



# Synergy DCM



# What is CM Synergy?

- CM Synergy is a Configuration Management tool



Version control supports Configuration Management, and is *not* an end in itself.



1                    2                    3                    4



Although version control is very important for **developers**, delivered configurations are most important for the enterprise



# What is CM Synergy?

- CM Synergy is a development process tool

CM Synergy supports the enterprise development processes, providing control for parallel and concurrent development, remote development, cooperative or formal development.

- CM Synergy uses a relational database

Information about individual configuration item versions, (including **who**, **what**, **where** and **when**), and the way these source object versions are grouped to make a configuration, is stored in a relational database.

Individual configuration item versions are stored in a repository in part of the server file system.



# Everyday Work

Topics in this Section:

**Your tasks**

**Task lifecycle**

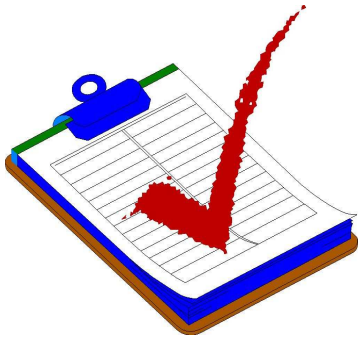
**Creating a task**



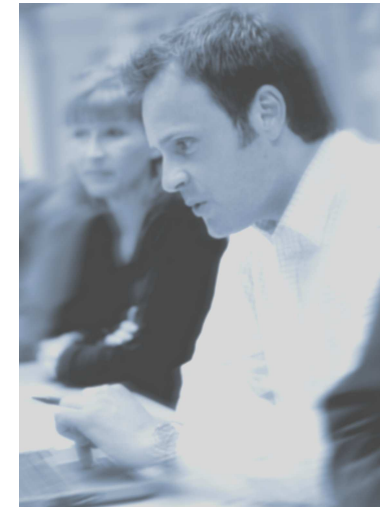
# Your Tasks

- **CM Synergy is “Task Based”**

You make changes to configuration items using a task



A task is just like an item on your “to-do” list



Other people working on this product will later use your tasks so that they can include *your* changes when they prepare this product for test, or for release to the customer

