HP OpenView Distributed Configuration Server Using Radia

Radia Distributed Configuration Server Guide

Software Version: 4.7

for the HP-UX, Solaris, and Windows operating systems

Technical Support

Please select Support & Services from the following web site:

<http://www.hp.com/managementsoftware/services>

There you will find contact information and details about the products, services, and support that HP OpenView offers.

The support site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

About this Guide

Who this Guide is for

This guide is for Radia administrators who want to simplify the management of multiple Radia Configuration Servers across a networked Windows environment, by sharing information about policies and managed applications.

What this Guide is about

This manual is a comprehensive planning guide that provides instructions on the installation, configuration, and use of the Radia DCS, a powerful Radia agent that improves the flexibility and extensibility of the Radia infrastructure in a Windows environment.

Summary of Changes

Version Differences

Note

This section has been included for the benefit of existing Radia DCS customers.

This version of the *Radia© Distributed Configuration Server* (Radia DCS) contains several *performance*, *operational*, and *user-interface* differences from previous versions. These differences include the addition of some features and options, and the removal of others. These changes are detailed in Table P.1.

Table P.1 ~ Differences in Version 4.7	
Change	**Description**
User Interface	In 4.7 there is a scaled-back user interface, with which a Radia administrator can only configure Radia DCS synchronizations. This interface is the file, **dcs.exe**, and it must be installed on any Radia Configuration Server that will act as Destination.
	Once the Destination component is installed, an invocation option (called **Configuration**) can be accessed by navigating through the **Start** menu (and optionally from a desktop icon called **DCS Configuration**). See *Installing the Radia DCS* starting on page 31.
	Note: This configuration-only interface cannot be invoked from a command line.
Executables	In 4.7 the Radia DCS Batch executable is **dmabatch.exe**, aligning it with version 4.5. See *Chapter 6:Radia DMABATCH* starting on page 85.
	Note: Customers that have been scripting Radia DCS with version 4.6 will need to make this adjustment from **dcsbatch.exe**.
Differencing	The tree-differencing feature that was introduced in 4.6 is not supported in 4.7.
	See *Chapter 5: Configuring Radia DCS Options*, *The Differencing Tab*, starting on page 78.
	The control differencing vs. CRC differencing choice in earlier versions is superseded by a single new differencing method in version 4.7.
	See *Chapter 5: Configuring Radia DCS Options*, *The Differencing Tab*, starting on page 78.
Patching	The byte-level differencing (BLD) *patching* feature that was introduced in version 4.6 (activated with the **Build Resources by Patching** check box on the **Staging** tab) is not supported in version 4.7.
	See *Chapter 5: Configuring Radia DCS Options*, *The Staging Tab*, starting on page 79.
Pre-Staging	The external pre-staging feature, introduced in version 4.6 (the **External Staging Location** field of the **Staging** tab), is not supported in version 4.7.
	See the section, *Radia DCS Processes Defined*, starting on page 25.

Table P.1 ~ Differences in Version 4.7	
Change	**Description**
Database Verification	The pre-synchronization database-verification feature that was optional in version 4.6 (enabled with the **Verify control data (pre-sync)** check box on the **Batch Operation** tab) is not selectable because it is permanently active in version 4.7.
	See the section, *The Batch Operation Tab*, starting on page 80.
Lock Exposure	The **Batch Lock Exposure Action** option which, in version 4.6 on the **Batch Security** tab, could be set to **Continue**, **Reset**, or **Continue if DB unchanged**, is, in version 4.7, hard-coded to **Continue if DB unchanged**.
	See the section, *The Batch Security Tab*, starting on page 82.
Radia Integration Server	In version 4.7, the Source component of Radia DCS (which contains the Radia Integration Server—Radia's HTTP server) must be co-resident with any RCS that will be a *master* in a Radia DCS synchronization.
	See the section, *Source Component*, starting on page 19.
Co-Residency	In version 4.7, the Destination component of Radia DCS (dmabatch.exe) must be co-resident with any RCS that will act as *destination* in a Radia DCS synchronization.
	See the section, *Destination Component*, starting on page 20.
COMMIT=NO	In 4.7, if the COMMIT=NO command-line option is specified, and the synchronization is reset without the commit being done (as specified), resource orphans for the added resources might be left in the Destination database.
	Note: This behavior was not present in 4.6 because COMMIT=NO allowed the synchronization to be done in two phases, at different times of day, but did not allow a partial synchronization.
	See the section, *Deferred Commit*, starting on page 90.
Lost & Found	In version 4.6, the goal was to accurately replicate databases, even if it meant replicating logical database errors. In 4.7, the intent is beyond simple replication—rather, it is the creation of a verifiable Destination database.
	Note: In 4.7, database items that can be identified as being "in-error" or inconsistent, such as resource orphans, will be placed into a new directory, **lost+found**, under the Destination database root, instead of being placed in the Destination database. The result should then pass the database verification.
	See the section, *Radia DCS Processes Defined*, starting on page 25.

Documentation Map

Table P.1 provides an overview of this book, so that specific information about the Radia DCS can be easily located.

Table P.1 ~ Documentation Map	
To learn more about:	**Go to:**
Radia DCS, including: *how Radia DCS works*; *the roles of the two Radia DCS components*; *RCS eligibility*, *domain ownership* and *domain eligibility...*	Chapter 1: *Introduction to Radia DCS*

Table P.1 ~ Documentation Map

To learn more about:	Go to:
Installing the two Radia DCS components, including: *system recommendations* and *desktop shortcut icons*...	Chapter 2: *Installing Radia DCS*
Setting up security for Radia DCS, including: *password protection on the host operating system*...	Chapter 3: *Radia DCS Security*
Defining Configuration Servers to Radia DCS, including: *specifying their properties*; and *choosing a synchronization pair*...	Chapter 4: *Setting Up a Radia DCS Synchronization*
Configuring Radia DCS, including: *a detailed look at the various DCS settings that are available on the DCS Options tabs*...	Chapter 5: *Configuring Radia DCS Options*
Executing a Radia DCS session on a command-line, including: *scripting*, and *automation and command-line options*...	Chapter 6: *Radia DMABATCH*
Troubleshooting Radia DCS, including: *logs*, *tracing*, and *domain eligibility*...	Appendix A: *Troubleshooting Radia DCS*

Conventions

The *style conventions* that are used in this book are listed in Table P.2.

Table P.2 ~ Styles		
Element	**Style**	**Example**
References	*Italic*	See the *Publishing Applications and Content* chapter in this book.
Dialog boxes and windows	**Bold**	The **Radia System Explorer Security Information** dialog box opens.

↵	Arial Unicode MS	When displaying lines of code that extend beyond the defined margins of the manuscript, this symbol indicates that the code continues uninterrupted and indented on the next line.
		 `Radskman ip=<RadiaConfigurationServerIPaddress>,↵`
		 `    port=<RadiaConfigurationServerport>`

Code	Andale Mono	`radia_am.exe`
Selections	**Bold**	Open the **\Admin** directory on the installation CD-ROM.

Table P.3 lists the ways that various *database elements* are referenced in this book.

Table P.3 ~ Usage		
Element	**Style**	**Example**
Drives (system, mapped, CD)	Italicized placeholder	*SystemDrive*:\Program Files\Novadigm might refer to C:\Program Files\Novadigm or /opt/Novadigm/ConfigurationServer on your computer. *CDDrive*:\client\radia_am.exe might refer to D:\client\radia_am.exe on your computer.
Files (in the Radia Database)	All uppercase	PRIMARY
Domains (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE May also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE May also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

Terminology

Table P.4 lists terms that might be used interchangeably throughout this book, as well as in other Radia publications.

Table P.4 ~ Terminology[*]	
	*Depends on the context. May not always be able to substitute.
Term	**May also be called**
Application	software, service
Client	Radia Client, Radia Application Manager and/or Radia Software Manager
Computer	workstation, server
edmprof file	Configuration Server Settings File; Profile File; Profile Editor; **edmprof.dat** (Windows); **.edmprof** (UNIX)
	Note: This is the text file wherein a Radia Configuration Server's operational parameters are specified. This manual uses this non-platform specific, generic reference.
NOVADIGM domain	PRDMAINT domain
	Note: As of the 4.0 release of the database, the NOVADIGM domain is being renamed the PRDMAINT domain. Therefore, if you are using an earlier version, you will see the NOVADIGM domain in the database.
Radia Configuration Server	Manager, Active Component Server
Radia Database	Radia Configuration Server Database

Table P.5 describes the Radia DCS-specific terms that are used in this document. It is recommended that you review these terms and their descriptions in order to better understand the concepts and materials contained herein.

Table P.5 ~ Distributed Configuration Server Terminology	
Term	**Description**
RCS - Radia Configuration Server	The combined Radia and EDM Manager Databases. Also commonly referred to as *Configuration Server*.
Radia DCS - Radia Distributed Configuration Server	Formerly known as *EDM DMA*, Radia DCS is an extension of the Radia Configuration Server. It synchronizes Radia Databases that are running on separate (DCS-enabled) machines across an enterprise.
RIS - Radia Integration Server	The Radia HTTP file server that gets installed on a Source RCS in order to facilitate multiple, concurrent file-transfer sessions and the creation of the container file.
Source Configuration Server	In a Radia DCS synchronization, the *master* Configuration Server.
Destination Configuration Server	In a Radia DCS synchronization, the *target* Configuration Server.
	Note: This is always a *replica* of the Source database.

Table P.5 ~ Distributed Configuration Server Terminology

Term	Description
Synchronization	The replication of a master Radia Database (Source) with a target Radia Database (Destination).
• Peer Synchronization	Synchronizing a domain on a Destination Configuration Server from a Source Configuration Server that does not own the domain. See *Foreign-Owned Domain* in this table.
• Synchronization Pair	Two Configuration Servers that have been configured (as Source and Destination) for synchronization.
Domain Ownership	All domains must be "owned" by a Configuration Server. Domains are either *self-owned* or *foreign-owned*.
• Self-Owned Domain	A domain that is owned by the current Configuration Server. Ownership is established at installation, or when a domain is added to the database. **Note**: The *owning* MGR_ID and *current* MGR_ID must be identical.
• Foreign-Owned Domain	A domain that is owned by a Configuration Server other than the current one. **Note**: The *owning* MGR_ID and *current* MGR_ID are different.
Unrelated Domains	Domains that are not owned by the same Configuration Server—that is, they do not have the same owning MGR_ID.
Middle-tier Configuration Server	A middle-tier Configuration Server is defined to more than one DCS agent in order to affect changes across Configuration Servers that lack communications access to a single DCS agent.
Cloned Manager	Cloned Managers are Configuration Servers that share a database and a MGR_ID. They are distinguishable only by their **IP name/address**. **Note**: The database is "owned" by only one of the Configuration Servers, which can update and synchronize it.
Container File	A file, created on the Source, in which the instance data is compressed before being transferred to the Destination. This file is much faster to transfer than a large number of small files. **Note**: At the Commit phase, the instance-container file is used as the data source, so the files are moved directly from it to their ultimate destination. This minimizes the number of times that the data is moved and the length of time that the Radia Database is hard-locked.

Contents

Contents

1

Introduction to Radia DCS

AT THE END OF THIS CHAPTER, YOU WILL KNOW:

■ How the Radia DCS works to synchronize Radia Databases

■ Why there are two Radia DCS components, and the role of each in ensuring a successful synchronization

■ How to define a pair of Radia Configuration Servers for synchronization, based on *RCS eligibility* and *domain ownership*

■ The role of Radia Database domains in Radia DCS operations, as well as: *domain-naming considerations*, *domain eligibility*, and *selecting domains*

■ How to establish *domain ownership across the enterprise*, and know how to use this to set up *simultaneous synchronizations*

■ The *steps of the Radia DCS process*, which will aid in troubleshooting

Overview

Radia DCS is a powerful tool that enables you to efficiently manage multiple Radia Configuration Servers in a networked environment. Using the Radia DCS, an administrator can share information about policies and managed applications across numerous Radia Configuration Servers, thereby improving the scalability, flexibility, and extensibility of the environment.

Radia DCS

The Radia DCS product is a two-piece extension of the Radia Configuration Server. The components, *Source* and *Destination*, function separate of, although in conjunction with, one another. Both, however, have some dependence on an RCS, and therefore, each must be co-located with an RCS.

To increase functionality, both components can be installed on the same machine. For more information on the functionality of the Radia DCS components, see the section, *Radia DCS Components* starting below.

Radia DCS is designed to synchronize DCS-enabled Radia Databases throughout an enterprise, and although it is not essential that the Radia Databases directly communicate with one another, Radia DCS must be able to communicate with both Configuration Servers that comprise the *synchronization pair* (see *Two Radia Configuration Servers: A Synchronization Pair* on page 20).

In a synchronization operation, Radia DCS compares the control information of one Radia Database with that of another, for the domains that have been selected.

Radia DCS Components

Inasmuch as there are two Configuration Servers involved in Radia DCS synchronizations, the two Radia DCS components perform different functions and must be installed separately, based on the intended role of the host RCS.

Notes
Each Radia Configuration Server that will act as a Source must have the Radia DCS **Source** component installed.
Similarly, each RCS that will act as a Destination must have the Radia DCS **Destination** component installed.
If an RCS has both components of the Radia DCS installed, it can act as Source and Destination in separate Radia DCS operations.

With the DCS components installed on the RCS machines, Radia DCS:

■ Provides the synchronization facilities to contact the Source and Destination,

■ Reconciles the differences between the selected domains, and

■ Provides the intermediate facilities to make identical the Source and Destination domains.

> **Note**
>
> The Destination is always a replica of the Source.

The following section offers a more detailed look at these components and their functions.

Source vs. Destination

The Source and Destination components perform different functions during the Radia DCS synchronization. Therefore, it is important to correctly install these components in order to ensure: 1) the availability and accessibility of the appropriate Source-Destination synchronization pairs, and 2) the expected synchronization results.

Source Component

The *Source* component must be installed on any Radia Configuration Server that is going to function as the *master* in Radia DCS synchronization. This component contains *Radia Integration Server* (RIS), Radia's HTTP server.

> **Radia Integration Server Notes**
>
> For a brief description of RIS and how it relates to Radia DCS, see the section, *Radia Integration Server* below.
>
> For a detailed description of RIS, see *Chapter 2: Radia Components and Benefits*, in the publication, *Radia Essentials*, available in the Radia library.

The Source component loads the database instances into a single repository. This repository can be directly accessed, thereby eliminating the excessive overhead of opening, storing, transferring, and writing individual files for each Radia Database instance.

RADIA INTEGRATION SERVER

Radia Integration Server is Radia's HTTP file server. It facilitates multiple concurrent file-transfer sessions (HTTP "get" requests) and the creation of the instance-container file (see *Container File* in Table P.5 starting on page 10).

Radia Integration Server is not a separately licensed HP product. It integrates several independent modules (such as Radia Management Portal, Radia Proxy Server, and Radia Inventory Manager) giving them access to all the functions and resources under its control.

Destination Component

The *Destination* component must be installed on any Radia Configuration Server that is going to function as the *target* in Radia DCS synchronization. This component provides direct access to the target file system.

Two Radia Configuration Servers: A Synchronization Pair

Two Radia Configuration Servers, one defined as the Source and the other as the Destination, comprise a Radia DCS *synchronization pair*. In order for an RCS to be part of a synchronization pair, it must be defined to the Radia DCS component on the Destination RCS, as described in the section, *Configuration Servers in Radia DCS*, starting on page 54.

Radia DCS will accept only one synchronization pair per execution. Operationally, because a synchronization can go in only one direction, this means that if two Configuration Servers (for example, MGR_001 and MGR_002) need domains from one another, two Radia DCS executions must be done—with MGR_001 being the Source in one synchronization, and MGR_002 being the Source in the other.

Radia Configuration Server Eligibility

In order to be eligible to participate in a Radia DCS operation, a Configuration Server must meet the following requirements.

■   In its edmprof file it must be DCS-enabled. This is done by specifying:

```
[MGR_STARTUP]
MANAGER_TYPE=DISTRIBUTED
```

> **Note**
>
> By default, all Radia Configuration Servers are installed as DISTRIBUTED.

■   It must have either the Radia DCS Source or Destination component installed.

Domain Ownership

Radia Database domains on each Radia Configuration Server have three distinguishing characteristics: *domain name*, *owning MGR_ID*, and *current MGR_ID*.

> **Tip**
>
> For planning purposes, we recommend maintaining unique names for Radia Database domains.

■ A *self-owned* domain is a Radia Database domain that is owned by the current RCS.

   The *owning MGR_ID* and *current MGR_ID* are the same.

■ A *foreign-owned* domain is a Radia Database domain that is owned by an RCS other than the current one, and which is present as the result of a DCS synchronization.

   The *owning MGR_ID* and *current MGR_ID* are *not* the same.

Domain Naming Considerations

To minimize the likelihood of synchronization problems, consider the following points when creating domain names and configuring synchronizations.

■ A Radia Configuration Server *cannot* contain two domains with the same name.

