

Project Guide for Oracle RAC Implementation
by Christopher Haskins

A guide for defining, designing, and delivering a successful Oracle RAC project.

Published April 2006

Oracle Real Application Clusters (RAC) is the premier database clustering solution in the RDBMS market space. Oracle RAC's configuration options and features offer companies a wide range of flexibility for designing their high-availability solutions. However, with all the configuration options, features, and flexibility, how do you guarantee a successful implementation?

This article is a guide for defining, designing, and delivering a successful Oracle RAC project. It details the steps required to reduce risks and increase your chances of a successful implementation. In addition, it highlights many major pitfalls you may encounter during your Oracle RAC project and offers suggestions on how to avoid them.

Although this article focuses on Oracle RAC, the following steps are applicable to many types of Oracle implementation projects. (Note that this guide is intended for informational purposes only; under no circumstances should you consider it a consulting offering.)

So let's get started!

Requirements Definition

The first major phase in delivering a successful Oracle RAC implementation is defining the actual goals of the project. The Requirements Definition step involves identifying and documenting the features and functionality delivered during the implementation phase of the project.

As you proceed with your Oracle RAC implementation, you will continually come back to your requirements lists. Having a set of documented requirements will lend direction to your Oracle RAC project. Without it, you will find the project difficult to manage, as new, unexpected changes creep into the project's implementation.

Avoiding Pitfall #1: Make sure your key business and technical personnel actively participate in identifying the requirements for the project. Clearly communicate all requirements to the project stakeholders, including key management staff, technical staff, and end users.

Step 1 – Defining Project Scope

The first step in the Requirements Definition phase is defining the project scope. The project scope, a collection of details that justify the business need for the project, describes the project's high-level deliverables. The project scope is sometimes referred to as "business requirements."

To determine the project scope, ask yourself the following questions:

- What are the business objectives of the project?
- What is the project trying to accomplish?
- What are the key benefits of successful project completion?

A sample project scope document detailing the high-level goals of a sample Oracle RAC project is shown below.

Justification	We are implementing Oracle RAC to make our applications scalable and highly available and to offer more-reliable services to our customers.
Goals / Deliverables	The final product of this project will be a new Oracle RAC system that supports the level of service detailed in our service-level-requirements document*. *attached below
Project Schedule Constraints	The project must be completed by August 2006.
Project Cost Constraints	The project cost should not exceed \$XXX,XXX.

Avoiding Pitfall #2: Strive to make the project objectives quantifiable. You will be able to come back to these objectives and measure the overall project’s success. Making objectives quantifiable includes documenting project schedule and cost constraints.

Step 2 – Defining the Project Team

Defining the project team involves identifying the individuals who will contribute to the project’s deliverables and the ones who will complete the tasks within the project plan. These individuals may include persons from multiple areas of your organization, including executive staff, business analyst staff, and technical staff.

The following matrix describes the makeup of a typical Oracle RAC project, describes their function, and specifies the steps at which they may contribute to the project.

Role	Responsibility	Participation Phases	Oracle RAC–Specific Tasks
Executive	<ul style="list-style-type: none"> • Sponsors project • Provides funding 	<ul style="list-style-type: none"> • Scope definition • Service-level requirements definition 	
IT manager	<ul style="list-style-type: none"> • Provides IT resources • Provides staff resources • Reports progress 	<ul style="list-style-type: none"> • Scope definition • Team definition • Service-level 	

	to executives	requirements definition	
Project manager	<ul style="list-style-type: none"> • Coordinates project • Manages project • Assigns tasks to project staff • Reports progress to managers 	<ul style="list-style-type: none"> • All 	
Database administrator	<ul style="list-style-type: none"> • Installs and upgrades database software • Creates, updates, administers, and monitors databases • Optimizes database performance • Backs up and restores databases • Creates physical and logical database designs 	<ul style="list-style-type: none"> • Service-level requirements definition • Schedule definition • Technical architecture design and build • Testing 	<ul style="list-style-type: none"> • Installs Oracle software • Configures Oracle Clusterware • Plans and configures shared storage • Configures Automatic Storage Management (ASM) • Creates databases and instances • Creates and configures services • Configures workload management • Monitors and tunes performance • Configures and tests backups • Performs backups and restores
Network administrator	<ul style="list-style-type: none"> • Configures network components • Administers the 	<ul style="list-style-type: none"> • System requirements definition • Schedule 	<ul style="list-style-type: none"> • Assigns server IP addresses • Configures networking