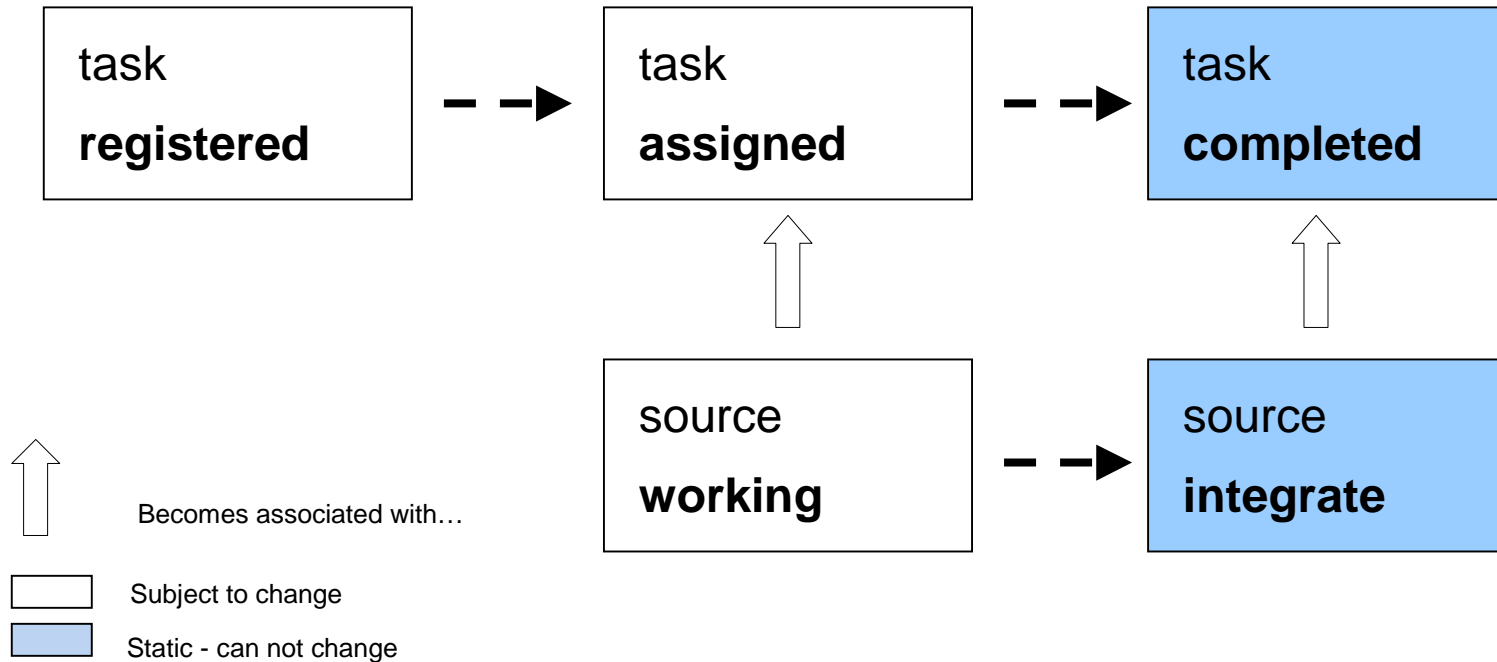


# A CM Synergy Task

- **Is a *logical unit of change***  
The changes can be backed-out without breaking dependencies
- **Is assigned to a single developer**  
Two or more people can not be asked to do the same thing  
Each person takes responsibility for what they are doing
- **Has associated with it versions of the source materials**  
Changes to source are associated with the task  
Changes to directory content are also associated with the task
- **Is completed within a reasonable amount of time**  
Tasks don't go on "for ever..."



# Task Lifecycle

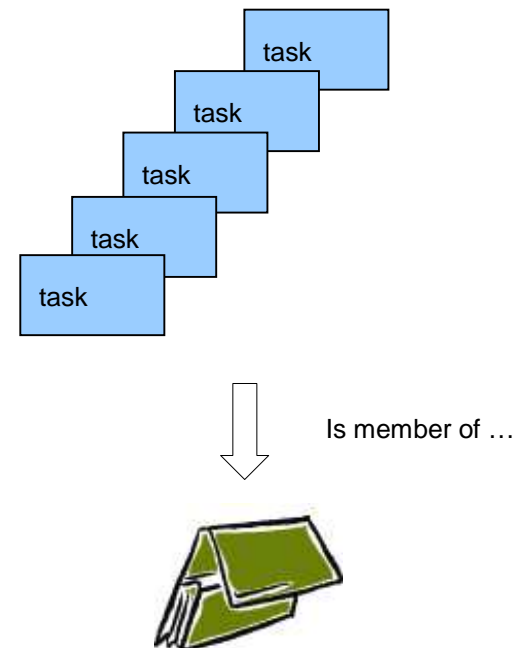


- **A “developer” is concerned about the versions of source material which belong with any individual task**



# Task Folders

- **Task Folders collect tasks together**
- **Tasks which are collected in task folders *must* have something in common**
  - Tasks share same release identifier
  - Tasks may have been selected to forward for test or for production build
  - Tasks may belong to a named developer





# Introduction to DCM



# What is Telelogic DCM?

**DCM distributes objects among databases:**

- q **Source objects including directories**
- q **Products and Projects**
- q **Tasks and Task Folders**

**DCS distributes change requests among databases**



## Why use DCM?

- q **To support multiple sites**
- q **To split up large databases**
- q **For localization or porting**
- q **For security purposes**
- q **For different levels of testing**



# Site picture

- **How is a dcm build (internal)**



# Features

- q **Transparent**
- q **Integrated**
- q **Flexible**
- q **Supports Parallel Development**
- q **Time-zone Independent**
- q **Automatable**



# Terms and Concepts

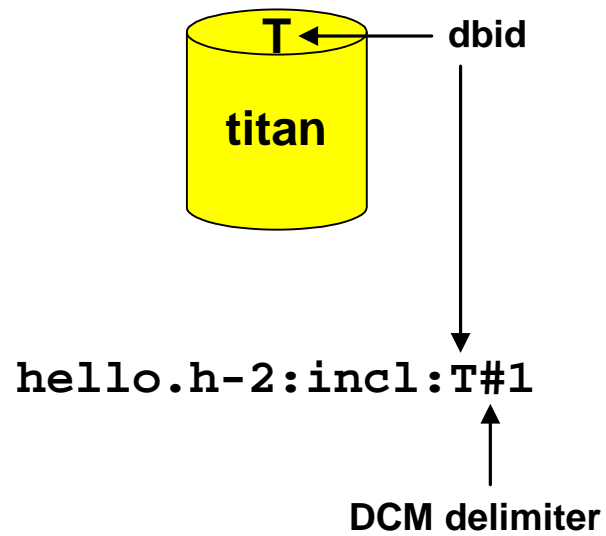


# Terms and Concepts

- q **Database ID (dbid)**
- q **DCM Delimiter (#)**
- q **DCM Cluster**
- q **Cluster Ids**
- q **DCM Methodology**
- q **DCM Properties**
- q **Destination Database Definition (DDD)**
- q **Transfer Modes, Transfer Set, Transfer Package, Transfer List**
- q **Distributing Change**