■ An RCS *cannot* obtain one of its self-owned domains from an RCS that foreign-owns the domain.
   For example, MGR_001 cannot receive from another RCS, any domain on which it (MGR_001) is listed as the *owning MGR_ID*.

> **Notes**
>
> The version that is resident at the owner is always considered the current and correct copy.
>
> Its contents will always supersede and replace any changes introduced by other Configuration Servers.

One Owner vs. Multiple Owners

When planning domain ownership, it is helpful to consider whether to assign the proprietorship of all the domains to one Radia Configuration Server, thereby centralizing control; or to disperse control by establishing domain ownership at several Configuration Servers at various, strategic points across the enterprise.

The tables in this section detail the advantages and disadvantages of each method. For additional planning considerations, see the section, *Radia DCS Planning*, starting on page 25.

One Owning Configuration Server

Table 1.1 lists the benefits and drawbacks of *one Configuration Server* owning all the domains.

Table 1.1 ~ One Domain-Owning Configuration Server	
Advantages	**Disadvantages**
• Control of all applications, access rules, and users	• Central control might make the database very large, depending on the organization and structure
• One Radia Database to backup	• Does not align well with highly de-centralized organizations
• Data flow throughout the environment is one-way	• Data flow throughout DCS is one-way
• Aligns with highly centralized organizations	

Multiple Owning Configuration Servers

Table 1.2 lists the benefits and drawbacks of domain ownership being assigned to *multiple Configuration Servers*.

Table 1.2 ~ Multiple Domain-Owning Configuration Servers	
Advantages	**Disadvantages**
• Aligns readily with highly de-centralized organizations	• Does not align well with highly centralized organizations
• Databases are smaller and indicative of regional Source Configuration Servers	• Multiple Configuration Servers must be administered and backed-up
• Applications and users can be managed locally	• Allows for two-way data flow, adding complexity to the DCS design
• Corporate or common information can be managed centrally, while local information is managed locally	
• Allows for two-way data flow between central and local Configuration Servers	

> **Note**
>
> Any Radia Configuration Server with self-owned domains should be backed up. Foreign-owned domains can always be obtained through synchronization with the owning Configuration Server.

Domain Eligibility

The list of domains that are eligible for synchronization is dynamically compiled by Radia DCS. This list is based on the chosen synchronization pair and:

■   The database control information concerning the last synchronization for the synchronization pair, or

■   The last update with Radia administrative components (such as Radia System Explorer, Radia Packager, and Radia Publisher).

Only domains that have the same owner (on the Source and Destination) can be synchronized between that pair of Configuration Servers.

Selecting Domains

It is not necessary to synchronize all eligible domains between two DCS Configuration Servers. At the start of each session, the administrator can specify which of the eligible domains are to be synchronized.

Domain Eligibility Rules

The primary Radia DCS domain synchronization eligibility rules are listed below. These apply to each domain independently.

■   Synchronization cannot occur into a self-owned domain.
     (See log error message 1 on page 94.)

■   There is no replication into an owning Configuration Server.
     (See log error message 2 on page 94.)

> **Note**
>
> If a self-owned domain is deleted, it must be restored from a backup; it cannot be replicated from a DCS Configuration Server on which it is foreign-owned.

■   Domains that are not owned by the same Configuration Server are considered *unrelated*. A domain must be owned by the same MGR_ID at the Source and Destination in order to be eligible for synchronization.
     (See log error message 3 on page 94.)

■   Once a foreign-owned domain is locally updated with another Radia component, it cannot be used as the Source in a peer synchronization.
     (See log error message 4 on page 94.)

> **Note**
>
> A *local* update occurs when a database is updated with a Radia component (Radia System Explorer, Radia Publisher, or Radia Packager) other than DCS.

■ When it is possible to make such a distinction, the Radia DCS will prevent the regression of a *more current* Destination by a *less current* peer Source. If the Destination domain has been locally updated, and the relative currency cannot be determined, the synchronization is allowed.
(See log error message 5 on page 94.)

Radia DCS Configuration

Radia DCS must be configured for two Radia Configuration Servers. The Configuration Servers must be *defined* to Radia DCS, as described in the section, *Configuration Servers in Radia DCS*, on page 54. After the Configuration Servers have been defined, the synchronization pair and eligible domains must be selected, as described in the section, *Choosing Configuration Servers and Domains*, on page 64.

Radia DCS requires a communications connection between the Source and Destination Configuration Servers.

Radia DCS: Batch Mode

The command-line, or *batch*, mode of Radia DCS is invoked by the executable, **DMABATCH.EXE**, which can be triggered via the desktop icon, **DCS Batch**. Once the synchronization is started, it will execute "under-the-covers," with no further administrator action required. This is discussed in further detail in *Chapter 6: Radia DMABATCH*, starting on page 85.

Synchronization Logs

When a synchronization is executed, logs and objects are created. Each subsequent run causes its predecessor's logs to be overwritten, so that these logs and objects represent only the most recent run of Radia DCS.

Simultaneous Synchronizations

A Radia Configuration Server can be simultaneously involved in multiple synchronizations in which it is the Source only. This is possible because a Source database is only being read from, whereas a Destination database is being written to.

■ An RCS cannot simultaneously be a Source and Destination for different synchronizations.

■ An RCS cannot be the Destination in multiple, simultaneous synchronizations.

Radia DCS Planning

This section offers planning considerations when Radia DCS is being implemented within a Radia Configuration Server environment.

When to Use Radia DCS

The following is a list of situations that might arise in a software-management enterprise, and in which the capabilities of Radia DCS would prove beneficial.

■ To replicate Radia Database contents across an enterprise.

■ When moving domains from a test environment to a production environment.

■ As an alternative to local connects.

Developing a viable, functional Radia DCS infrastructure requires knowledge of:

■ The Radia resolution process within an environment,

■ The hardware and communications configuration of an environment, and

■ The Radia-managed information within an infrastructure.

Radia DCS Processing

Although the Radia DCS runs under-the-covers after the graphical configuration, it is helpful to know the processing that takes place after the synchronization has begun. This knowledge will aid in troubleshooting and problem resolution. The following section, *Radia DCS Processes Defined*, describes the Radia DCS processes.

Radia DCS Processes Defined

This section describes the Radia DCS processes, including the lock status of the Radia Database on the Source and Destination.

■ **Step 1: Define Configuration Servers**
Radia Configuration Servers are defined (added, copied, and deleted) for Radia DCS.

During this phase, the Source and Destination databases are *unlocked*—they are fully accessible to all Radia components to which they are defined .

■ **Step 2: Select Configuration Servers and domains to be synchronized**
the synchronization pair and eligible domains are selected.

> **Note**
>
> The synchronization pair's **property** information is saved in the **ZMANAGER** object on the Destination RCS.
>
> The **synchronization** information is saved in the **ZMGRSYNC** object on the Destination RCS.
>
> - **Note**: This synchronization pair will stay defined to Radia DCS until a new synchronization pair is selected.

The list of eligible domains is dynamically created based on the:

- Synchronization pair
- Database control information for the last synchronization, or update from a Radia component.

During this phase, the Source and Destination databases are *unlocked*—they are fully accessible to all Radia components to which they are defined.

> **Note**
>
> The next three steps run "under the covers" in the *batch* mode, DMABATCH.

- ■ **Step 3: Difference the Source and Destination control data**
  the database instances are packed into the *container file*, which is compressed.

  Radia DCS uses *instance list retrieval and differencing* to detect only those database changes that are made with Radia components.

  - The only exception to this is if an entire domain is deleted.

> **Important Note**
>
> Class and instance modifications, additions, and deletions should be done at the Source (owning) database only, because during synchronization, DCS will use domain information from the Source causing any changes that were made at the Destination to be overwritten.

  During this phase, both databases are *soft-locked*—they are read-only to other Radia components to which they are defined.

- ■ **Step 4: Download and Transfer the Data Resources from the Source**
  the *container file* is transferred from the Source to the Destination, and the instance data is written directly to its ultimate location—but with a special file extension in order to avoid overlaying.

  During this phase, both databases are *soft-locked*—they are read-only to other Radia components to which they are defined .

■ **Step 5: Commit the changes to the Destination RCS**
the database changes that were discovered as a result of the synchronization are committed to the Destination database.

During this phase, the Source database is *unlocked* and the Destination is *hard-locked*. This means that the Source is fully accessible to all Radia components, but there is no reading from or writing to the Destination until the commit is finished.

2

Installing Radia DCS

AT THE END OF THIS CHAPTER, YOU WILL HAVE:

- Successfully installed the Radia DCS **Source** component
- Successfully installed the Radia DCS **Destination** component

Two-Phase Installation

In order set up a "distributed" Radia Configuration Server synchronization environment, Radia DCS enables a Radia administrator to install:

■ The Radia DCS **Source** component,

■ The Radia DCS **Destination** component, or

■ Both Radia DCS components.

The installations are outlined in the section, *Installing the Radia DCS*, starting on page 31.

■ The Radia DCS **Source** component must be installed on any RCS whose database is going to be the *master* in Radia DCS synchronization.

■ The Radia DCS **Destination** component must be installed on any RCS that is going to be the *target* in Radia DCS synchronization.

> **Note**
>
> A Radia Configuration Server can have both components installed, in which case it can function as Source and Destination in separate Radia DCS operations.

For a detailed description of these components, see the sections, *Source Component* and *Destination Component* starting on page 19.

Recommendations and Requirements

To ensure the successful installation and operation of Radia DCS, the following system requirements are recommended.

■ Communications protocol: **TCP/IP** only.

■ Operating systems:

- **UNIX HP-UX**,

- **UNIX Solaris**,

- **Windows XP**,

- **Windows 2000**, and

- **Windows 2003 Server**.

■ Pentium processor: **120 MHz** or higher.

Radia DCS Space Requirements

The amount of free-disk space that is required by the Radia DCS components will vary because it is dependent on the number of domains selected, their size, which domains are selected, and the size of the synchronization differences.

Installing the Radia DCS

This section details the installation of the Radia DCS. In the following exercise, the Source and Destination component installations were selected.

Although this exercise is performed in a Windows environment, the UNIX steps are similar, but with the expected platform differences. Additionally, there are pre-installation steps for a UNIX environment, which are described in the next section.

UNIX Pre-Installation Notes

■   Make sure the user performing the installation has adequate UNIX operating system rights, in order to create, and update, the target installation directory.

■   Make sure the user performing the installation has a *home* directory on the UNIX workstation, and is logged in as ***root***.

To install the Radia DCS

1.   Insert the installation media CD-ROM and navigate the **extended_infrastructure** directory to the **distributed_configuration_server** installation files.

2.   Double-click **setup.exe**.

The **Radia Distributed Configuration Server Install Welcome** window opens. See Figure 2.1.

Figure 2.1 ~ The Radia Distributed Configuration Server Install Welcome window.

3. Click **Next**.

The **HP Software License Agreement** window for Radia DCS opens. See Figure 2.2.
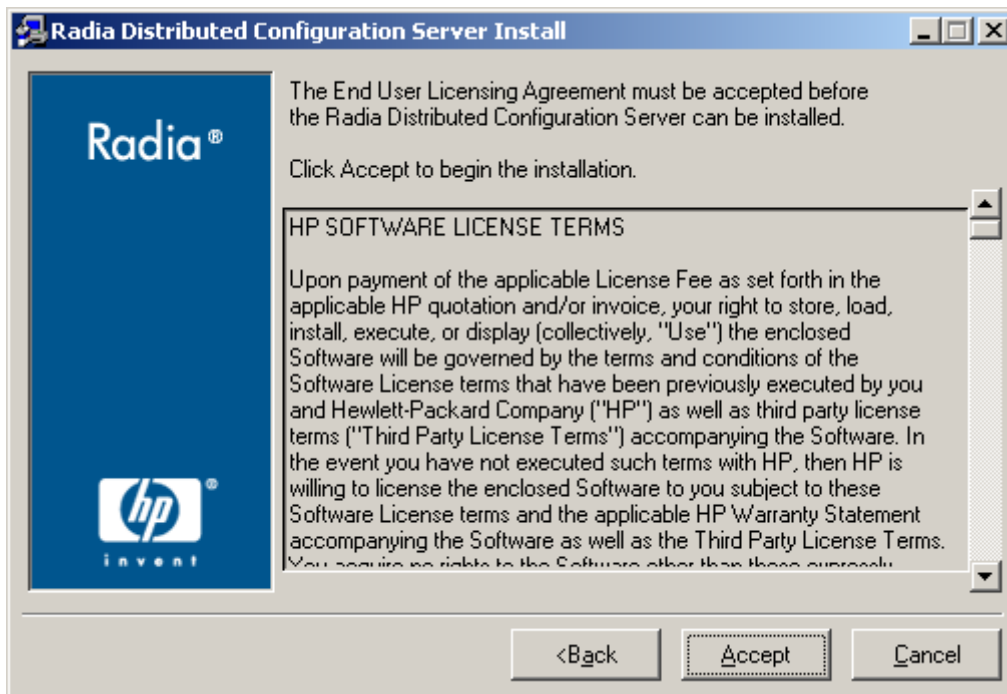
Figure 2.2 ~ The HP Software License Agreement window.

4. Click **Accept**.

Note
If **Accept** is not selected, the Radia DCS installation program will terminate.

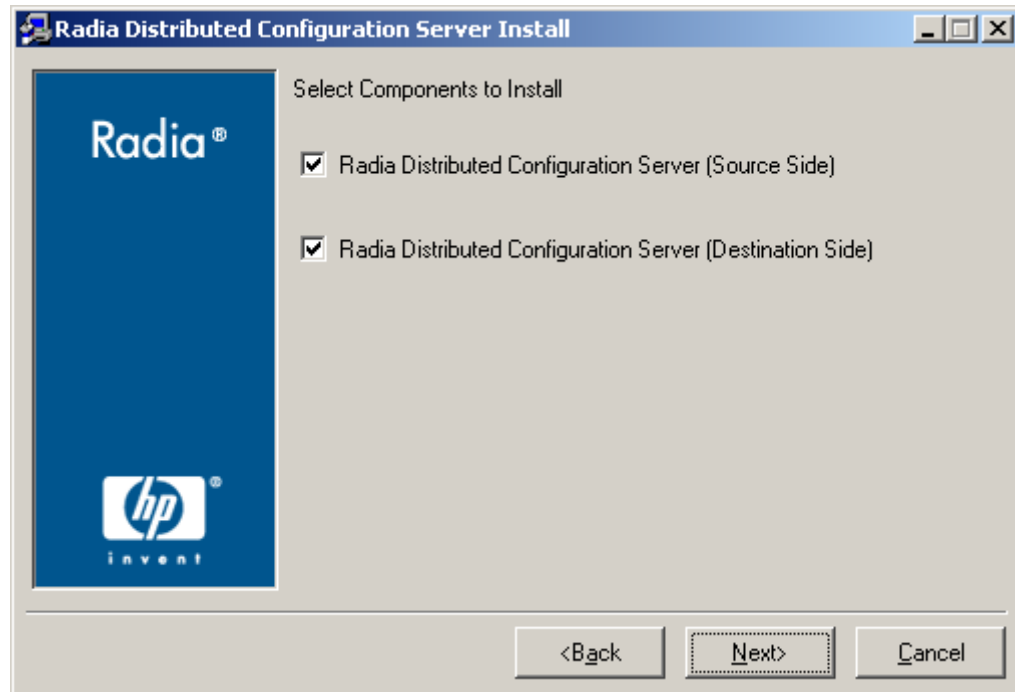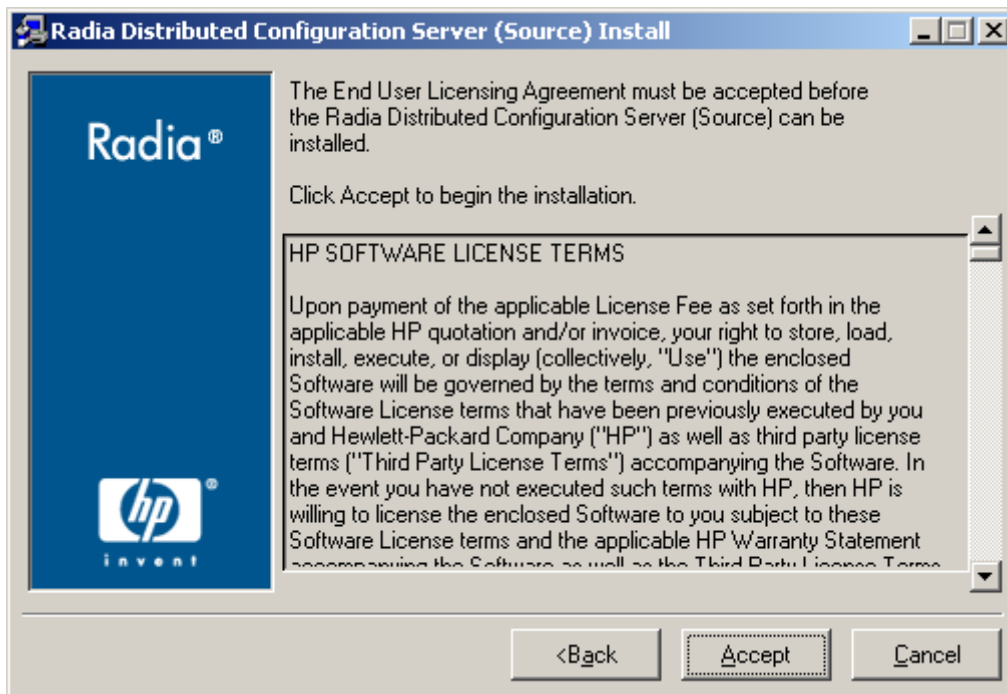The **Radia DCS Installation Component Selection** window opens. See Figure 2.3.