	network	definition <ul style="list-style-type: none"> • Technical architecture design and build • Testing 	components <ul style="list-style-type: none"> • Configures private interconnects • Configures virtual IPs
System administrator	<ul style="list-style-type: none"> • Administers application and database server hardware and software • Monitors system performance • Advises on system design and the use of system resources • Provides administrative support • Configures hardware and software components 	<ul style="list-style-type: none"> • System requirements definition • Schedule definition • Technical architecture design and build • Testing 	<ul style="list-style-type: none"> • Configures server hardware • Installs and configures operating system software • Configures networking components • Plans and configures shared storage • Installs Oracle software • Plans and maintains backups
Application developer	<ul style="list-style-type: none"> • Designs, develops, and maintains database applications • Designs, develops, and maintains software components and scripts 	<ul style="list-style-type: none"> • System requirements definition • Schedule definition • Technical architecture design and build • Testing 	<ul style="list-style-type: none"> • Does application configuration • Creates Oracle Clusterware application profiles • Provides unit/integration test support
Tester	<ul style="list-style-type: none"> • Designs test plans • Performs testing • Verifies that requirements are met 	<ul style="list-style-type: none"> • Schedule definition • Testing 	<ul style="list-style-type: none"> • Does unit testing • Does user acceptance testing • Does integration testing

			<ul style="list-style-type: none"> • Does stress testing
Application user	<ul style="list-style-type: none"> • Uses database applications • Performs testing • Verifies that requirements are met 	<ul style="list-style-type: none"> • System requirements definition • Testing 	<ul style="list-style-type: none"> • Does user acceptance testing

The responsibilities of your Oracle RAC project team members may differ from site to site, depending on the size of the site and the system requirements.

While you're putting this project team together, the most qualified staffers for the project may be unavailable. This constraint may force you to go with the people who *are* available. In such a situation, you can lessen implementation risks by sending your project team members to get the appropriate technical training. Technical training often leads to reduced project risk and higher-quality project deliverables.

Avoiding Pitfall #3: If the new Oracle RAC system is replacing an existing legacy system, include individuals who are sufficiently experienced in the old system. Having these team members will help ensure that all project requirements have been met.

Step 3 – Defining Service-Level Requirements

The third step in the requirements definition phase is defining the service-level requirements. Service-level requirements are the levels of service your Oracle RAC project implementation is expected to support. They document the service-level expectations and operational requirements and provide guidelines for handling delays and failures.

Service-level requirements can be broken up into two categories of requirements: service-level requirements and operational requirements.

Service-level requirements assist in aligning the Oracle RAC technology implementation with the project's scope—the project's business goals. You begin to identify your service-level requirements by first analyzing requirements of existing systems. This analysis includes reviewing existing system operational, technical, and support procedures and documentation.

You can further identify service-level requirements by asking questions such as

- What are the critical business hours during which the Oracle RAC system is expected to be online?
- What are the various levels of service required of the system?
- What are the minimum acceptable levels of performance and availability?
- What are the procedures for handling delays and failures?

The answers to these questions are typically grouped into a rated, tiered service-level-requirements matrix that defines the differing levels of service.

Below is a sample service-level-requirements matrix. Exact definitions and the number of tiers will depend on your individual organizations and business units.

Tier	Severity-Level Description	Performance	Availability	Resolution Requirement
5	Normal operation	System is responding at normal operating baseline.	System is 100% available. All outages are properly scheduled.	None
4	Severity level 4: Trivial problem with little or no impact	Performance is 10% to 30% below the required baseline.	90% to 95% of the applications or application functionality is available.	Must be resolved within five days
3	Severity level 3: Minor problem with minimal impact	Performance is 30% to 50% below the required baseline.	85% to 90% of the applications or application functionality is available.	Must be resolved within three days
2	Severity level 2: Noticeable problem with measurable impact	Performance is 50% to 70% below the required baseline.	80% to 85% of the applications or application functionality is available.	Must be resolved within one day
1	Severity level 1: Severe problem with high business impact	Performance is 70% or more below the required baseline.	75% or less of the applications or application functionality is available.	Must be resolved within three hours

Operational requirements define the procedures required to maintain the Oracle RAC system and the service-level requirements defined above. Often, operational requirements include information on scheduled maintenance outages, system startup and shutdown, system backups, Oracle RAC system availability, failover procedures, and disaster recovery plans.

You identify operational requirements by asking questions such as

- How do we maintain Oracle RAC system performance baselines?
- How long should maintenance operations take?
- Which maintenance and backup operations should be performed “online”?
- What are the required procedures for shutting down and starting up the system?
- What type of backups should be performed to maximize system recoverability?
- How do we prepare for disasters?