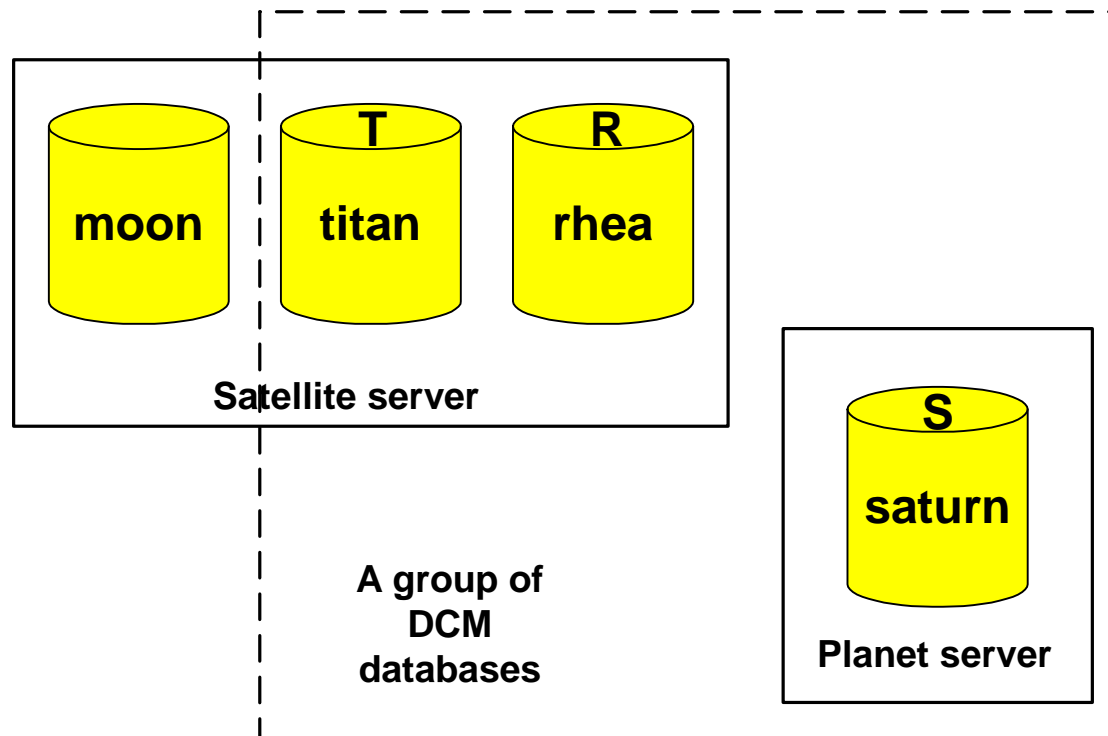


# Database ID and DCM Delimiter





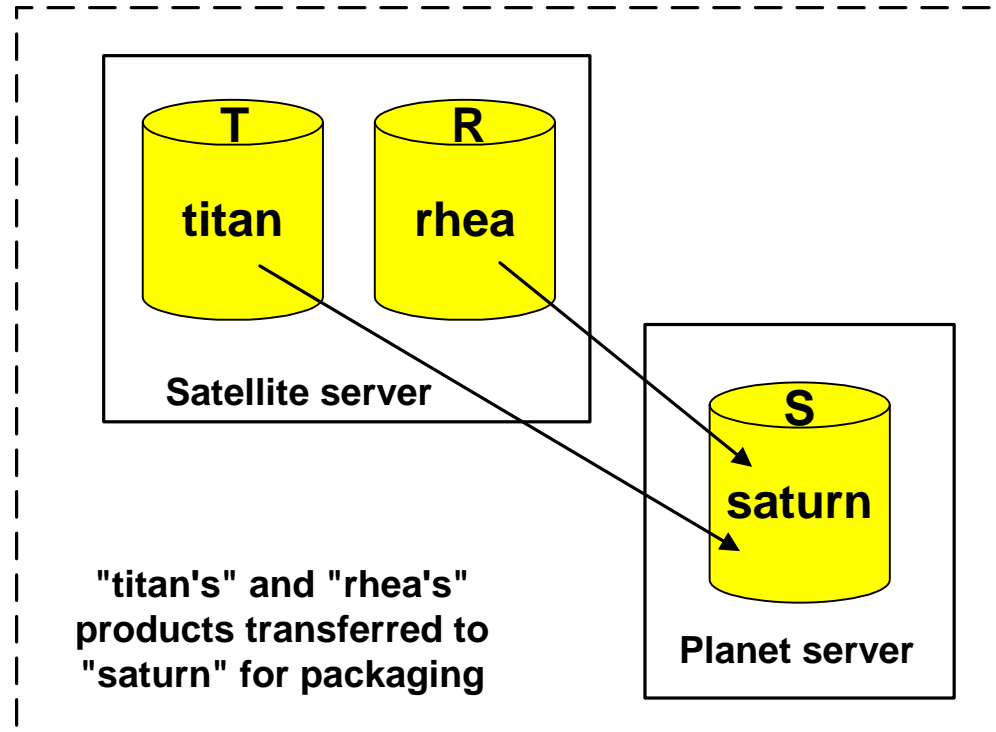
# DCM Cluster





# DCM Methodology

## EXAMPLE





# Transfer Modes Transfer Set - Big Picture

Transfer Mode	Auto Receive	Mandatory Fields
Manual Copy	No	none
	Yes	none
Local Copy	No	Path
	Yes	Host, OS, Path, CCM_HOME
Remote Copy	No	Host, OS, Path
	Yes	Host, OS, Path, CCM_HOME
File Transfer Protocol	No	Host, OS, Path
	Yes	Host, OS, Path, CCM_HOME
User Defined	No	Host, OS, Path, CCM_HOME
	Yes	Host, OS, Path, CCM_HOME