Figure 2.3 ~ The Radia DCS Install component selection window.

- Select either or both of the Radia DCS components

5. Click **Next**.

The **Radia DCS Installation Summary** window opens. See Figure 2.4.
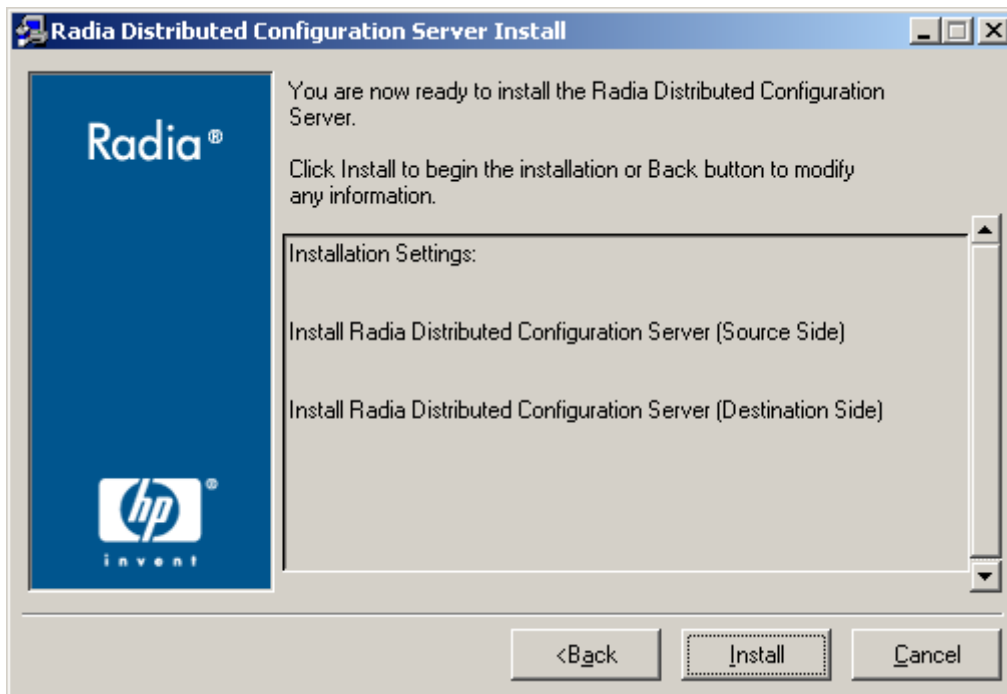
Figure 2.4 ~ The Radia DCS Install Summary window.

The **Summary** window displays which Radia DCS components will be installed.

● To change the selections, click **Back** and make the necessary changes.
● To install the displayed components, click **Install**.

The standard "transferring files" window will display. After a brief interval, the **Radia Distributed Configuration Server (Source) Install Welcome** window. See Figure 2.5.

Installing the Radia DCS Source Component

This section describes the installation of the Source component of Radia DCS.
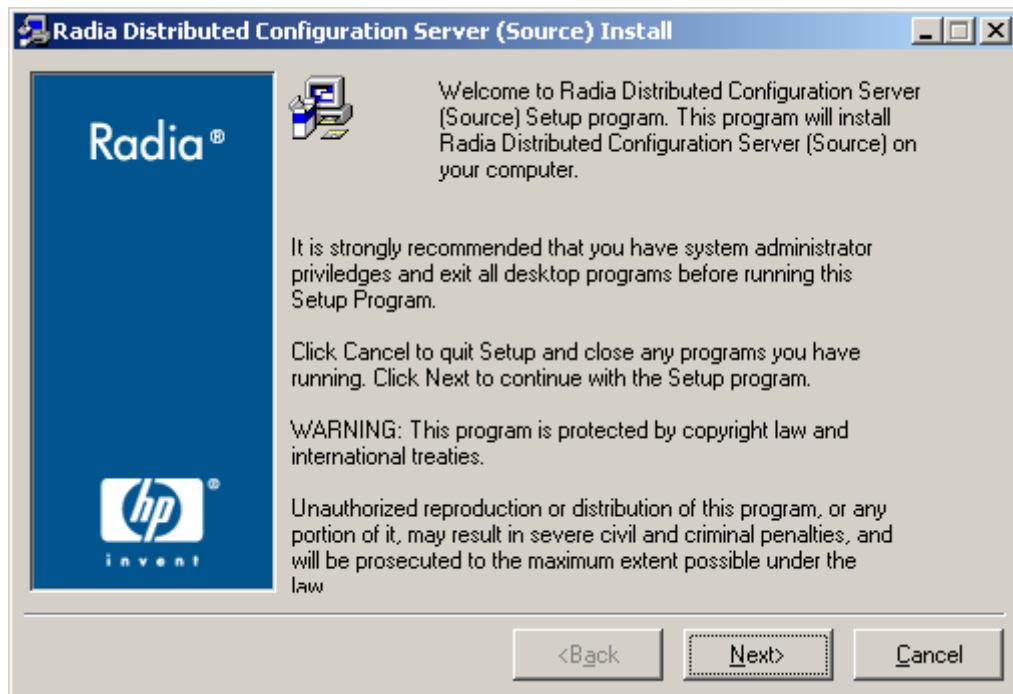
Figure 2.5 ~ The Radia DCS (Source) Install Welcome window.

6. Click **Next**.

The **HP Software License Agreement** window for the Radia DCS Source component opens. See Figure 2.6.

Figure 2.6 ~ The HP Software License Agreement window.

7. Click **Accept**.

> **Note**
>
> If **Accept** is not selected, the Radia DCS installation program will terminate.

The **Radia DCS (Source) Install Database Path** window opens. See Figure 2.7.
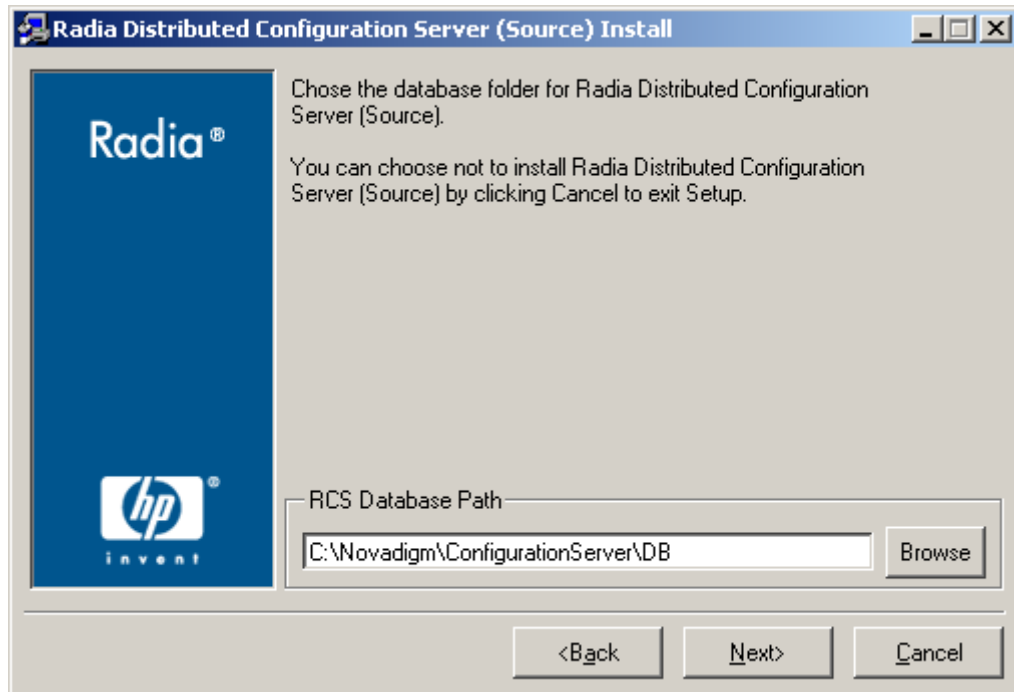
Figure 2.7 ~ The Radia DCS (Source) Install Database path window.

8. The **RCS Database Path** field displays the directory in which the Radia Database was installed.

- Accept the path that is displayed in the window; or
- Specify a different location; or
- Click **Browse** to navigate to a different location.

9. Click **Next**.

The **Radia DCS (Source) Install Summary** window opens. See Figure 2.8.
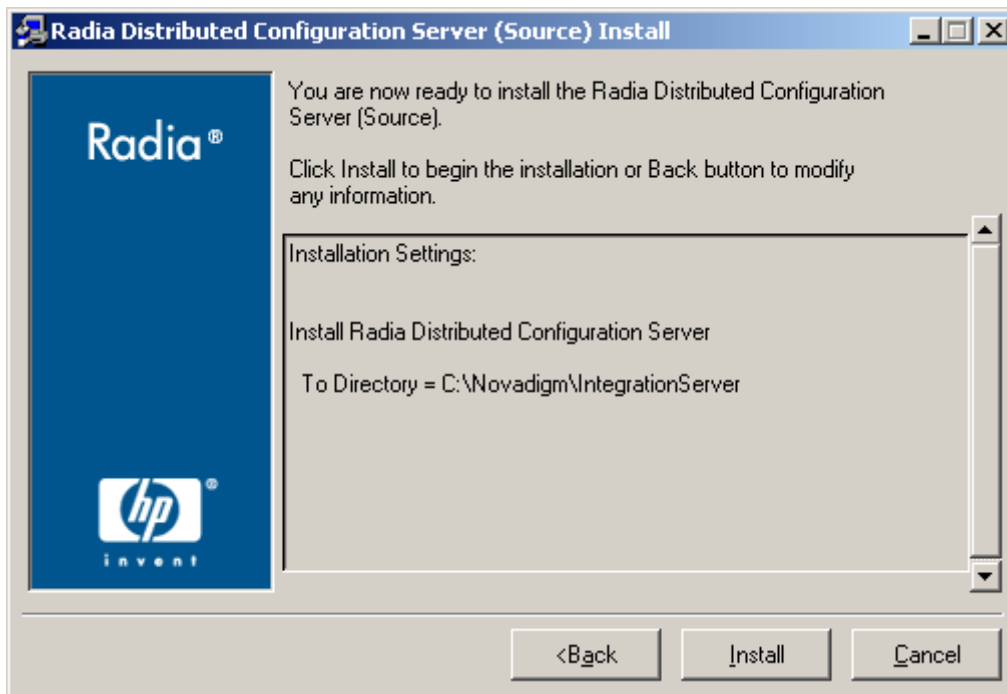
Figure 2.8 ~ The Radia DCS (Source) Install Summary window.

The **Summary** window displays the directory into which the Radia DCS Source component will be installed.

● To change the selections, click **Back** and make the necessary changes.

● To accept the specified settings, click **Install**.

The **Radia DCS (Source) Install Finish** window opens. See Figure 2.9 on page 40.
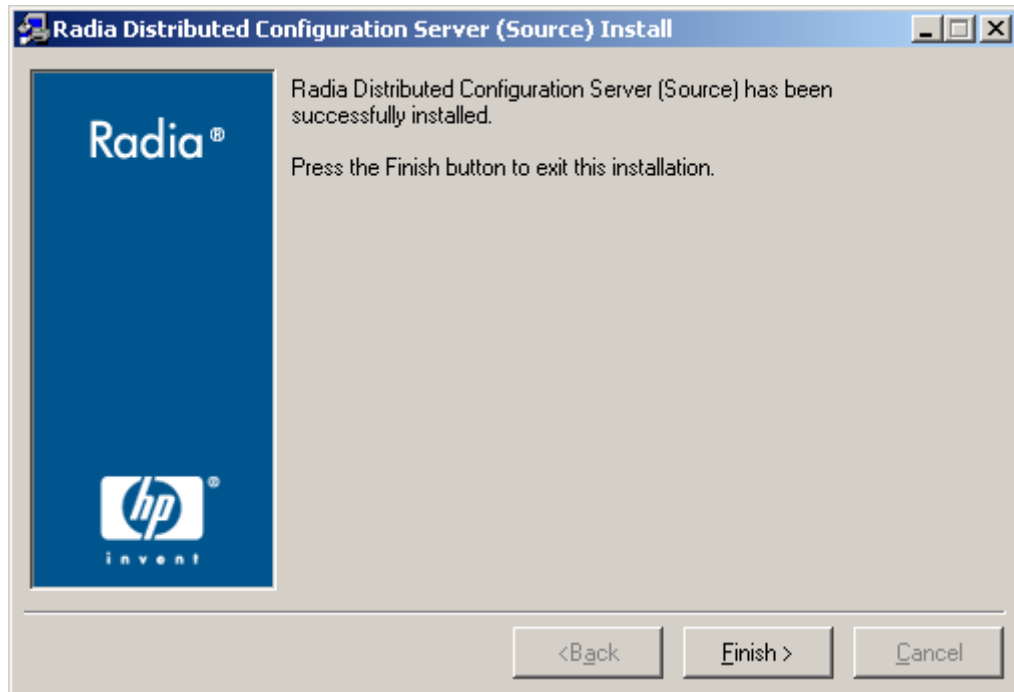
Figure 2.9 ~ The Radia DCS (Source) Install Finish window.

10. Click **Finish**.

The Source component of Radia DCS has installed successfully.

> **Note**
>
> If the Radia DCS Destination component installation was also selected, it will automatically start now.

Installing the Radia DCS Destination Component

This section describes the installation of the Destination component of Radia DCS.

Figure 2.10 ~ The Radia DCS (Destination) Install Welcome window.

11. Click **Next**.

The **HP Software License Agreement** window for the Radia DCS Destination component opens. See Figure 2.11.
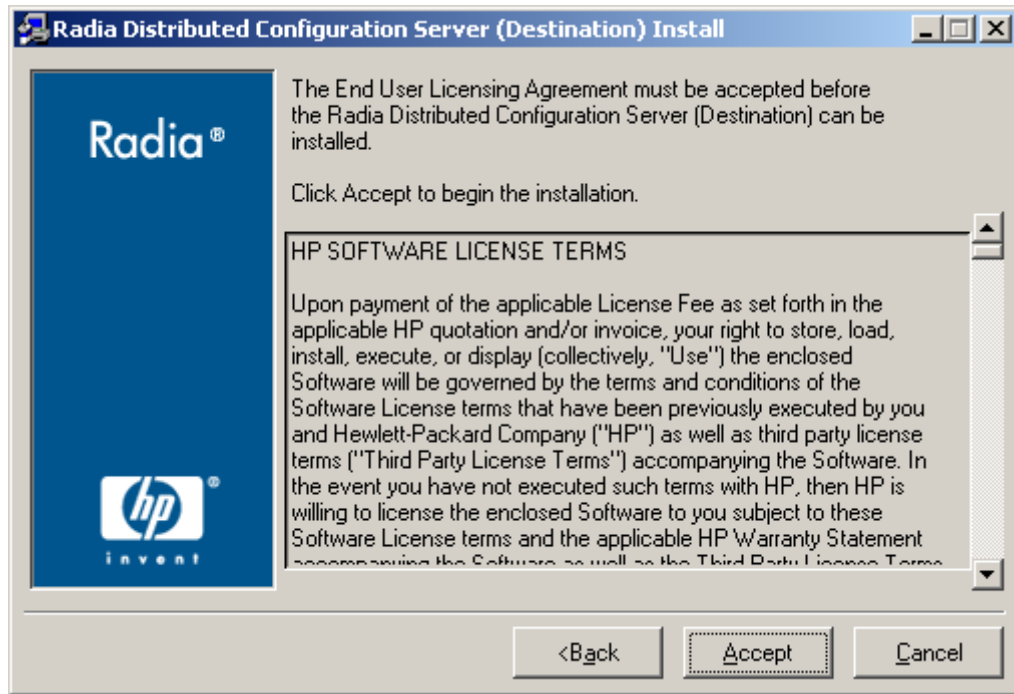
Figure 2.11 ~ The HP Software License Agreement window.

12. Click **Accept**.

Note
If **Accept** is not selected, the Radia DCS installation program will terminate.