A sample Oracle RAC operational requirements list is shown below.

Scheduled Maintenance Outages	The last weekend of every month will be reserved for Oracle RAC system maintenance operations. The outage will not last more than 56 hours, starting Friday evening. These outages will be reserved for maintenance operations that cannot be performed “online.”
System Backups	Full backups will be run online on the weekends, with incremental backups performed in the evenings during the week. Four weeks’ worth of backups will be maintained on tape, with one day’s worth of backups maintained on disk.
Failover Procedures	All application sessions should fail over to available Oracle RAC nodes in the event of a single-node failure. In the event of a localized disaster in which all Oracle RAC nodes are unavailable, the local standby environment should come online within three hours.
Disaster Recovery Procedures	In the event of a sitewide disaster, the off-site standby environment will be brought online within three hours.
System Capacity	The system should support our current user load, with a projected two-year user increase, and support the current set of applications. In the event that the system is not keeping up with the user load, additional Oracle RAC nodes will be added. Processor, memory, and storage requirements will be based on data gathered on current application performance on the existing hardware.

Avoiding Pitfall #4: Obtain approval and official “sign-off” for service-level and operational requirements from system end users, customers, and operational staff. This may include

negotiating the terms of performance, availability, and the appropriate responses to system failures.

Step 4 – Defining the Project Schedule

The last step in the requirements definition phase is defining the project schedule. Schedule development is one of the most critical factors of project success, because you need to make sure you have enough time to build your Oracle RAC solution while meeting all of the requirements defined above.

Schedule development involves detailing all of the tasks involved in building the system, assigning a duration to each task, and sequencing the tasks in the optimal order.

Avoiding Pitfall #5: Strive for clear communication of any time constraints (documented in Step 1) to the entire project team when developing the project schedule. Seek input from all team members to estimate and plan the project schedule accurately.

Occasionally, multiple tasks in your project schedule can be performed at the same time. Strategically parallelizing your work efforts often leads to on-time delivery and reduced project costs.

A sample high-level Oracle RAC project schedule is shown below. It demonstrates common tasks performed during an Oracle RAC deployment.

	Task Name	Duration	Start	Finish	Prerequisite Task
1	Server hardware configuration	2 days	Thu 12/1/2005	Fri 12/2/2005	
2	Shared storage configuration	1 day	Thu 12/1/2005	Thu 12/1/2005	
3	OS install	1 day	Mon 12/5/2005	Mon 12/5/2005	1
4	Network configuration	1 day	Tue 12/6/2005	Tue 12/6/2005	3
5	Oracle Database Software install	1 day	Wed 12/7/2005	Wed 12/7/2005	4
6	Database build	2 days	Thu 12/8/2005	Fri 12/9/2005	5
7	Data load	5 days	Mon 12/12/2005	Fri 12/16/2005	6
8	Unit testing	2 days	Mon 12/19/2005	Tue 12/20/2005	7

9	Stress/integration testing	5 days	Wed 12/21/2005	Thu 12/29/2005	8
10	Failover testing	2 days	Fri 12/30/2005	Tue 1/3/2005	9
11	Backup-and-recovery testing	19 days	Wed 12/12/2006	Wed 1/4/2005	5
12	System integration	5 days	Thu 1/5/2006	Wed 1/11/2006	11
...	...				

An appropriately detailed project schedule enables the Oracle RAC team to track the project schedule's progress while assisting it in proactively responding to schedule delays. When a schedule change is required, make sure you thoroughly document the changes. The original project schedule, along with the report of changes, creates a powerful tool for estimating project schedules for future projects.

Avoiding Pitfall #6: Take advantage of multiple tasks that can be performed at the same time. In the project schedule above, note how Task #11 can run simultaneously with Task #7 through Task #10.

After defining and documenting the project scope, project team, service-level requirements, and project schedule, implement a strong change-control system. Carefully manage any changes to the requirements, to keep the costs within budget and the project on schedule.

Technical Architecture Design and Build

The second major phase in delivering a successful RAC implementation is determining and implementing the technical architecture specifications of your Oracle RAC deployment. The technical architecture details the hardware, software, and configuration that will constitute the new system. Because most Oracle RAC implementations focus on moving from single-instance environments to Oracle RAC instance environments without a complete redesign of their applications and databases, you will both design and build the Oracle RAC environment as you proceed through this phase.

The following steps explain how to transform the requirements into a working design.

Step 1 – Determine the Hardware and Software Specifications

This step involves taking the service-level requirements and operational requirements, defined above, and translating them into hardware and software specifications. It also considers hardware compatibility, specific operating system requirements, and Oracle RAC-specific software requirements.