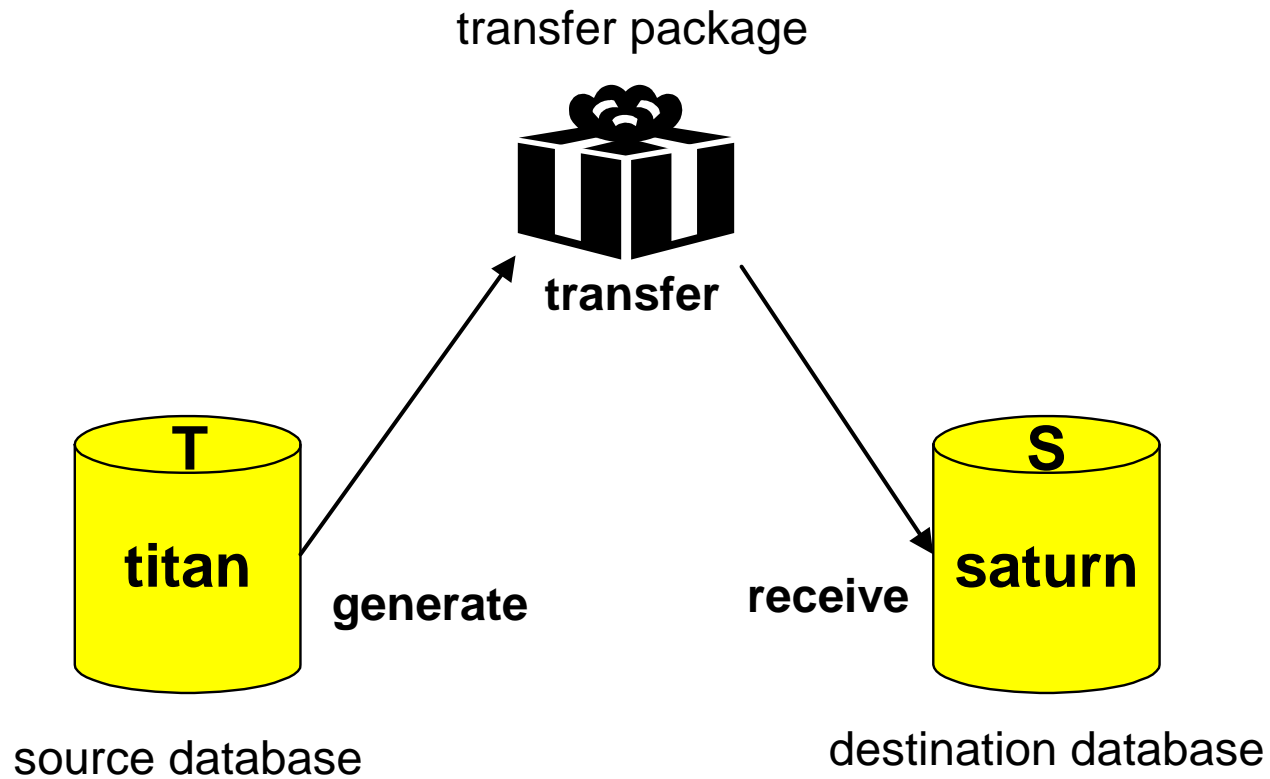


# Transfer Set Membership

Object Type	History	Objects Added to the Transfer Set
folder	N/A	The folder itself, its member tasks, and all object versions associated with the tasks.
task	N/A	The task itself and all object versions associated with the task.
project	optional	<p>The project itself and all member object versions of the project hierarchy.</p> <p>For non-static projects, such as prep projects, the following are also added:</p> <ol style="list-style-type: none"><li>1) Any project baseline that the project is using;</li><li>and 2) All folders and tasks in the project's reconfigure properties.</li></ol> <p>Note: The associated object versions of tasks are not included unless they are members of the project hierarchy.</p>
source	optional	The object version itself and the object version's associated task(s).