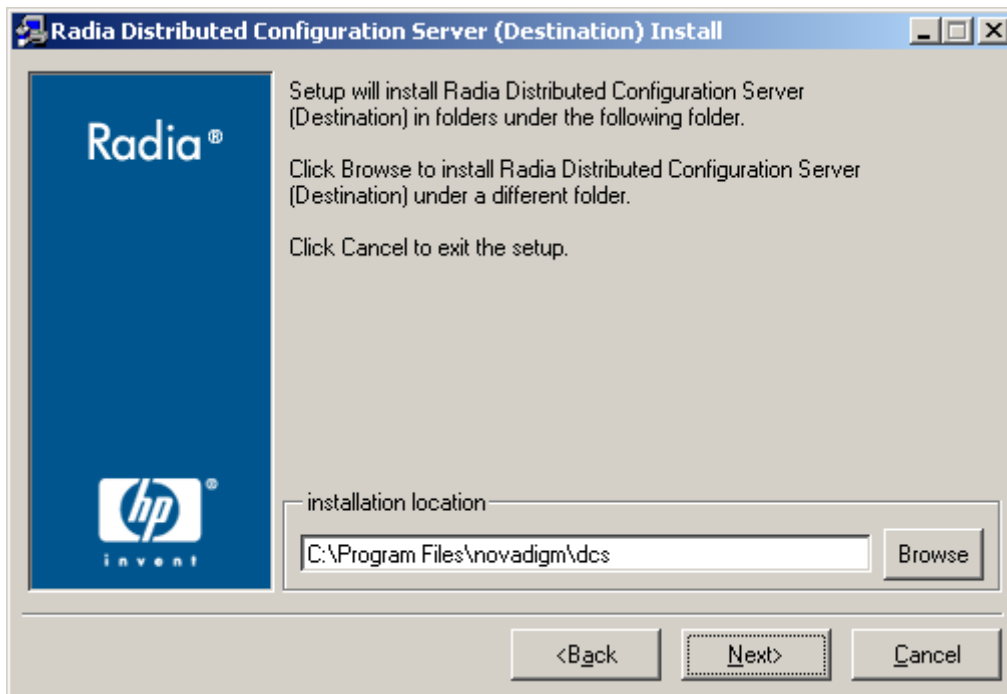The **Radia DCS (Destination) Install Installation Location** window opens. See Figure 2.12.

Figure 2.12 ~ The Radia DCS (Destination) Install Installation Location window.

13. The **Installation Location** field displays the directory in which the Radia DCS Destination component's files will be installed.

♦ If the installation program detects any type of existing Radia element (such as a Radia Client, Radia Administrator Workstation, or a previous version of Radia DCS), the window that appears will have one text field, as shown in Figure 2.12.

♦ If the installation program detects no existing Radia element, the window that appears will have **Object Location** and **Log Location** fields under the **Installation Location** field.

Note
In either case, a message will appear, warning that the directory will be updated.
Click **OK** to proceed and allow the update, or click **Cancel** to return to the **Installation Location** window and specify a different directory.

● Accept the default path that is displayed; or

- Specify a different location; or
- Click **Browse** to navigate to a different location.

14. Click **Next**.

The **Radia DCS (Destination) Install Summary** window opens. See Figure 2.13.
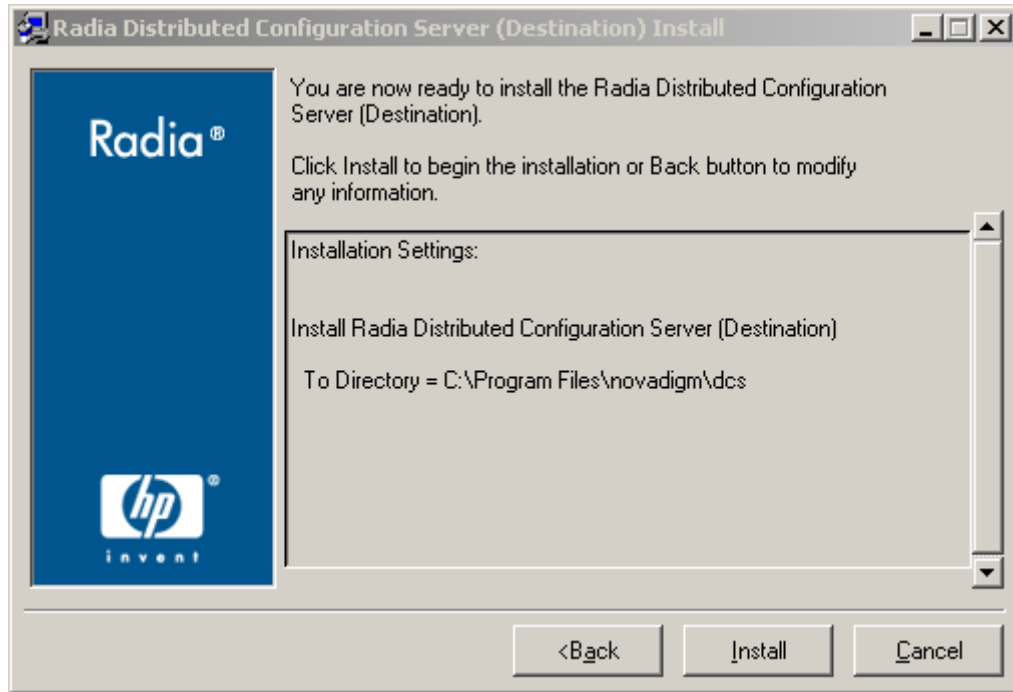

Figure 2.13 ~ The Radia DCS (Destination) Install Summary window.

The **Summary** window displays the directory into which the Radia DCS Destination component will be installed.

- To change the selections, click **Back** and make the necessary changes.
- To accept the specified settings, click **Install**.

The **Radia DCS (Destination) Install Finish** window opens (Figure 2.14).
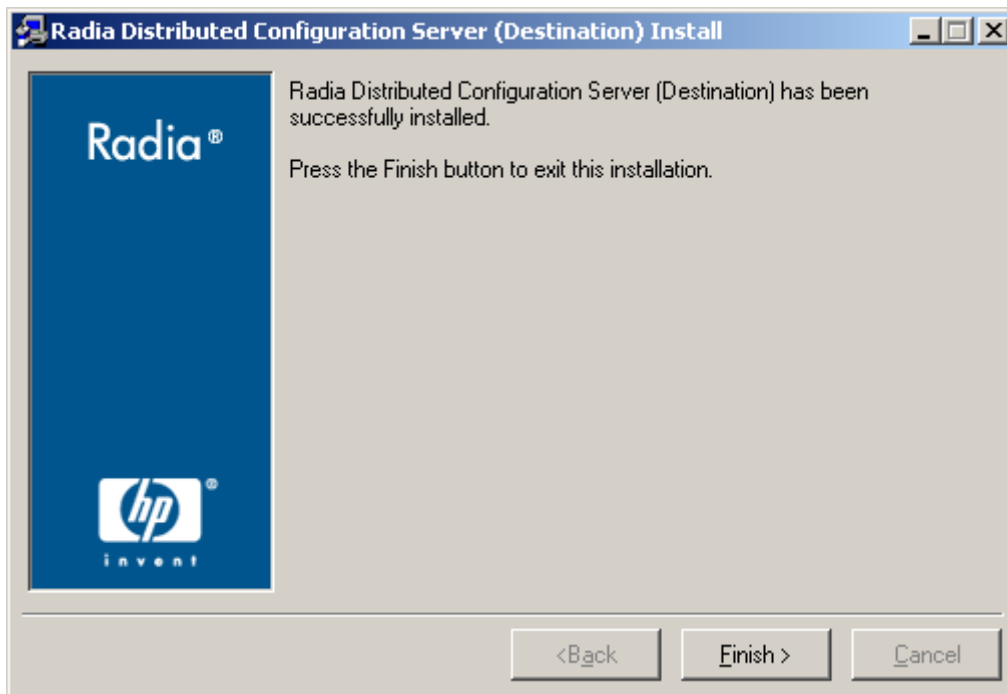
Figure 2.14 ~ The Radia DCS (Destination) Install Finish window.

15. Click **Finish**.

The Destination component of Radia DCS has installed successfully.

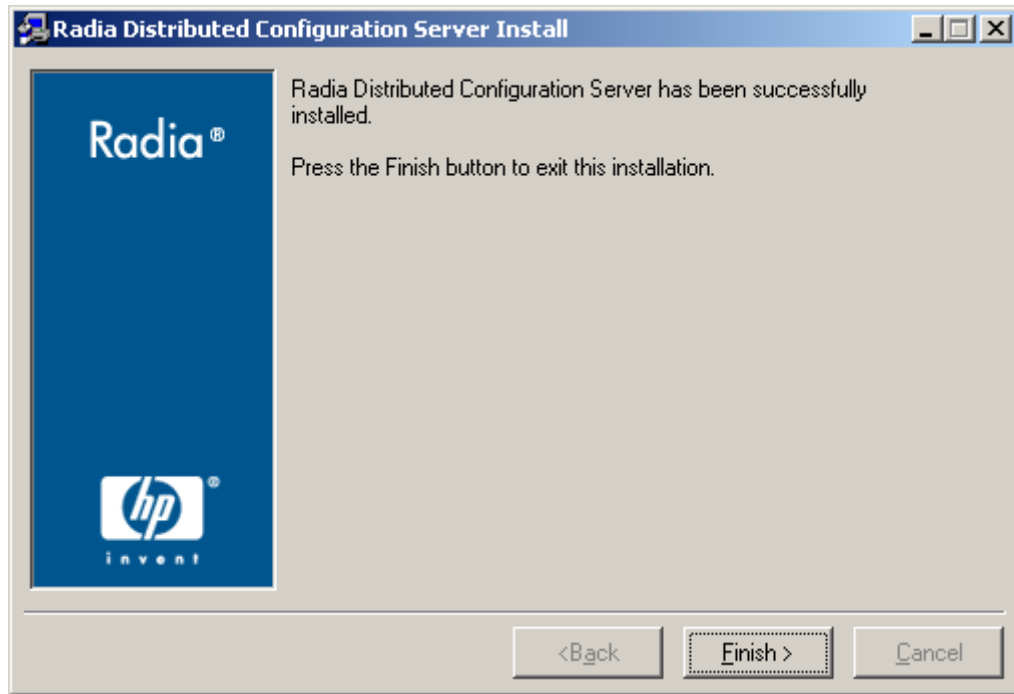The **Radia Distributed Configuration Server Install Finish** window opens. See Figure 2.15 on page 46.

Figure 2.15 ~ The Radia Distributed Configuration Server Install Finish window.

16. Click **Finish**.

Both components of Radia DCS have installed successfully.

Notes
Windows
· This installation might create a new (or update an existing) **nvd.ini** file in **C:\Program Files\Novadigm\lib**.
· If there was an nvd.ini file under **C:\Program Files\Novadigm\lib**, it gets renamed to **nvd.ini.old**.
UNIX
· This installation might create a new (or update an existing) **~/.edmprof**.
· If there was an .edmprof file, it gets renamed to **.edmprof.old**.

3

Radia DCS Security

AT THE END OF THIS CHAPTER, YOU WILL BE ABLE TO:

■   Set up password protection for Radia DCS synchronizations by assigning security control to the host operating system

Setting up Security

Radia DCS has an optional security feature that enables an administrator to assign password protection to one or both of the synchronization-pair's Radia Databases, using native operating-system security.

Native Operating-System Security

This section details the assignation of password protection to the native operating system.

A special user ID and password are used to access secured Radia Databases. Radia DCS defines only one user ID and password. Therefore, all secured Radia Databases that Radia DCS might access must:

■ Be defined in their host's security system,

■ Have the user ID in their ADMIN_LIST, and

■ Have the same password for that user ID.

> **Note**
>
> The user ID and password are defined in the **Batch Security** area of the **Batch Security** tab (see Figure 3.2 on page 50) of the **Radia DCS Options** panel.

Enabling Native Operating-System Security

The following sections describe the steps that are required in order to assign password protection-security to the native operating system.

To set up password protection in Radia DCS

1. On the Destination RCS, open the **Radia Distributed Configuration Server (Config Mode)** dialog box by clicking **Start**, **Programs**, **Radia Distributed Configuration Server (Destination side)**, and **Configure**.

The **Radia Distributed Configuration Server (Config Mode)** dialog box.
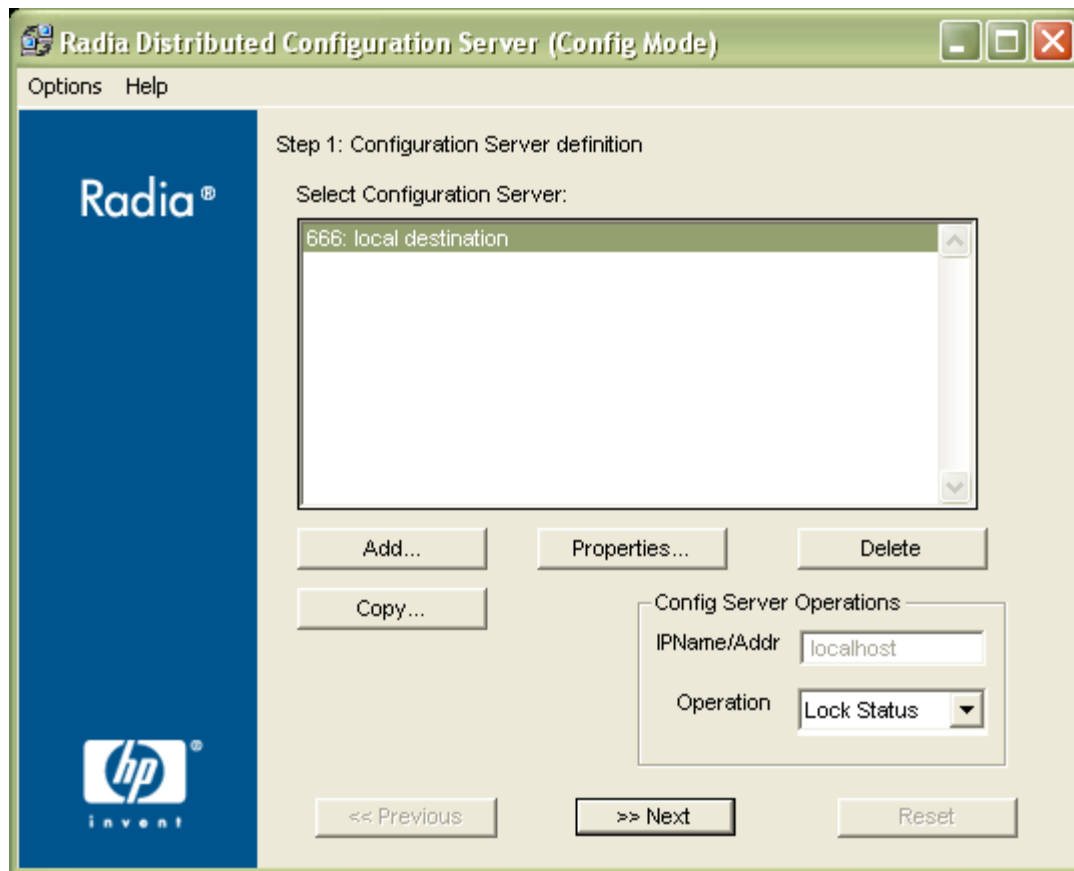
Figure 3.1 ~ The Configuration Server definition panel.

2. Highlight the Configuration Server that is to be assigned password protection.
3. Click **Options**.
4. Click **Options…**.
   The **Radia DCS Options** panel appears.
5. Click the tab, **Batch Security**.
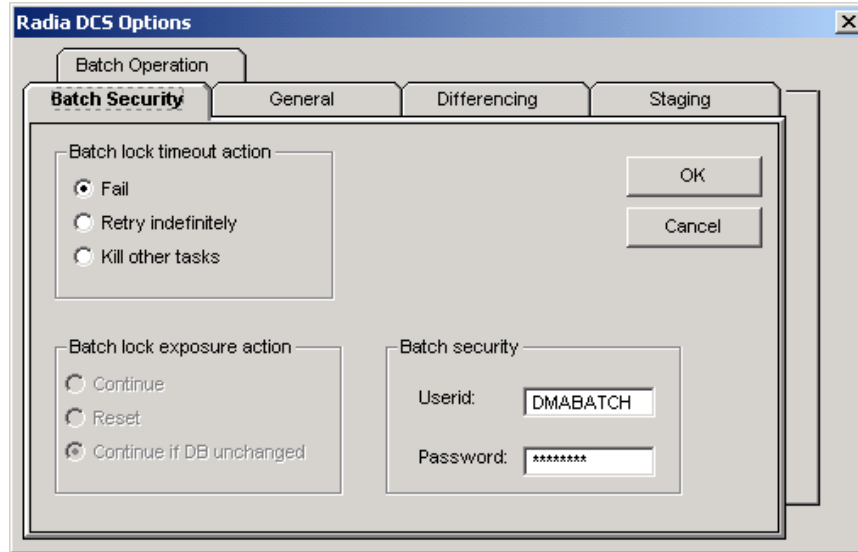   The **Radia DCS Options**, **Batch Security** tab is displayed.