Use the Hardware/Software Considerations Chart below as a checklist and for documenting the decisions made in this step. For your individual implementation, fill in the actual hardware-component and software-component items you are using for your project.

When filling in this chart, ask questions such as

- Does the component assist in meeting service-level requirements?
- Is the component and the quantity of this component sufficient to meet the operational requirements?
- Is the component compatible and/or certified to work with the other hardware components?
- Is the component compatible and/or certified to work with the operating system?
- Does the component meet the Oracle RAC software requirements?
- Is Oracle RAC certified and supported to run on the component?

Avoiding Pitfall #7: Ensure that the Oracle RAC project team knows the capabilities and features for each of the components constituting the Oracle RAC system and that all of the components are certified to work together. You can reduce your Oracle RAC project risks via the appropriate technical training and proof-of-concept testing.

Components	Meets Project Requirements ?	Meets OS Requirements ?	Meets Oracle RAC Requirements ?	Compatible with Other Hardware/Software Components?
Hardware Component				
Server (# of nodes)				
Processor (# CPUs per node)				
Memory (GB per node)				
HBA(s)				
Network cards (# cards per node)				
Local disk (GB per node)				
SAN/shared storage (# of GB)				
Software Component				

Operating system				
Hardware drivers				
Volume management/ multipathing software *includes ASM, RAW, or OCFS volume management decisions				
Oracle Clusterware/Oracle Database software				
Oracle client software				

Avoiding Pitfall #8: If you're moving to an entirely new hardware and/or software platform, make sure to test your applications. Changing platforms may require adding additional processors or memory to meet service-level requirements.

Step 2 – Implementing the Specifications

After completing the checklist above, it's time to construct the Oracle RAC environment.

These tasks include the following:

I. Configure the Server Hardware

- A. Install the CPUs, memory, and local disk
- B. Install and configure the HBAs, network cards, and networking components
- C. Configure the hardware interconnects
- D. Install and configure storage switch devices, and attach them to shared storage

II. Configure the Operating System

- A. Install the operating system
- B. Configure operating system kernel parameters
- C. Configure the hangcheck-timer or interconnect heartbeat module
- D. Create operating system user groups and users
- E. Create and configure shared storage devices
- F. Install and configure raw partitions or Oracle Cluster File System
- G. Configure Secure Shell (SSH)

III. Configure the Oracle Software

- A. Install Oracle Clusterware
- B. Install the Oracle server software
- C. Configure Automatic Storage Management (ASM)
- D. Create the databases
- E. Create the database instances
- F. Create services
- G. Create Oracle Clusterware application profiles

IV. Operational Tasks

- A. Perform data loads
- B. Perform index builds
- C. Set up OS and database backups
- D. Create standby/Oracle Data Guard environments
- E. Install and configure performance monitoring utilities, such as Oracle Enterprise Manager Grid Control

RAC System Testing

An Oracle RAC test strategy should consist of at least four types of testing: proof-of-concept testing, unit testing, integration testing, and load testing.

This test strategy is not a function that is to be performed separately from the above phases; instead, it is a process that is integrated into the Definition, Design, and Build phases.

This section highlights the four types of testing and identifies the project phase in which each test should be performed.

Proof-of-Concept Testing

Proof-of-concept testing is testing of the feasibility of a concept. It can mean testing a new technology, a new software architecture, or new hardware. Proof-of-concept testing allows the project team to test the validity of project decisions and gives them the ability to quickly make important decisions about the project's direction. Proof-of-concept testing is usually performed during the Service-Level-Requirements and Technical Architecture Design and Build steps.

Test	Description	Project Phases	Benefits
Proof-of-concept test	Validates or invalidates project decisions, specifically in regard to hardware and software decisions	<ul style="list-style-type: none">• Service-Level Requirements Definition• Technical Architecture Design and Build	Allows the project team to make “go/no-go” project decisions

Unit Testing

Unit testing involves the testing of a single hardware or software component or the testing of a single application or application module. This isolated test determines whether a single component or module is working within its specified requirements.

Oracle 10g Release 2 includes a verification utility called Cluster Verification Utility (CVU), which is a tool for unit-testing the hardware and software configuration of an Oracle RAC node. Use the utility for verifying the configuration of the Oracle RAC node, for checking the operating system, and for checking the network setup.

One important element of unit testing is the inclusion of “destructive testing,” in which the tester simulates abnormal activity and tries to break the system. An example of a destructive test in an Oracle RAC environment is to intentionally corrupt your Oracle Cluster Registry (OCR) and perform the steps required to get the system up and running again. A test such as this allows the team to identify vulnerable areas in the system and to prepare an action plan.