# Distributing Change





# Planning a DCM Cluster



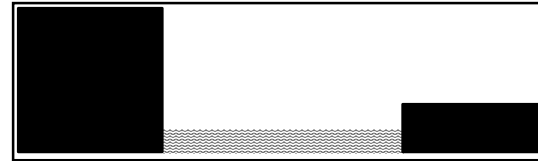
# Planning a DCM Cluster

- q **SCM Maturity**
- q **DCM Naming Conventions and Standards**
- q **DCM Methodologies**
- q **Common Database Parameters**

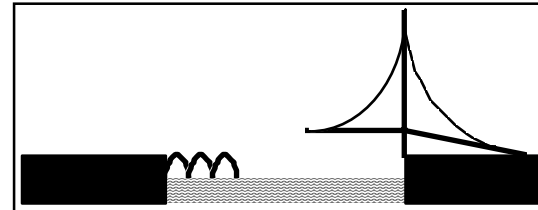


# DCM is Like a Bridge

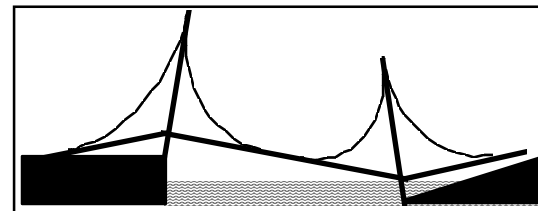
**Compatible Starting Points**



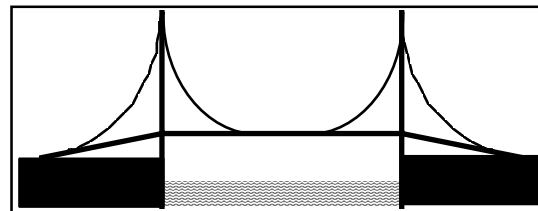
**Compatible Goals**



**Firm Foundations**

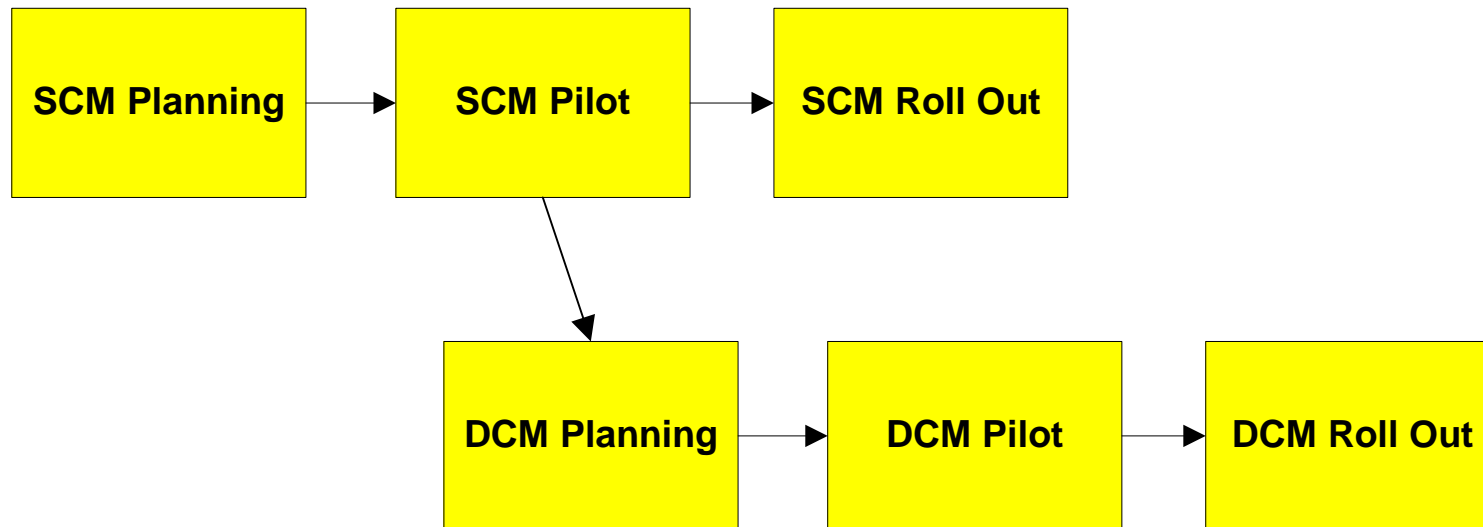


**Competent Oversight**





# SCM Maturity





# Enforced DCM Naming Conventions

- q **Instances**
- q **Object Versions**

<b>Local To</b>	<b>Initial Object Version</b>	<b>From</b>	<b>New Object Version</b>
T	hello.c-1	T	hello.c-2
T	hello.c-1	S	hello.c-S#2
T	hello.c-S#2	T	hello.c-3
T	hello.c-S#2	S	hello.c-S#3



# DCM Naming Standards

## Release Tags

- Descriptive text strings up to 32 characters
- No commas, spaces, colons, tabs, or newlines

## Task Folders

- Alphanumeric text string up to 60 characters
- Suggested convention:  
`<application>:<release>:<dbid>:<purpose>`