Figure 3.2 ~ The Radia DCS Options, Batch Security tab.

6. In the **Batch security** area, specify a **user ID** and **Password** to be used for password protection.

> **Note**
>
> The default user ID and password are **DMABATCH**.

7. Click **OK**.

You are returned to the **Configuration Server definition** dialog box.

8. Close Radia DCS.

Password protection has been defined.

Configuration Server Security Settings

In addition to the steps outlined in the previous section, the MGR_DMA section must be added to the edmprof file, as described in this section.

> **Note**
>
> The MGR_DMA section is not included in the edmprof file at Radia Configuration Server installation because it is not needed for default operations.
>
> It can be added to the edmprof file in order to configure Radia DCS as a default function of the RCS.

To modify the edmprof file

1. Bring down the RCS.

2. Open the edmprof file using a text editor.

3. Add the section, MGR_DMA, and the settings shown below:

   ```
   [MGR_DMA]
   SECURITY_METHOD = EDMSIGN
   ADMIN_LIST = <list_of_administrators>
   ```

   For a description of these settings, see Table 3.1 below.

4. Save the changes, close the edmprof file, and restart the Configuration Server.

> **Note**
>
> The administrators that are specified for ADMIN_LIST must have *user rights* under local policy settings on the host operating system (for example, **Act as part of the operating system** on **Windows NT**).
>
> For information on establishing operating system-specific user rights and policies, consult the operating system's product documentation.

Table 3.1 presents a description of the two MGR_DMA settings that are required to establish native operating-system security.

Table 3.1 ~ MGR_DMA Settings and Values

Setting	Explanation of Value
SECURITY_METHOD	Optional. If not specified, security verification is *disabled*. To enable native operating-system security, specify **EDMSIGN**.
ADMIN_LIST	Required if a SECURITY_METHOD is specified. Specify the list of administrators (user IDs) that are allowed to use Radia DCS on this Radia Database. The format is a comma-separated (no spaces), case-sensitive list of operating system account names.

4

Setting Up a Radia DCS Synchronization

AT THE END OF THIS CHAPTER, YOU WILL KNOW HOW TO:

■   Define (add, copy, and delete) Radia Configuration Servers to Radia DCS

■   Specify the properties of the Configuration Servers that are defined to Radia DCS

■   Set up a Radia DCS operation by choosing a synchronization pair and eligible domains

Configuration Servers in Radia DCS

This section describes how, in Radia DCS, to add, copy, and delete Configuration Servers.

Adding Configuration Servers

To add Configuration Servers in Radia DCS

1. On the Destination RCS, open the **Radia Distributed Configuration Server (Config Mode)** dialog box by clicking **Start**, **Programs**, **Radia Distributed Configuration Server (Destination side)**, and **Configure**.

The **Configuration Server definition** dialog box appears.
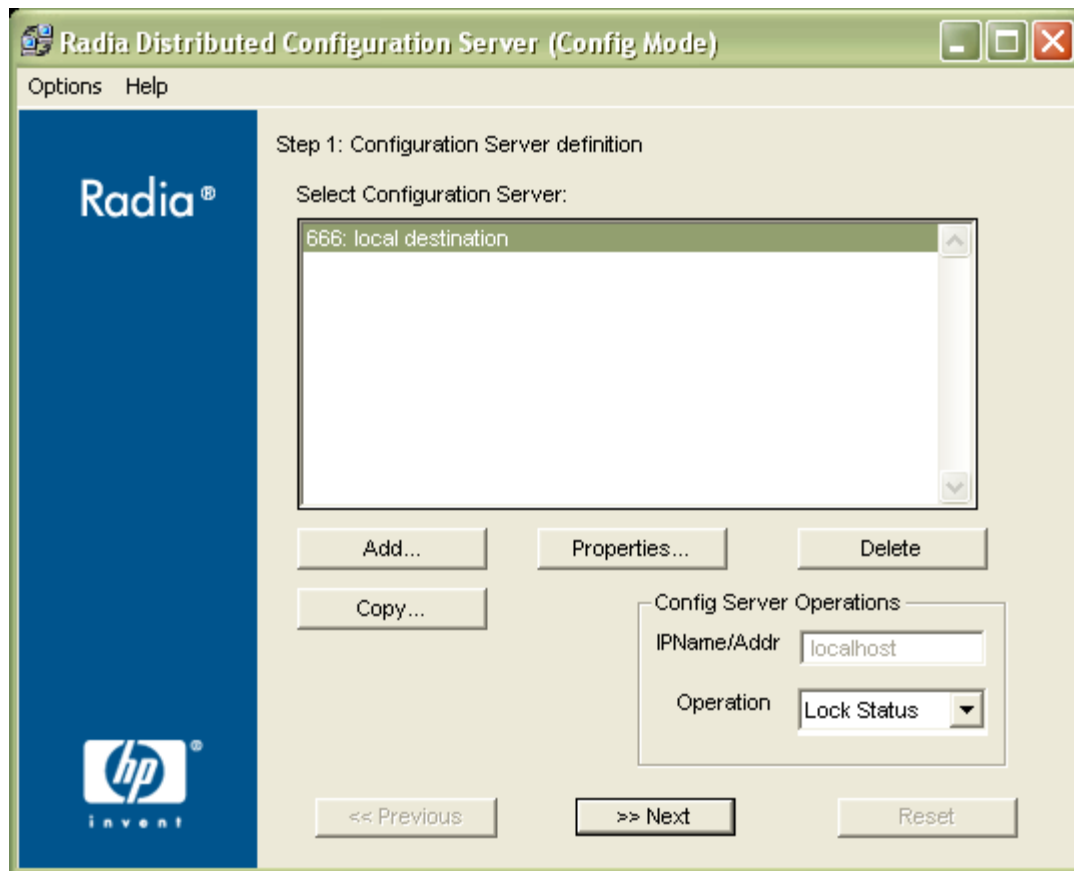
Figure 4.1 ~ The Configuration Server definition panel.

Notes

If a previous version of Radia DCS/EDM DMA exists, the previously defined Radia
Configuration Servers will appear.

2. Click **Add**.

The **Configuration Server Properties** dialog box appears.
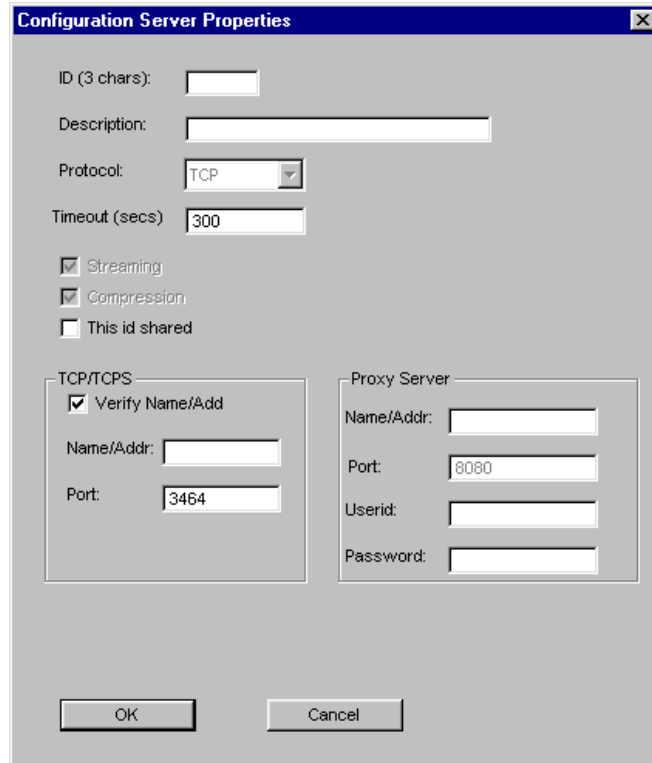
Figure 4.2 ~ The Configuration Server Properties dialog box.

3. Specify the appropriate Configuration Server information, as shown in Table 4.1 below.

Table 4.1 ~ Configuration Server Properties		
Field	**Description**	**ZMANAGER Variable**
ID (3 chars.)	Using the characters 0-9 and A-F specify the three-character, hexadecimal ID for the Radia Configuration Server. This must match the value of MGR_SETUP.MGR_ID for the selected RCS. **Exception**: The 256 consecutive positions from **F00** through **FFF** are reserved for use with Radia.	ZMGRID
Description	Specify an alphanumeric (255 characters maximum) RCS description that will readily identify this RCS. For example, **DocServer_001**. **Note**: This description is independent of the RCS name that was assigned at installation, but can be the same.	ZMGRNAME

Table 4.1 ~ Configuration Server Properties		
Field	**Description**	**ZMANAGER Variable**
Protocol	This field is disabled. The only supported protocol is **TCP/IP**.	ZCOMTYPE
Timeout	Specify a timeout (in seconds) for how long DCS is to wait to complete a task. **Note**: If DCS times out before the tasks end, it will abort.	ZTIMEO
Streaming	This option is disabled. The only value for streaming is ON.	N/A
Compression	This option is disabled. The only value for compression is ON.	N/A
This ID shared	Select this check box to enable the **IPName/Addr** field in the **Config Server Operations** area of the **Configuration Server definition** panel. **Note**: This must be selected in order to support *Cloned Managers* (see Table P.5, on page 10).	N/A
TCP/TCPS		
• Verify Name/Addr	Check this option in order to have Radia DCS verify that the host name or IP address that is specified for **Name/Addr** is known in the host's table. (Optional)	N/A
• Name/Addr	Specify the IP name, IP address, or URL of the host Radia Configuration Server. (Required)	ZTCPADDR
• Port	Specify the IP port of the host Radia Configuration Server. (Required)	ZTCPPORT
Proxy Server	This option is disabled.	N/A

4. After specifying the properties for the RCS, click **OK**.

A Radia Configuration Server has been added to Radia DCS.

Notes
To define additional Configuration Servers to Radia DCS, repeat steps 2 and 3, or use the **Copy** option that is described in the next section, *Copying Configuration Servers*.

Copying Configuration Servers

In addition to adding Radia Configuration Servers to Radia DCS, they can be defined by copying the definition of an existing one, and then modifying it.

To copy Configuration Servers in Radia DCS

1. On the Destination RCS, open the **Radia Distributed Configuration Server (Config Mode)** dialog box (Figure 4.1).

2. Select the Radia Configuration Server that is to be duplicated, and click **Copy**.

The **Configuration Server Properties** dialog box appears (Figure 4.2).

The **ID** field will be blank.

3. Type a unique, 3-character, hexadecimal ID.

> **Note**
>
> Only a unique ID will be accepted; if a duplicate ID is specified, an error message will appear.

4. Modify any of the other fields as needed. (For assistance, refer to Table 4.1 on page 56.)

5. Click **OK**.

A Radia Configuration Server has been copied within Radia DCS.

Deleting Configuration Servers

To delete Configuration Servers from Radia DCS

1. With the **Configuration Server definition** dialog box open, highlight the Radia Configuration Server that is to be deleted.

2. Click **Delete**.

A confirmation message will appear to verify that the specified MGR_ID should be deleted.

3. If the wrong RCS was selected, click **Cancel** and select the correct RCS.

To delete the RCS specified, click **Delete**.

The RCS will be removed from the **Select Configuration Server** list on the **Configuration Server definition** dialog box.

4. Close Radia DCS.

A Radia Configuration Server has been deleted from the Radia DCS agent.

List of Configuration Servers

Each time the **Configuration Server definition** dialog box is accessed, Radia DCS will display the list of Configuration Servers that are defined to it (see Figure 4.3, on page 60).
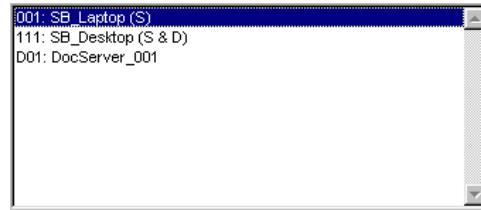
Figure 4.3 ~ A list of the Configuration Servers that are defined to Radia DCS.

Radia DCS Configuration

The configuration-only interface allows an administrator to set up the parameters of the synchronization, which must then be run in the batch mode. The interface is comprised of two panels, as described in the sections, *Configuration Server Definition Panel* (starting on page 61) and *Choose Configuration Servers and Domains Panel* (starting on page 63).

Navigation Buttons and Menu Options

Both of the Radia DCS configuration dialog boxes have *navigation* buttons in the lower part, and two *menu* options (**Options** and **Help**) in the upper left corner.

Navigation Buttons

The navigation buttons (**Previous**, **Next**, and **Finish**) allow for movement between the panels.

> **Note**
>
> To stop Radia DCS without completing the configuration, click the standard **Close** button (**X**) on the title bar.

Menu Options

In the upper left corner of the Radia DCS windows, there are two menu options (**Options** and **Help**).

■   **Help** offers one option, **About**, which presents Radia DCS version information.

■   **Options** offers one option, **Options**, which presents the five-tab **Radia DCS Options** panel (shown in Figure 4.4).
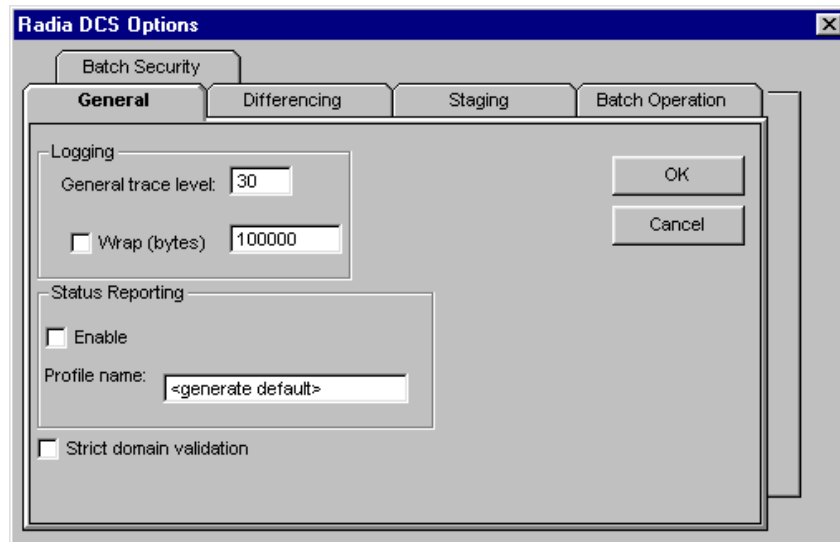
Figure 4.4 ~ The five-tab Radia DCS Options panel.

The Radia DCS configuration options are discussed in detail in *Chapter 5: Configuring Radia DCS Options*, starting on page 71.

Configuration Mode Panels

The **Config Mode** provides a two-panel interface that takes an administrator through the configuration of the Radia DCS synchronization process. The two panels are:

■   Configuration Server Definition Panel (see the section below)

■   Choose Configuration Servers and Domains Panel (see page 63)

Configuration Server Definition Panel

In the first panel (Figure 4.5), Radia Configuration Servers are defined (added, copied, and deleted), as described in the section, *Configuration Servers in Radia DCS*, starting on page 54; and their lock-status operations can be queried, as described in the section, *Operation: Lock Status*, starting on page 63.
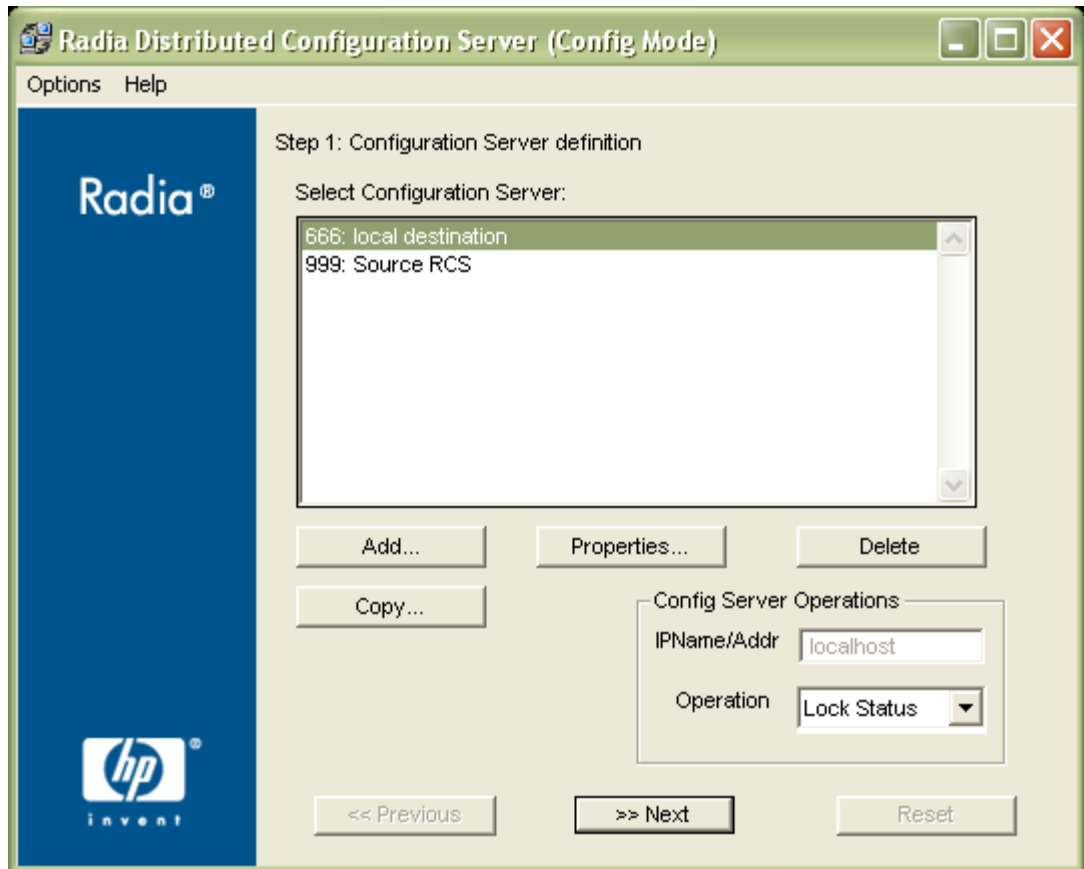
Figure 4.5 ~ The Configuration Server definition panel after an RCS has been added.

