAnetUI that live updates should not be shown, and no progress should be updated for current requests. To enable live updates, just delete the end time filter. Finally, check the Request Filters pane to ensure that there are not any Request Type or Request Status filters enabled (likewise for Start and End time).
Newly archived objects are not showing up in the Assets View	<ul style="list-style-type: none"> Check that the DbSync service is running. Check the configuration, ensuring that the network addresses are correct. You may need to ensure that the ManagerAdapter on the site that you are syncing is configured and running. If the configuration is correct and problems persist, it may be advisable to initiate a resync of that site through DIVAnet's Admin utility.
A Copy, Delete, Cancel operation in the UI is returning access denied.	<ul style="list-style-type: none"> Ensure that the correct Workflow Profile is configured in the <i>WebDefaultWorkflowProfile</i> in the ClientAdapter. Ensure that the request is permitted in the list of valid Messages section of the Workflow Profile. If you are using Access Rules, ensure that WebConnect appears in an Include rule.

Sample Configuration Files

- [ClientAdapter Sample Configuration](#)
- [ManagerAdapter Sample Configuration](#)
- [DbSync Sample Configuration](#)

ClientAdapter Sample Configuration

```
<config>
  <!-- TOP-LEVEL -->
  <LocalSiteName>diva1</LocalSiteName>
  <LogLevel>INFO</LogLevel>
  <MaxClientConnections>50</MaxClientConnections>
  <WebServicePort>9801</WebManagementPort>

  <!-- ADAPTER PORTS -->
  <AdapterPorts>
    <AdapterPort>
      <ListenPort>7100</ListenPort>
      <RoutingType>Direct</RoutingType>
      <Sitename>diva1</Sitename>
    </AdapterPort>
    <AdapterPort>
      <ListenPort>7101</ListenPort>
      <RoutingType>MultiDiva</RoutingType>
      <WorkflowProfile>default</WorkflowProfile>
    </AdapterPort>
  </AdapterPorts>

  <!-- DIVA MANAGERS -->
  <DivaManagers>
    <DivaManager>
      <Sitename>diva1</Sitename>
      <ConnectionType>Socket</ConnectionType>
      <Address>127.0.0.1</Address>
      <Port>9000</Port>
    </DivaManager>
    <DivaManager>
      <Sitename>diva2</Sitename>
      <ConnectionType>WebService</ConnectionType>
      <BaseUrl>https://172.16.3.59:8101</BaseUrl>
    </DivaManager>
  </DivaManagers>

  <!-- DIVAnet DATABASE -->
```

```

<DivanetDatabase>
  <Address>127.0.0.1</Address>
  <Port>1521</Port>
  <User></User>
  <Password></Password>
</DivanetDatabase>

<!-- WORKFLOW PROFILES -->
<WorkflowProfile>
  <Name>default</Name>
  <RestoreRetryAttempts>3</RestoreRetryAttempts>
  <Messages>
    <Message>Copy</Message>
    <Message>Archive</Message>
    <Message>Restore</Message>
    <Message>Delete</Message>
    <Message>Cancel</Message>
    <Message>AllInfo</Message>
    <Message>PartialRestore</Message>
  </Messages>

  <!-- SITE TO SITE TRANSFER CONFIG -->
  <Mappings>
    <SiteToSiteTransfer>
      <!--IF transferring...-->
        <FromSitename>diva1</FromSitename>
        <ToSitename>diva2</ToSitename>
      <!--THEN use these...-->
        <FromSrcDest>viva</FromSrcDest>
        <Type>RestoreAndArchive</Type>
        <ToDefaultMedia>disk_001</ToDefaultMedia>
        <ToSrcDest>arch</ToSrcDest>
    </SiteToSiteTransfer>
    <SiteToSiteTransfer>
      <!--IF transferring...-->
        <FromSitename>diva2</FromSitename>
        <ToSitename>diva1</ToSitename>
      <!--THEN use these...-->
        <FromSrcDest>viva</FromSrcDest>
        <Type>RestoreAndArchive</Type>
        <ToDefaultMedia>disk_002</ToDefaultMedia>
        <ToSrcDest>arch</ToSrcDest>
    </SiteToSiteTransfer>
  </Mappings>
</WorkflowProfile>
</config>