## Transfer Sets

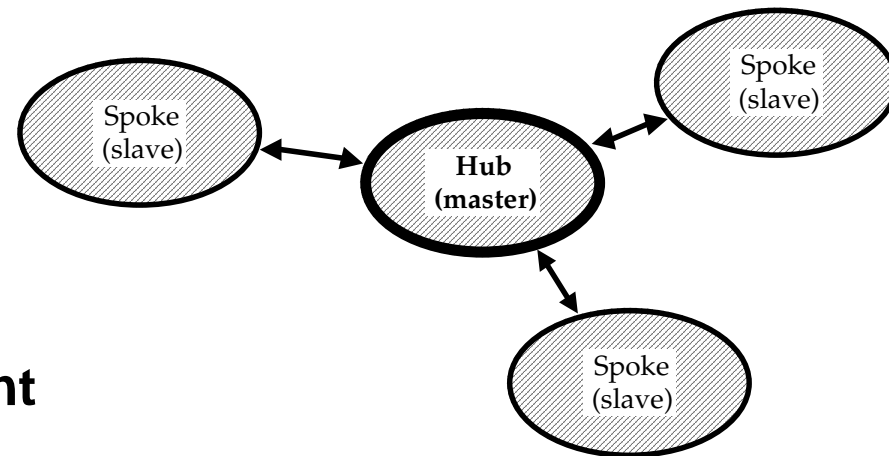
- Any alphanumeric string
- Must be unique within a given database
- Suggested conventions:  
`<dbid>_<application>_<release>_<purpose>`