Test	Description	Project Phases	Benefits
Unit test	Tests individual hardware, software, and application components and includes “destructive testing” activity to identify weak points in the system	Technical Architecture Build tasks: <ul style="list-style-type: none">• Hardware configuration• OS configuration• Oracle Database configuration	Verifies that individual components and modules are working

Integration Testing

Integration testing involves verifying that multiple hardware, software, or application modules are working together. Integration tests determine if the system is running according to specifications.

Test	Description	Project Phases	Benefits
Integration test	Tests multiple hardware, software, and application components running together	Technical Architecture Build tasks: <ul style="list-style-type: none">• Hardware configuration• OS configuration	Verifies that integrated components and modules are working together

		<ul style="list-style-type: none"> Oracle Database configuration 	
--	--	---	--

Stress Testing

Stress testing, also referred to as load testing or system testing, is an end-to-end test that simulates a live production load. It is used to determine if the system can sustain production usage levels and if service-level requirements can be met and to gather performance data. It is also used to predict current and future usage capacity. Stress testing is often performed after all of the above tests return with positive results and after the hardware, software, and application components have been fully configured. Because it represents a major project milestone, it can be considered a separate project phase.

Test	Description	Project Phases	Benefits
Stress test	Simulates a live production load on the system	Stress testing	Verifies that the system is ready for production usage

Avoiding Pitfall #9: Testing can consume large amounts of time and money. Carefully weigh the benefits of your test plan against the resources required to perform the tests and against the risks of having system failures in production.

Operational Readiness

When should you go live with your new system?

The previous project phases and their associated steps facilitate a litmus test for assessing the readiness of the new system. Although the specifics of a completion checklist will depend on your particular site, the following generic outline will help you define, design, build, and test your Oracle RAC implementation.

Deciding your operational readiness depends on the number of completed tasks, the amount of time left in the project schedule to complete any uncompleted tasks, and the stability of the new system in its current state. It also depends on how many of the project requirements have been met.

Below is a detailed project plan with all of the implementation phases and steps this article has covered. It includes an integrated test plan and a “Critical to Launch?” column to help you determine if that particular item is absolutely required to bring the system online—or if that particular item can be brought online after the system launch.

Task	Task Description	Critical to Launch?	Completed?
------	------------------	---------------------	------------

DEFINE	Requirements Definition		
Project scope definition	Defines the high-level business goals of the project		
Project team definition	Defines the project team		
Service-level-requirements definition	Defines the service-level requirements		
Operational requirements definition	Defines the operational requirements		
Proof-of-concept test	Familiarizes the team in a preliminary way with the technologies involved, to help define the project schedule and to prepare for the Design and Build phase		
Project schedule definition	Defines the project schedule		
DESIGN AND BUILD	Technical Architecture Design and Build		
Hardware and software specification	Determines the hardware and software components to be used for the project		
Proof-of-concept test	Validates the hardware and software component choices		
Server hardware configuration	Builds the server hardware		
Operating system configuration	Installs and configures the operating system		
Server unit test	Tests the node unit, using the CVU to prevalidate the server configuration before you install Oracle Database		

Operating system unit test	Tests the OS unit, using the CVU to validate the OS configuration before you install Oracle Database Software		
Network unit test	Tests the node unit, using the CVU to validate the network configuration before you install Oracle Database		
Oracle software configuration	Installs and configures the Oracle Database software		
Integration test	Verifies that all of the hardware and software components are working, by creating a test Oracle RAC database, for example		
Operational tasks	Ready the system for stress testing and production usage		
Integration test	Verifies that applications run properly in the new Oracle RAC environment		
TEST	Oracle RAC System Testing		
Stress test	Simulates a production load on the Oracle RAC system		

Summary

During the previous three project phases, you identified the core project goals, identified the project requirements, translated the requirements into specifications, and created a test plan. Further, you created criteria for determining the Oracle RAC implementation's operational readiness. It all adds up to become your Oracle RAC implementation project plan.

This plan becomes an invaluable tool, getting you to your final destination and allowing you to foresee any problems along the way. Using such a plan *can* guarantee a successful Oracle RAC implementation—delivering on time and on budget.

Christopher Haskins (Christopher108@gmail.com) is a senior level Oracle technology consultant and project manager. He is a Project Management Professional (PMP), Oracle Certified Professional (OCP), and Red Hat Certified Engineer (RHCE).

[Send us your comments](#)