Configuration Server Operations

In the **Config Server Operations** area, a Radia Configuration Server's lock-status can be queried and its IP address overridden.
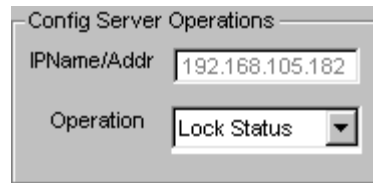
Figure 4.6 ~ The Config Server Operations area.

IPNAME/ADDR

In this field, an administrator can specify an override for the selected Radia Configuration Server's IP name/address. The default is the IP name/address that is configured in ZMANAGER.

This field is disabled and displays the IP address of the Configuration Server that is highlighted in the **Select Configuration Server** field. The field will become enabled if, on the **Configuration Server Properties** panel, the option **This id shared**, has been selected for that Configuration Server.

> **Note**
>
> This supports Configuration Server operations on cloned Managers (see Table P.5 on page 10).

OPERATION: LOCK STATUS

This lets an administrator query the lock status of an RCS.

1. Highlight an RCS in the **Select Configuration Server** field.

2. Click the down arrow next to **Lock Status**.

3. Click **Lock Status**.

Shortly, a message will pop up, reporting the database lock-status of the selected RCS.

Choose Configuration Servers and Domains Panel

In the second panel, an administrator selects the synchronization pair and the Radia Database domains that are to be included in the synchronization, as described in the section, *Choosing Configuration Servers and Domains*, starting on page 64.
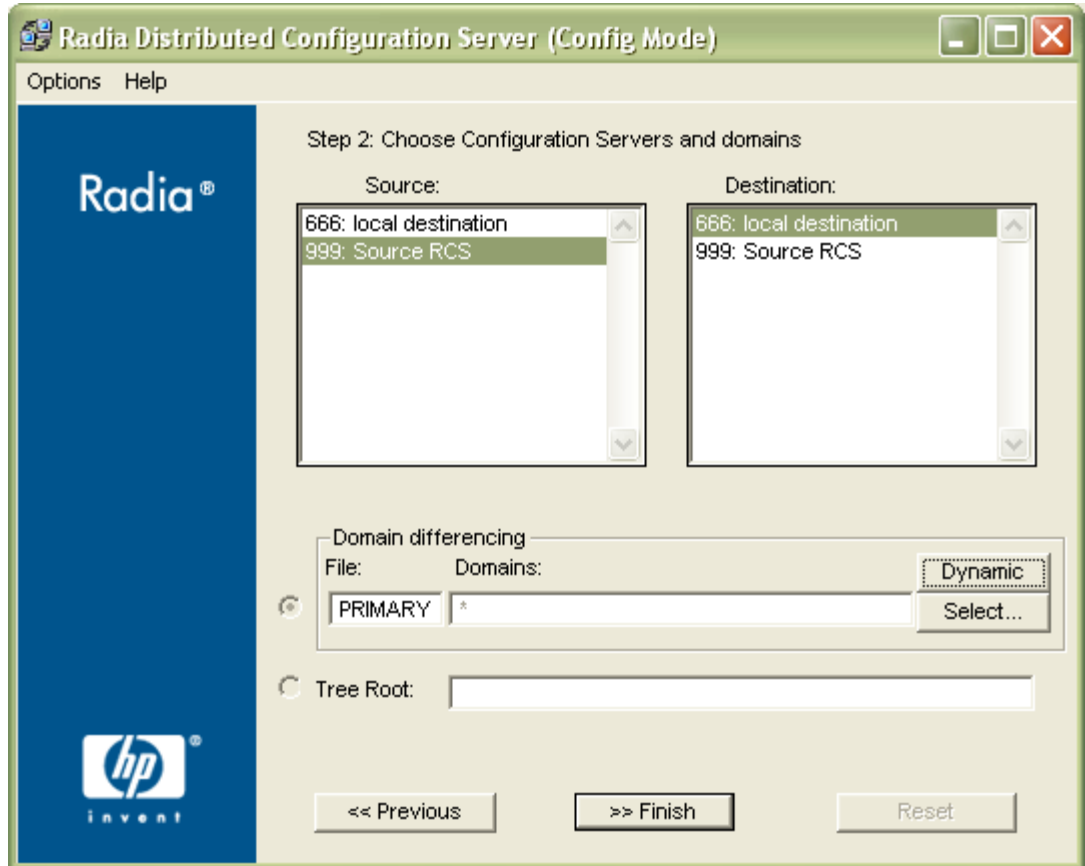
Figure 4.7 ~ The Choose Configuration Servers and domains panel.

Note
In this window, the buttons **Previous** and **Finish** are activated.

With these two steps completed, the Radia DCS synchronization is ready to be run in the batch mode, which is detailed in *Chapter 6: Radia DMABATCH*, starting on page 85.

Choosing Configuration Servers and Domains

The two Radia Configuration Servers that will participate in a DCS session are the synchronization pair. Once the synchronization pair is selected, the list of eligible domains is

generated based on the Configuration Servers and their database control information, such as last synchronization and updates from other Radia components.

To set up a synchronization pair

1. Open the **Radia Distributed Configuration Server (Destination Component) Config Mode** panel.

2. Click **Next**.

    The **Choose Configuration Servers and domains** panel appears.

    The list of eligible Configuration Servers that have been defined to Radia DCS will be listed in the **Source** and **Destination** fields of this panel.

3. Select one Source and one Destination Configuration Server.

4. In the **Domain Differencing** area (Figure 4.8),

Figure 4.8 ~ The Domain Differencing area.

click either:

- **Select** – to manually specify which domains are to be synchronized. (Continue with **step 5**.)

    or…

- **Dynamic** – to allow Radia DCS to automatically synchronize all of the eligible domains. (Click **Finish**, the synchronization is configured and ready to be run.)

> **Note**
>
> **Dynamic domain selection** – a list of all eligible domains will be automatically generated at run-time. (All domain-eligibility rules still apply.) If new domains are created after the configuration and before the synchronization, they are automatically included in the synchronization.
>
> If **Dynamic** is selected, an asterisk (*****) will be displayed in the **Domains** field.

With both domain-selection options, a list of eligible domains is built, based on the two Configuration Servers that have been chosen.

Exception

If there are no eligible domains, a message will appear.

If this message appears unexpectedly, the ownership assigned for the expected domains might be incorrect at either of the Configuration Servers. Shut down Radia DCS and the associated Configuration Servers, and correct the domain ownership using the ZEDMAMS utility, UPDATE_MGRIDS. After correcting the domain ownership, restart the Configuration Servers and resume DCS.

If **Select** was chosen, when the eligible-domains list is complete, the **Domain Selection** dialog box (Figure 4.9) appears.
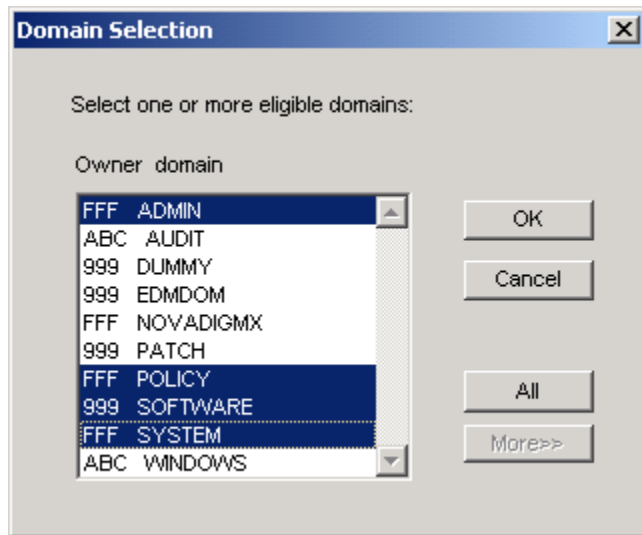
Figure 4.9 ~ The Domain Selection dialog box.

The eligible domains are listed alphabetically, with the owning MGR_ID, in the **Domain Selection** dialog box.

5. Select one or more of the eligible domains.

6. Click **OK**.

The domains that were selected for synchronization are now displayed in the **Domains** field of the **Choose Configuration Servers and domains** panel.
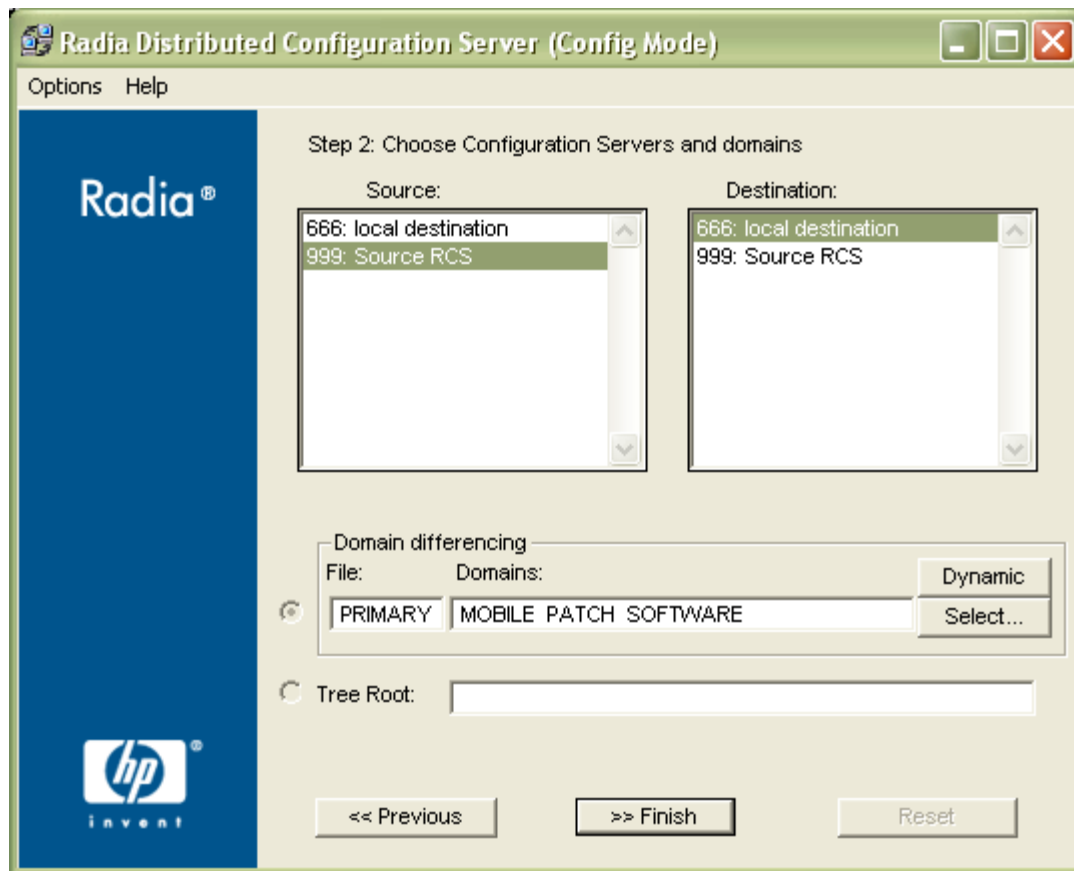
Figure 4.10 ~ The Choose Configuration Servers and domains panel with domains specified in the Domains field.

7. Click **Finish** to exit the configuration of this Radia DCS synchronization.

> **Note**
>
> If either Configuration Server requires password setup that has not been assigned, a message will appear indicating that Radia DMABATCH cannot be run.

A Radia DCS synchronization has been configured.

The Radia Configuration Server's EDMPROF File

The edmprof file is the text file where the Radia Configuration Server's operational parameters are configured and stored. Two of its sections (MGR_STARTUP and MGR_DMA) are integral to enabling Radia DCS and ensuring its proper operation.

Information on these edmprof sections, including their settings, acceptable values, and their impact on DCS processing is presented in this section.

Note
For a comprehensive look at the edmprof file, refer to *Chapter 1: Radia Configuration Server Settings* in the *Radia Configuration Server Guide*.

MGR_STARTUP Section

The MGR_STARTUP section dictates startup behavior for the Configuration Server. The following MGR_STARTUP settings are essential to the operation of the Radia DCS.

Table 4.2 ~ MGR_STARTUP Settings and Values		
Setting	**Designated Value**	**Explanation**
MANAGER_TYPE	DISTRIBUTED	This is the default value that is established when the Radia Configuration Server is installed. **Note**: To ensure that an RCS is DCS-enabled, do not change this value.
MGR_NAME	32 alphanumeric characters (max.)	This is an RCS identifier. **Note**: This value is independent of the Configuration Server names that are listed in the **Select Configuration Server** area of the **Configuration Server definition** window.
MGR_ID	Unique, 3-digit, hexadecimal ID	A Configuration Server's unique identifier. This must match the ID that is specified in the **ID** field of the **Configuration Server Properties** dialog box. • Radia DCS uses this value to generate object IDs in the Radia Database. • Each character in this identifier can have the values 0-9 and A-F. **Exception**: The 256 consecutive positions from **F00** through **FFF** are reserved for use with Radia.
TCP_PORT	Port number on which the Configuration Server is configured to run.	Must match the DCS-specified port.

MGR_ID

The MGR_ID setting is used to establish a unique identity for each Radia Configuration Server. Radia DCS uses the MGR_ID to determine each domain's *owning* RCS. (All domains have an owning RCS.) Domain ownership is important because in order for a domain to be eligible for synchronization, its owning MGR_ID must be the same on the Source and Destination RCS machines. If they are not, synchronization cannot occur.

> **Note**
>
> Although the MGR_ID must match for both domains, it is possible that neither the Source nor the Destination is the owner. See *Peer Synchronization*, in Table P.5 on page 11.

MGR_DMA Section

In addition to the MGR_DMA settings that are needed to establish Radia DCS password protection (described in the section, *Configuration Server Security Settings* on page 50), there is another DCS-related setting, DMA_TIMEOUT, which is detailed in Table 4.3.

Table 4.3 ~ MGR_DMA Settings and Values	
Setting	**Explanation of Value**
DMA_TIMEOUT	Specify how long (in seconds) Radia DCS is to wait for non-DCS tasks to complete before applying a lock to the Radia Database. If Radia DCS times out before the task ends, it will abort. The default is **0**.
	When soft-locking the Radia Database, Radia DCS must wait for all administrator tasks to end.
	When hard-locking the Radia Database, Radia DCS waits for all non-DCS tasks to end.
	Note: If Radia DCS is unable to lock either of the databases, it will query the value of **Batch Lock Timeout** (see Table 5.8 on page 82).

5

Configuring Radia DCS Options

AT THE END OF THIS CHAPTER, YOU WILL BE ABLE TO:

■ Use the five **Radia DCS Options** tabs to specify actions to be taken by Radia DCS during the database synchronization

Radia DCS Options

Radia DCS has a configuration area, **Radia DCS Options**, which has five tabs that enable a Radia administrator to specify a variety of processing actions, ranging from report generation to database-lock timeout actions. The **Radia DCS Options** are comprised of the following tabs:

■ *The General Tab*, on page 73
   logging and status reporting options

■ *The Differencing Tab*, on page 78
   options to generate a differencing report and to skip database resources

■ *The Staging Tab*, on page 79
   option to skip missing resources

■ *The Batch Operation Tab*, on page 80
   synchronization verifying and session failure options

■ *The Batch Security Tab*, on page 82
   options for password protection and database-lock timeouts

The following sections describe these options.