# Choose Logical and Replication Methodologies

## □ Logical Methodology

- Publish and Subscribe



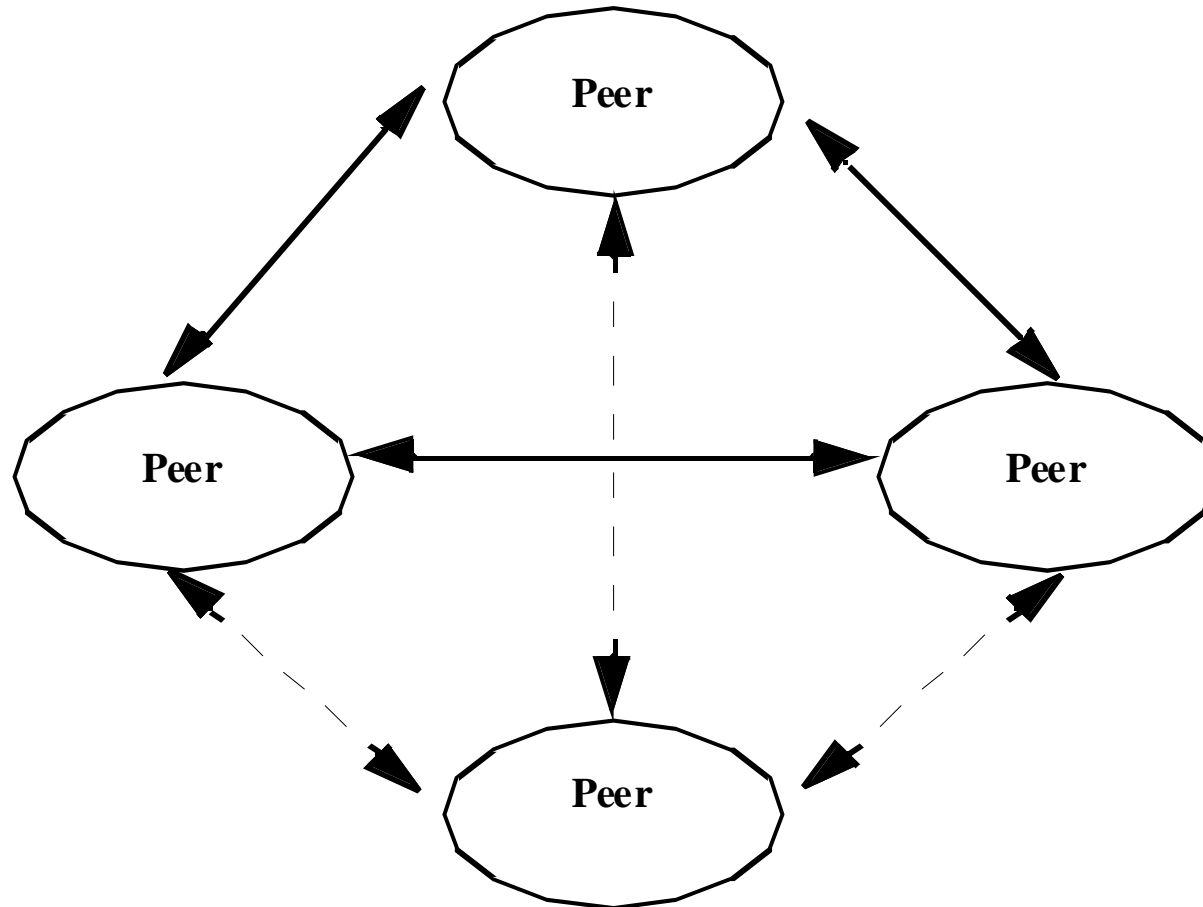
- Distributed Development

## □ Replication Methodology

- Peer-to-Peer
- Hub and Spoke

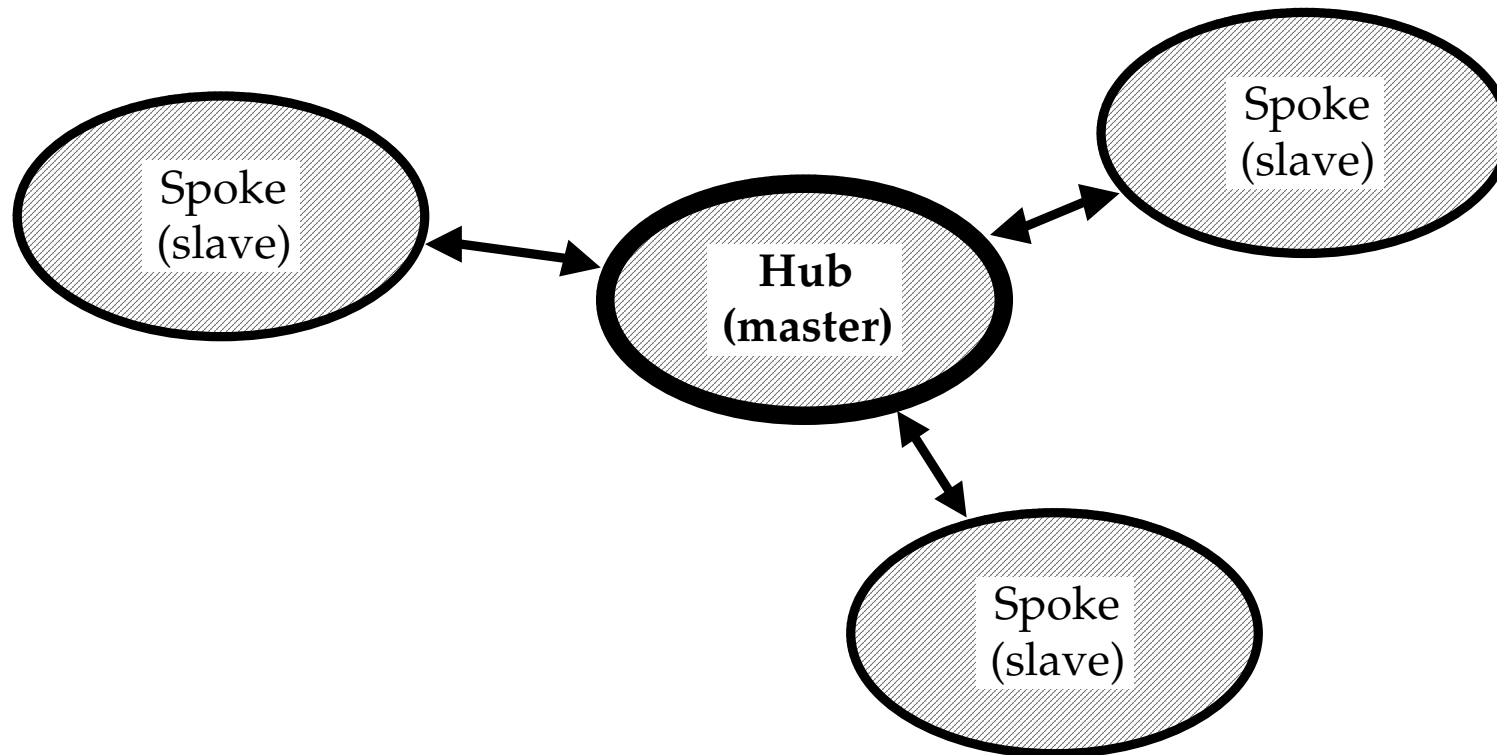


# Peer-to-Peer Methodology





# Hub and Spoke Methodology





# Side board



# Setting up Databases for DCM



# Explanation the board



# Ongoing DCM Operations



# Ongoing DCM Operations

- q **Generating Transfer Packages**
- q **Transferring Transfer Packages**
- q **Receiving Transfer Packages**



# Generating a Transfer Package

## Objects excluded via built-in rules

- Projects that are in either the working or shared state
- Projects that were created in the destination database
- All tasks not in completed state (if not licensed for DCS)
- Automatic Tasks
- Admin. objects, transfer sets, and models
- Modifiable objects, except as noted above

## Optionally excluded objects

- All products
- All imported objects (those created in other databases)
- All tasks not in completed state
- All user-defined types
- All objects of a specified type



# Summary of DCM Operations

## Setting up for DCM

- q Database Initialization
- q Destination Database Definition
- q Transfer Set Definition
- q Transfer Set Population

## Ongoing DCM

- q Transfer Package Generation
- q Transfer Package Transfer
- q Transfer Package Receive



# Labs Chapter 5

## Lab 5-1: Ongoing Operations



# Advanced Topics



# Advanced Topics

- q **Automatic Receive**
- q **Remote Shell Services and ccm\_remd on NT**
- q **Work Area Maintenance**
- q **Ongoing Administration**
- q **Relationships and Miscellaneous Commands**
- q **Additional Information**



# Work Area Maintenance

## Problems:

- q If inappropriate file system, project creation fails
- q Can potentially run out of disk space
- q Project may be for a different client than where it was received
- q Path may not be visible or modifiable