```

ManagerAdapter Sample Configuration

```

<config>
  <!-- TOP-LEVEL -->
  <LocalSiteName>diva1</LocalSiteName>
  <ManagerAddress>127.0.0.1</ManagerAddress>
  <ManagerPort>9000</ManagerPort>
  <LogLevel>INFO</LogLevel>
  <WebServicePort>9800</WebServicePort>

  <!-- Manager DATABASE -->
  <ManagerDatabase>
    <Address>127.0.0.1</Address>

```

```
        <Port>1521</Port>
        <User></User>
        <Password></Password>
    </ManagerDatabase>
</config>
```

DbSync Sample Configuration

```
<config>
  <!-- TOP-LEVEL -->
  <LocalSiteName>diva1</LocalSiteName>
  <LogLevel>INFO</LogLevel>
  <WebServicePort>9802</WebServicePort>

  <!--Sites to SYNC -->
  <DivaManagers>
    <DivaManager>
      <Sitename>diva1</Sitename>
      <BaseUrl>https://127.0.0.1:9800</BaseUrl>
    </DivaManager>
    <DivaManager>
      <Sitename>diva2</Sitename>
      <BaseUrl>https://172.16.3.59:9800</BaseUrl>
    </DivaManager>
  </DivaManagers>

  <!-- DIVAnet DATABASE -->
  <DivanetDatabase>
    <Address>127.0.0.1</Address>
    <Port>1521</Port>
    <User></User>
    <Password></Password>
  </DivanetDatabase>
</config>
```

Glossary

Archive Exchange Format (AXF)

A vendor-neutral format, supported by DIVArchive, for storing content on various mediums (including tape and disk). It can store many files and directories in a single file (container) much like a ZIP file.

Content Request

System operations that handle archived content — Archive, Copy, Delete, Restore.

DIVAnet Home Directory

The file system directory where DIVAnet is installed. DIVAnet can be installed in its own directory (recommended), or within the DIVArchive home directory.

DIVAnet Object (Asset)

An archived object describes digital content in the archive system. A DIVAnet object is a set of archive objects, archived at one or more DIVArchive sites, sharing the same name (object name and category pair). Each object on a particular site can have one or more copies (object instances).

DIVAnet Site

A sitename uniquely identifies a DIVAnet site, which is defined as exactly one DIVArchive installation and one or more DIVAnet services.

Drop Folder Monitor (DFM)

The DIVArchive DFM service monitors content that is being written to disk in a folder. It then performs the operation assigned to that folder (Archive, Restore, Copy, and so on) to the content (placed in the folder) into a DIVArchive system after the content has been completely written to the folder location.

Intersite Copy

An operation that copies content from one DIVA site to another, archiving the content to the target site.

Intersite Transfer

An operation that moves content from one site to another, but does not archive the content to the target site.

Media

A name that represents a set of archive media (disks or tapes) in a DIVA system. There exists one media per copy of an asset.

MultiDiva Mode

When an API client is connected to DIVAnet in MultiDiva mode, multiple DIVArchive sites appear to the client as one large archive system.

Site-to-Site Mapping

Describes how DIVAnet will copy objects from one site to another.

Source/Destination

Contains information necessary to communicate with a connected server or disk. You transfer content to and from DIVArchive through these servers and disks. Examples are Video servers, FTP servers, and disk arrays.

Storage Plan Manager (SPM)

You can configure the Oracle DIVArchive Storage Plan Manager (SPM) service to automatically perform an operation when content arrives at a DIVArchive site (such as restore or copy).

Workflow Profile

A set of parameters that defines how incoming requests will be processed by DIVAnet. Multiple profiles can be created and mapped to input ports.