The General Tab

On the **General** tab, configure: *logging options*, *status reporting*, and *domain validation*.
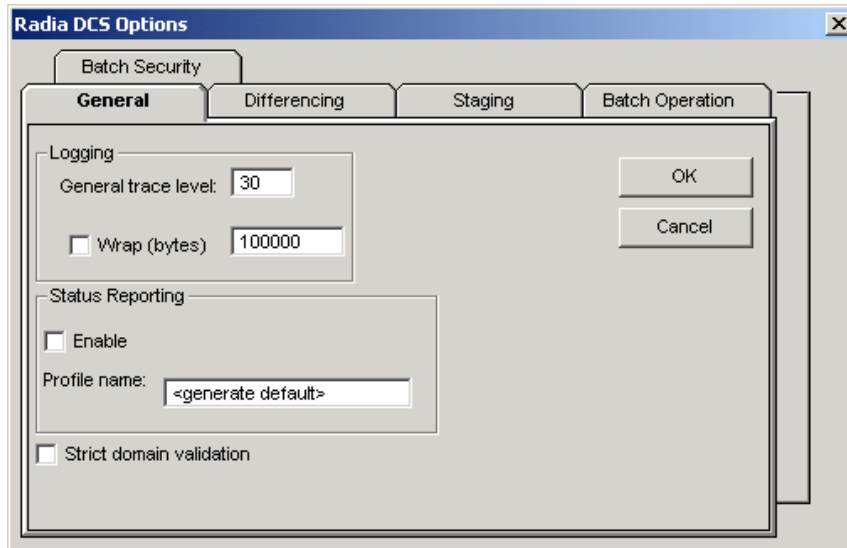
Figure 5.1 ~ The Radia DCS Options: General tab.

Table 5.1 describes the options available on this tab.

Table 5.1 ~ General Tab Options		
Option (Object Variable)	**Settings (defaults)**	**Description**
Logging		
• General Trace Level (ZTRACEL)	0 – 999 (**30**)	The level of detail for tracing.
• Wrap (LOGWRAP)	**0**=**disable** 1=enable	Select to enable (1) log wrapping.
• Number of Bytes (LOGSIZE)	**100,000**	Number of bytes at which log text will wrap.

Table 5.1 ~ General Tab Options

Option (Object Variable)	Settings (defaults)	Description
Status Reporting		This allows Radia DCS to send, at several processing points, a DMASTATS object (see Table 5.2 on page 75) to the Source Configuration Server. This object contains data about the state and configuration of DCS. **Note**: To save these objects for future reference, use the Radia System Explorer to configure the Source Configuration Server to put them in the PROFILE file of the database, using the PUTPROF command, as described in the section, *Using PUTPROF* below.
• Enable (REPORT)	**0**=**disable** 1=enable	The status-reporting facility is enabled by selecting the **Enable** check box.
• Profile Name (REPTNAME)	<generate default>	See the section, *ZUSERID* starting on page 77.
Strict Domain Validation (STRICT)	When local database updates occur and neither Configuration Server owns the domain, two *peer validation* rules govern how the updates are handled. By default, this option is **disabled**. • The domain must not be locally updated on the Source after it was last synchronized from the owner. This would propagate updates not present at the owner. • The Destination domain must not have been synchronized from a Source domain version that has been updated more recently than that of the owner. Doing so would regress the version at the Destination. **Note**: With the current Radia Database, these rules cannot be correctly enforced if the Configuration Servers are not all in the same time zone, or if their clocks are not reasonably close. These rules are now enforced *optionally*, by selecting **Strict domain validation**.	

USING PUTPROF

1. In ZSYSTEM.ZPROCESS create a new instance, such as DMASTATS.

- Specify the *Method* attribute as:

  `ZSYSTEM.ZMETHOD.PUTPROF_DMASTATS`

2. In ZSYSTEM.ZMETHOD create a new instance, such as PUTPROF_DMASTATS.

- Specify the *Parameter* attribute as:

  `DMASTATS`

- Specify the *Method Name* attribute as:

  `EDMMPPRO`

> **Tip**
>
> Each execution of DCS might generate several reporting objects at various points in the
> processing (see the section, *DMASTATS* below). Each of these reporting objects will
> overwrite the previous one.
>
> In order to see every reporting object, configure ZSYSTEM.ZPROCESS.DMASTATS to invoke
> a customized REXX method.

DMASTATS

Table 5.2 defines the fields of the DMASTATS object.

Table 5.2 ~ DMASTATS Fields Defined	
Field	**Definition**
BATCHDAT	Date of this report
BATCHTIM	Time of this report
BATSTDAT	Date of correlated starting (id=1) report
BATSTTIM	Time of correlated starting (id=1) report
BATCHRC	Character return code (if REPORTID > 1) **Notes**: Refer to Table 5.3 on page 76 for REPORTID values. Refer to Table 5.4 on page 76 for detailed BATCHRC information.
BATCHMSG	Completion message (if REPORTID > 1) **Note**: Refer to Table 5.3 on page 76 for REPORTID values.
BATARGS	DMABATCH command line
BATLKSTA	Result of **DMABATCH ACTION=LOCKSTATUS**: • **U** (Unlocked) • **S** (Soft-locked) • **X** (Exclusive Soft-locked) • **H** (Hard-locked)
DMASTATE	• **0** = Initial • **1** = Compared • **2** = Downloaded • **3** = Committed
ZSRCMGID	Source MGR_ID
ZDSTMGID	Destination MGR_ID
REPORTID	Identifies which DCS processing point sent the report. **Note**: Refer to Table 5.3 on page 76 for detailed REPORTID information.
SCOPE	Scope of synchronization=**DOMAIN**
ZDOMAINS	List of domains
ZUSERID	User name for use with PUTPROF method (see *ZUSERID* on page 77)

Table 5.3 identifies which DMASTATS.REPORTID processing point sent the report.

Table 5.3 ~ REPORTID Values Defined	
REPORTID	**Definition**
1	Starting
2	Differencing completed, differences found
3	Differencing completed, no differences found
4	Staging completed
5	Commit completed
6	Ending

Every DMABATCH execution sends REPORTIDs **1** and **6**. In addition, synchronizations might send REPORTIDs **3** or **2**, **4**, and **5** for intermediate status.

Table 5.4 lists the return codes that can be expected from DMASTATS, and what each means.

Table 5.4 ~ BATCHRC Values Defined	
BATCHRC	**Definition**
000	Completed OK.
003	Radia DCS timed-out waiting for tasks to end.
001-090	Configuration Server errors. See the Configuration Server log.
099	Client internal errors. See the Radia DCS log.
111	ACTION=RESET failed.
112	ACTION=DBVERIFY failed.
113	Differencing failed.
212	Client timeout.
218	Unable to establish communications to a Configuration Server.
999	Synchronization in progress (when REPORTID < 6).

ZUSERID

If a value is specified for **Profile name** on the **General** tab, ZUSERID uses that value. This name can be:

■   a 32-character (max.) alphanumeric name. If it is longer than 32 characters, it will be truncated.

■   US national characters, such as @, $, #, and _ are allowed.

If no value is specified for **Profile name**, the default, **<generate default>**, will be used. In this case, ZUSERID is generated based on one of the following.

■   If:

```
SCOPE=DOMAIN
```
DMA_<*src_id*>_<*dest_id*>_**DOMS**_<*domains*>

where <*domains*> is an underscore-separated list of domains in this sync, or ALL if ZDOMAINS=*. For example,

```
DMA_100_203_DOMS_SOFTWARE_POLICY
```

■   If a special batch operation:

DMA_<*target_id*>_<*action*>

```
DMABATCH ACTION=LOCKSTATUS MGRID=123
```

generates

```
DMA_123_LOCKSTATUS
```

> **Note**
>
> <*target_id*> can be independent of <*src_id*> and <*dst_id*>.

The Differencing Tab

On the **Differencing** tab, specify whether to enable *resource skipping* and *report generation*.
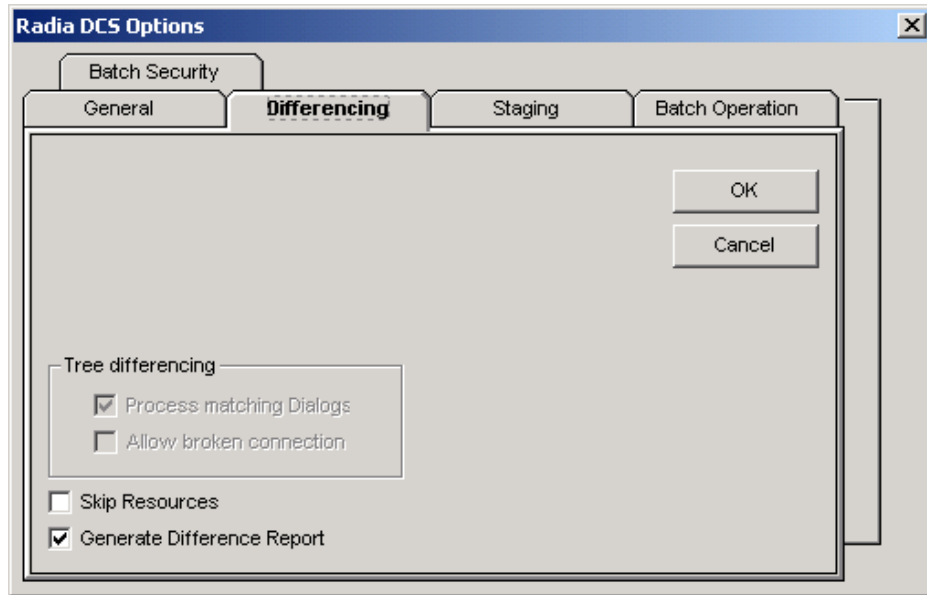
Figure 5.2 ~ The Radia DCS Options: Differencing tab.

Table 5.5 describes the options available on this tab.

Table 5.5 ~ Differencing Tab Options		
Option (Object Variable)	**Settings (defaults)**	**Description**
Skip Resources (NORES)	**0**=**use resources** 1=skip resources	This is an option to ignore resources during differencing.
Generate Difference Report (DIFFREPT)	0=disable **1**=**enable**	Enabling this option will result in a report of the database differencing being generated.

The Staging Tab

On the **Staging** tab, instruct Radia DCS whether to overlook *missing database resources*.
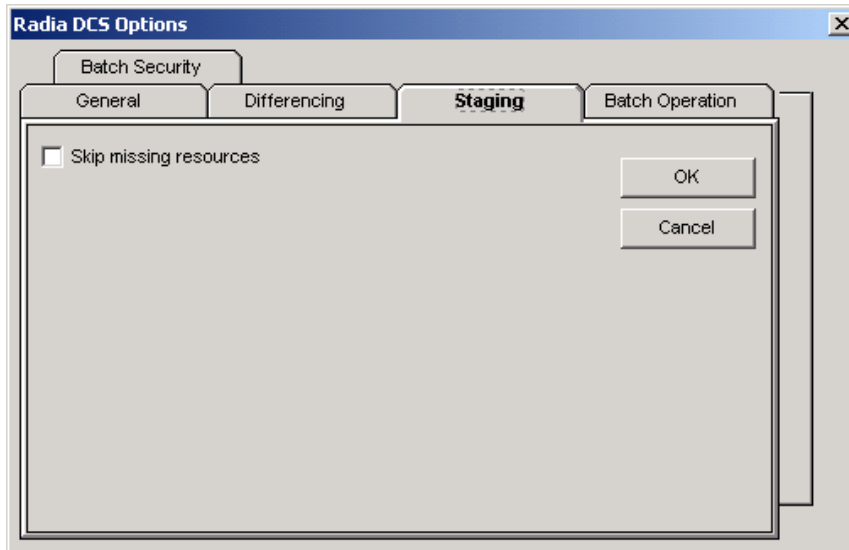
Figure 5.3 ~ The Radia DCS Options: Staging tab.

Table 5.6 describes the options available on this tab.

Table 5.6 ~ Staging Tab Options		
Option (Object Variable)	**Settings (defaults)**	**Description**
Skip Missing Resources (SKIPMISS)	1=enable **0=disable**	Skips missing resources. **Note**: By default, if the Staging step does not find a resource on the Source, it will terminate. This option allows Staging to skip the resource and continue.

The Batch Operation Tab

On the **Batch Operation** tab, enable or disable *synchronization verification* and *session resetting*.
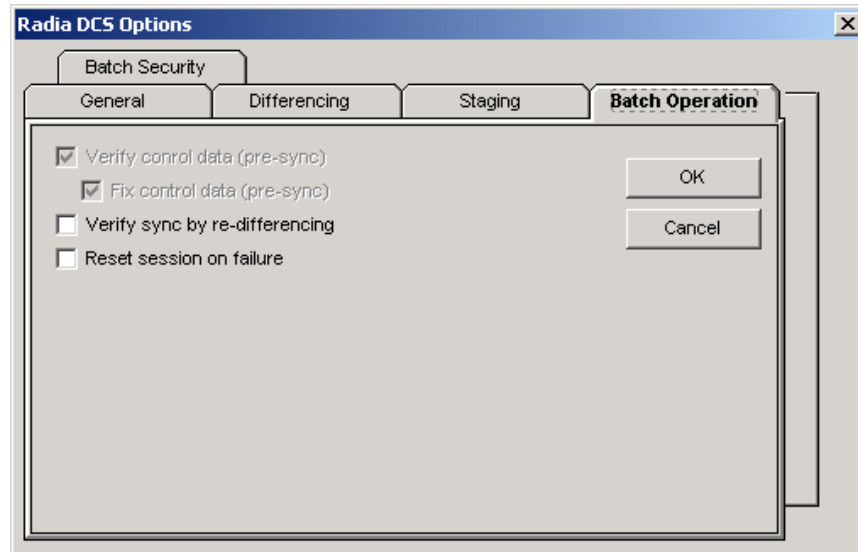
Figure 5.4 ~ The Radia DCS Options: Batch Operation tab.

Table 5.7 describes the options available on this tab.

Table 5.7 ~ Batch Operation Tab Options		
Option (Object Variable)	**Settings (defaults)**	**Description**
Verify Control Data	N/A	This option is disabled.
• Fix Control Data	N/A	This option is disabled.
Verify Synch by Re-differencing (SYNCHVFY)	**0**=**disabled** 1=enabled	Verify the synchronization results by re-running the differencing. If any additional differences are found, it restarts the process, and the differences are, in turn, transferred to the Destination.

Table 5.7 ~ Batch Operation Tab Options

Option (Object Variable)	Settings (defaults)	Description
Reset Session on Failure (BATRESET)	**0**=**disabled** 1=enabled	If synchronization fails, reset it to its initial status. **Notes**: Normally, if synchronization fails during staging (for example, due to a lost connection), it is left in a state that ensures that it can subsequently be restarted from the point of failure. This means leaving both Configuration Servers locked. If leaving the Configuration Servers locked, pending a restart, is not acceptable, this option allows the session to be reset to the initial state, and immediately unlocks both Configuration Servers, releasing staged resources. Any subsequent synchronization of this synchronization pair will start from scratch; if staging failed near the end of a long process, all of that processing is lost.

The Batch Security Tab

On the **Batch Security** tab, specify *database-lock actions* and *security settings*.
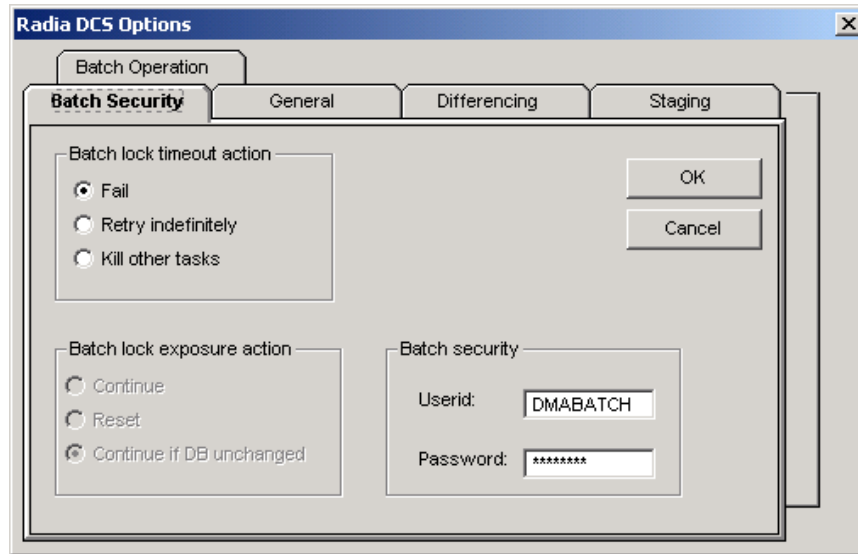
Figure 5.5 ~ The Radia DCS Options: Batch Security tab.

Table 5.8 describes the options available on this tab.

Table 5.8 ~ Batch Security Tab Options		
Option (Object Variable)	**Settings (defaults)**	**Description**
Batch Lock Timeout Action (BATLOKTO)	**X** – (**Fail**) R – (Retry) F – (Force)	Specify what to do in the event of a batch-lock timeout. **Note**: A **Batch Lock Timeout** occurs when active non-DCS tasks running on the Destination RCS prevent Radia DCS from achieving a hard-lock at the Commit step.
• Fail	• X	Cancel the current database-lock attempt, and report a BATCHRC=003.
• Retry Indefinitely	• R	Continue retrying until a database hard-lock is obtained.
• Kill Other Tasks	• F	Terminate all non-DCS tasks that are running, obtain the hard-lock, and continue processing.
Batch Lock Exposure Action	N/A	This option is disabled—its behavior is hard-coded to **Continue if DB unchanged**.

Table 5.8 ~ Batch Security Tab Options

Option (Object Variable)	Settings (defaults)	Description
Batch Security		
• User ID (BATUSER)	(**DMABATCH**)	User ID for all Radia Configuration Servers in batch mode.
• Password (BATPWD)	(**DMABATCH**)	Password for all Radia Configuration Servers in batch mode.

6

Radia DMABATCH

AT THE END OF THIS CHAPTER, YOU WILL KNOW:

■ The DMABATCH command-line arguments that can be used in a script that executes DMABATCH and handles error conditions

■ The time-saving automated solutions that can automatically handle error and non-error situations during Radia DCS batch processing

■ The DMABATCH synchronization-message variable, BATCHMSG, and return-code variable, BATCHRC, that are found in the ZMGRSYNC object

Overview

DMABATCH is called by the executable file, **DMABATCH.EXE**. It requires no arguments and completes the database synchronization using the information that was specified in the **Choose Configuration Servers and domains** panel of the Radia DCS configuration-only phase.

Note
The configuration-only information was saved in the ZMGRSYNC object on the Destination.

DMABATCH Considerations

Before Radia DMABATCH can be run, it must be configured in the **Choose Configuration Servers and domains** panel, as discussed starting on page 63.

■ The pair of Radia Configuration Servers (and selected domains) that was defined in the **Choose Configuration Servers and domains** panel will be synchronized.

■ The associated Radia DCS log file and objects will be created in IDMLOG and IDMLIB, respectively, or the default directory for Radia DCS. The log file is **DMABATCH.LOG**.

DMABATCH Scripting Commands

The functions that are described in this section are DMABATCH command-line arguments, intended for use in a script that executes DMABATCH and handles error conditions. These functions are called with the command,

```
DMABATCH ACTION=
```

Notes
If no value is specified for ACTION, or if ACTION=SYNC, a normal synchronization is done.
Any other ACTION value does the indicated action only, with *no synchronization*.

DMABATCH Line Commands

This section details the use and functionality of the DMABATCH commands.

DMABATCH ACTION=QUIESCE [MANAGER=*ID*][QTYPE=TASK|TRANS]

This command quiesces the Destination, thereby increasing the chance of later obtaining a hard-lock.

■ If MANAGER= is omitted, the default is the Destination's ID.

■ QTYPE=TASK: prevents any new, non-DCS client tasks from starting.

  QTYPE=TRANS: same as TASK, but also, for currently running non-DCS client tasks, the client connection terminates when the client sends the next transaction.

■ This action would likely be scripted to run before a synchronization, thereby decreasing the likelihood of later having to kill any tasks.

DMABATCH ACTION=RESUME [MANAGER=*ID*]

This command ends a "quiescent" state on the Destination.

■ If MANAGER= is omitted, the default is the Destination's ID.

DMABATCH ACTION=KILLTASKS [MANAGER=*ID*]

This command should be used if QUIESCE was not sufficient to clear out other tasks in time for Radia DCS to enter the Commit phase. It terminates all non-DCS client tasks on the Destination, allowing a Radia DCS run to obtain a hard-lock and commit the changes.

■ If MANAGER= is omitted, the default is the Destination's ID.

■ Use KILLTASKS in a script after a synchronization terminates with BATCHRC=003 (hard-lock timeout).

DMABATCH ACTION=RESET

This command will reset an incomplete synchronization session to its initial state, release any locks held by the session, and release staged resources (if any). Any subsequent synchronization of the defined synchronization pair will start from scratch.

■ Use RESET in a script after a synchronization fails (BATCHRC not = 000), if it is determined that the synchronization cannot be resumed in a timely manner and the Configuration Servers can't be left locked.

DMABATCH ACTION=SOFTLOCK [MANAGER=*ID*]
DMABATCH ACTION=UNLOCK [MANAGER=*ID*]

This pair of commands is intended for use with multiple synchronizations from a shared Source database.

■ Soft-locking the Source guarantees that resource data that is generated for Source domains will be retained in cache instead of being recalculated for each synchronization, thereby increasing performance.

■ If MANAGER= is omitted, the default is the Destination's ID.

DMABATCH Automation

Radia DCS can be set up to automatically handle a variety of situations. The following are examples of error and non-error situations that might arise during Radia DCS's batch processing; and for which, automated solutions can save time and administrator intervention.

Error Situations

■ The synchronization is interrupted due to a physical (for example, line dropped) or logical problem. Should the databases remain locked?

■ The synchronization is resumed after an interruption, but the Configuration Server has been recycled in the interim, with its database possibly exposed to updates. Should the synchronization resume or start over?
See *Reset Session on Staging Failure* on page 89.

■ At commit time, the Destination database cannot be hard-locked because non-DCS client tasks are running. Should the DCS wait for the tasks to end, terminate them, or fail?
See *Batch Lock Timeout Action* on page 89.

Non-Error Situations

■ When it's time to 'commit' the changes, the Destination must be hard-locked in order to deny access to data that is partially updated. However, this cannot occur while client tasks are running on it. The Destination waits ([MGR_DMA].DMA_TIMEOUT) for these tasks to end before aborting them, generating BATCHRC=003.
See *Batch Lock Timeout Action* on page 89.

■ If a partially completed synchronization terminates, and cannot be brought to completion quickly, its state will need to be reset, freeing the Configuration Server locks and other resources.
See *Reset Session on Staging Failure* on page 89.

■ If one of the Configuration Servers being synchronized becomes unlocked before the operation is completed, should the synchronization be allowed to resume, or be forced to start over?
See *Batch Lock Timeout Action* on page 89.

Automated Solutions

This section details the set of Radia DCS options that can be specified in order to control DMABATCH's actions when an error occurs. These DCS options can be stand-alone executions of DMABATCH with special command-line arguments.

Note
These options are also available via the **Radia DCS Options** tabs.

The following options can be specified to affect how a single execution of Radia DMABATCH will respond to certain error conditions. These options can be set to control DMABATCH operation without the use of a script.

Reset Session on Staging Failure

Normally, if synchronization fails during the staging phase (for example, due to a lost connection), it is left in a state that ensures that it can subsequently be restarted from the point of failure. This entails leaving both Configuration Servers locked. If leaving both Configuration Servers locked, pending a restart, is not acceptable, this option allows the session to be reset to the initial state. Both Configuration Servers will be unlocked, and staged resources will be freed immediately. The trade-off is that "re-startability" is sacrificed, which can be a problem if staging failed near the end of a long process.

To manually reset a failed session

■   Specify:

    DMABATCH ACTION=RESET

> **Note**
>
> This is equivalent to selecting **Reset session on failure** on the **Batch Operation** tab (Figure 5.4).

Batch Lock Timeout Action

This option controls which of three possible actions Radia DCS will take if active non-DCS tasks are running, and preventing the Destination from achieving a hard-lock at the Commit step.

Table 6.1 ~ Batch Lock Timeout Actions	
Action	**Result**
Fail	This is the default. It cancels the current attempt, and reports a BATCHRC=003.
Retry Indefinitely	Continue retrying until a lock is obtained.
Kill Other Tasks	Terminate all non-DCS tasks that are running on the Configuration Server. DCS obtains the hard-lock and continues commit processing.

To manually set a Batch Lock Timeout option

■   Specify:

    DMABATCH ACTION=Fail|Retry|Force

> **Note**
>
> This is equivalent to selecting **Fail**, **Retry indefinitely**, or **Kill other tasks** in the **Batch Lock Timeout Action** section on the **Batch Security** tab (Figure 5.5).

DMABATCH Command-Line Options

Deferred Commit

If COMMIT=NO is specified,

    DMABATCH COMMIT=NO

all DCS processing is halted after the Staging step.

■   If Staging is successful, BATCHRC=**000** and the following message will be returned,

    ZMGRSYNC.BATCHMSG="Commit bypassed by COMMIT=NO".

At this point, the Source is *unlocked*, and the Destination is *soft-locked*. The commit must be done subsequently without COMMIT=NO.

IP Address Support for Cloned Managers

Cloned Managers share a database and a MGR_ID. They are distinguishable only by IP name/address; and only one of the Configuration Servers "owns" the database. Only the owning Configuration Server can update and synchronize the database. To distinguish between cloned Managers for non-synchronization actions, on the command line, specify:

    RCSADDR=ipaddress

(The MGR_ID still needs to be specified.)

The ZMANAGER entry that matches the ID will supply all the other communications parameters. For example:

    DMABATCH ACTION=SOFTLOCK MGRID=123 RCSADDR=*local_host*

Hard-lock Operation

By specifying,

    DMABATCH ACTION=HARDLOCK

the specified Configuration Server is *hard-locked*, thereby preventing clients from connecting. Also, any client tasks that are running on the Configuration Server are killed. The specified Configuration Server must currently be *unlocked*. The default is the Destination Configuration Server.

This operation is intended for use with Cloned Managers (see *IP Address Support for Cloned Managers*, above) that must be prevented from receiving connects while the database-owning Configuration Server is being synchronized.

Results of DMABATCH

The results of a DMABATCH synchronization are found in the batch message variable, BATCHMSG, and the associated batch return code variable, BATCHRC, in the ZMGRSYNC object. At the Differencing step, a BATCHRC of **000** indicates that no domain differences were found. The associated BATCHMSG message is **No domain differences found**.

During the Staging and Commit phases of DCS, the **000** return code indicates that the phase was successful. The corresponding BATCHMSG is **Synch OK**.

Configuration Server Response to DCS Request

Table 6.2 lists the Configuration Server BATCHRC and corresponding BATCHMSG responses to DCS requests.

Table 6.2 ~ BATCHRC and BATCHMSG Responses	
BATCHRC	**BATCHMSG**
000	"No domain differences found." (At the level indicated.)
000	"Synch OK." Differences were found and the synchronization was successful.
001	"General warning."
002	"Database is locked. Some client tasks were forced to end."
003	"DCS timeout expired."
005	"Timestamp mismatch. Unable to execute the transaction."
009	"Lock timestamp must be specified. DMALTS is wrong or missing."
010	"_Handle must be specified. It is wrong or missing."
011	"DMA_Handle verification failure."
012	"Invalid request object missing var(s)."
013	"userID/Password verification error. Transaction terminated."
014	"Processing started."
015	"Unknown function."
016	"DMA general error."
017	"DMA mgr softlocked (shared)."
018	"DMA mgr hardlocked."
019	"DMA mgr not locked."

Table 6.2 ~ BATCHRC and BATCHMSG Responses

BATCHRC	BATCHMSG
020	"DMA wrong mgr id."
021	"Configuration Server is Standalone."
022	"Stage file write err (out of space?)."
023	"Stage file read err."
024	"Stage file iocontrol err."
025	"Cntrl file write err (out of space?)."
026	"Cntrl file read err."
027	"Cntrl file iocontrol err."
028	"Database CreateInst failed (out of space?)."
029	"Database LockInst failed."
030	"Database WriteInst failed (out of space?)."
031	"Database DeleteObj failed."
032	"Database locked by more than one task."
033	"Configuration Server is softlocked (exclusive)."
034	"Userid not defined to Configuration Server."
035	"Stage directory not found."
036	"Stagefile delete failed."
037	"Stage directory delete failed."
038	"Inconsistent lock state."

A

Troubleshooting Radia DCS

AT THE END OF THIS CHAPTER, YOU WILL KNOW SOME OF THE WAYS TO TROUBLESHOOT RADIA DCS, INCLUDING:

■   How to activate tracing for DCS operations

■   Which EDMAMS verbs are associated with Radia DCS and how they are used

■   How to prevent and correct domain eligibility issues

Logs to Obtain

For DCS problems, the DMABATCH.LOG and the Destination Configuration Server log are needed. Although, it is unlikely that the Source Configuration Server log will be helpful, do not discard it or allow it to be overwritten before contacting HP Technical Support.

Log Error Messages

The five DCS-related log messages shown below correspond to the domain eligibility rules. Depending on the circumstances of the failed synchronization, these messages might be found in the DCS.LOG.

■ The "last synch" and "last update" dates refer to the local time on the Destination Configuration Server.

■ The bracketed numbers [999] and [222] indicate the MGR_ID.

   a. GetDomains
   NVD002223 99069 16:48:06
   Domain [SYSTEMX] not eligible: self-owned at Destination [999]

   b. GetDomains
   NVD002223 99069 16:48:06
   Domain [NEWDOMAN] not eligible: replication to owner [999]

   c. GetDomains
   NVD002223 99069 16:49:18
   Domain [SYSTEMX3] not eligible: Source owner [222] and Destination owner [999] unequal

   d. GetDomains
   NVD002223 99069 16:50:18
   Domain [SYSTEMX2] not eligible: peer mode and Source was updated [97/12/17 19:42:26] after its last synch [97/12/01 12:00:00]

   e. GetDomains
   NVD002223 99069 16:52:18
   Domain [SYSTEMX2] not eligible: peer regression: last dest update [99/12/18 19:42:26] more recent than last Source update [99/12/13 19:42:26]

Log and Object Locations

Like the Radia Clients, the default locations for the Radia DCS objects and logs are **IDMLIB** and **IDMLOG**, respectively.

Configuration Server Tracing for DCS

In the MGR_TRACE section of the edmprof file, add the following setting and value:

```
DMA=YES
```

Radia DCS Objects

Obtain the following objects from the default DCS directory. If Radia DCS is running on a desktop with either a Radia Client or Administrator, look in IDMLIB.

- **ZMANAGER**
  contains the properties of all the Radia Configuration Servers that have been defined to Radia DCS. TP parameters (including TP trace level) are defined here, per Configuration Server.

- **ZMGRSYNC**
  contains information about the synchronization pair, including any applicable password information. This object is refreshed when: 1) another synchronization pair is defined and 2) when there is a domain change. The object retains information of saved Radia DCS sessions for subsequent recall.

  The non-TP trace level is determined by ZMGRSYNC.ZTRACEL.

  There are two ZMGRSYNC variables that relate to DMABATCH, BATCHMSG and BATCHRC. These are described in Table 6.2 on page 91.

The EDMAMS Utilities

EDM Access Method Services (EDMAMS) is a set of utilities that can be used to create, delete, copy, change, and list Radia Database objects. The *verbs*, *keywords*, and *values* can be specified in uppercase, lowercase, and mixed-case.

> **Note**
>
> For more information on the EDMAMS verbs, see *Chapter 6: EDM Access Method Services* in the *Radia Configuration Server Guide*.

Three of the EDMAMS verbs will display or update Radia DCS control data:

- **REFRESH_DMA**
  Rebuilds instance counts, class counts, and update dates in the database control data. If DOMAIN is omitted, all domains are refreshed.
  ```
  ZEDMAMS
  VERB=REFRESH_DMA(,PREVIEW=YES/NO)(,DOMAIN=domain_name)(,CLASS=class_name)
  ```

- **UPDATE_MGRIDS**
  Updates the MGR_ID, Configuration Server name, owning MGR_ID, and owning

Configuration Server name in the database control data. If DOMAIN is omitted, all domains are updated. All keywords are optional; however, at least one keyword other than DOMAIN must be specified.

```
ZEDMAMS
VERB=UPDATE_MGRIDS(,FILE=file_name)(,DOMAIN=domain_name)(,CLASS=class_name)
(,MNAME=local_mgr_name)(,MID=local_mgr_id)(,MMNAME=owning_mgr_name)(,MMID=ow
ning_mgr_id)
```

■ **LIST_PREFIX**
Displays formatted database control data. If DOMAIN is omitted, all domains are displayed. If CLASS is omitted, all classes are displayed.

```
ZEDMAMS
VERB=LIST_PREFIX(,FILE=file_name)(,DOMAIN=domain_name)(,CLASS=class_name)
```

Output is written to ZEDMAMS.LOG, with the exception of error conditions, which are written to STDERR.

Domain Eligibility

■ *What if Radia DCS presents no domains as eligible for synchronization?*

If no domains are presented as eligible for synchronization, make sure that both Configuration Servers were installed properly, as DCS-enabled Configuration Servers, because domain ownership is assigned during installation.

> **Note**
>
> Refer to the *Radia Configuration Server Guide, Chapter 1: Radia Configuration Server Settings* for more information about the MANAGER_TYPE setting of the MGR_STARTUP section of the edmprof file.

■ *Which Configuration Server owns the default domains (SYSTEMX, ZACCESS, and ZSYSTEM)?*

If Configuration Server 001 owns them, make sure that they were not assigned ownership to Configuration Server 002 during its installation.

> **Note**
>
> For more information on the EDMAMS verbs, see *Chapter 6: EDM Access Method Services* in the *Radia Configuration Server Guide*.

Lists

Figures

Tables

Procedures

Index