## Solutions:

- q Set wa\_path\_template option in ccm.ini file
- q Disable work area maintenance for all received projects

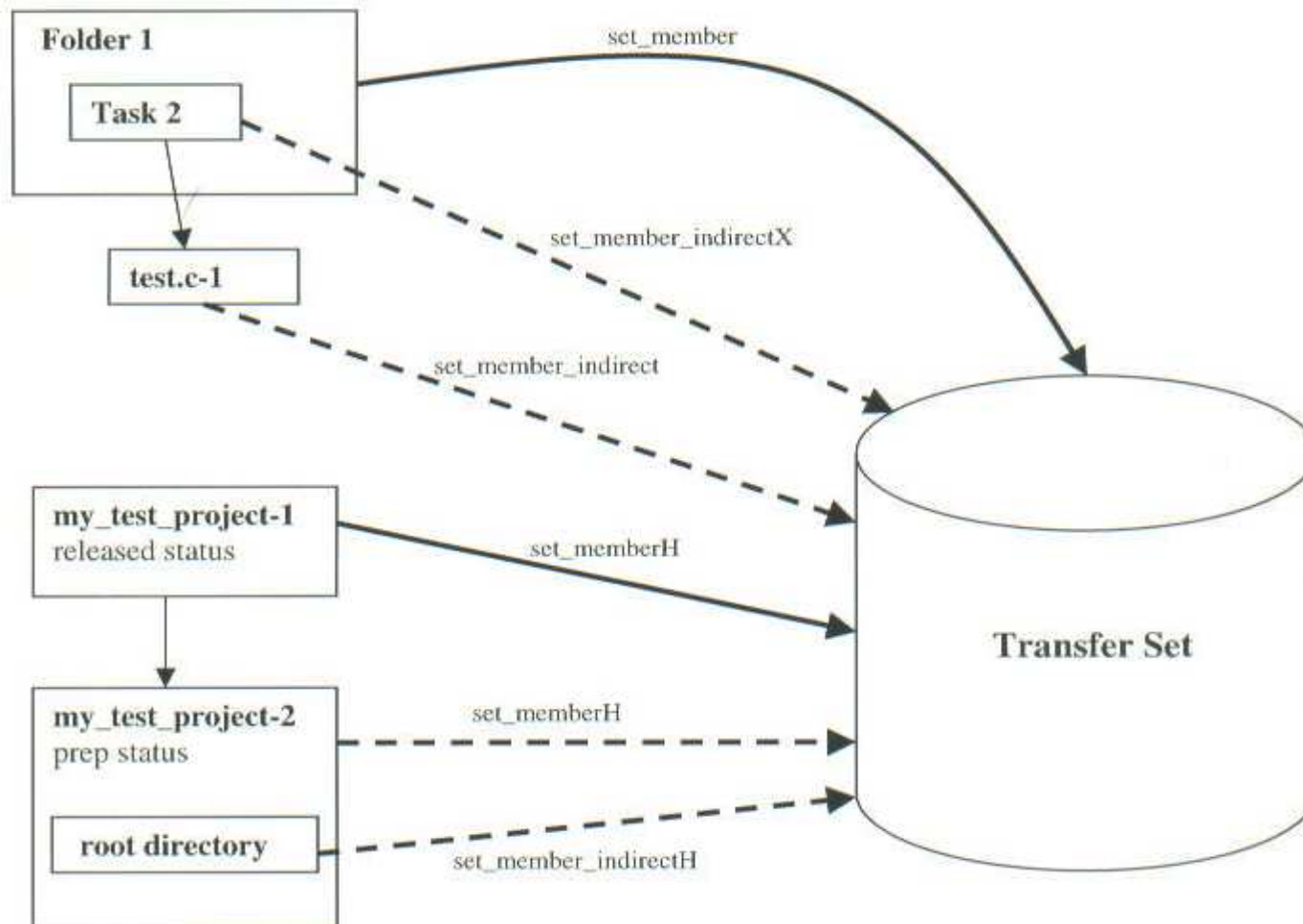


## Ongoing Administration

- q **Check log files and e-mail for errors and warnings**
- q **Verify that parallels are addressed**
- q **Monitor disk space availability**
- q **Monitor server dbspace availability**
- q **Check the contents of “dcm” directories**
- q **Perform and verify database backups**
- q **Optimize the timing of transfers**
- q **Review the Common Database Parameters**
- q **See “Distributed CM” for other Admin. activities**



# Relationships and Miscellaneous Commands





# Chapter 6: Labs

## Scenario One: Porting

- Lab 6-1: Prepare “odd” database for DCM
- Lab 6-2: Prepare “even” database for DCM
- Lab 6-3: Prepare and perform initial DCM from your “odd” database
- Lab 6-4: Prepare and initiate ongoing DCM transfers from your “odd” database
- Lab 6-5: Prepare “Prep” projects in “even” database
- Lab 6-6: Prepare and perform initial DCM transfer from your “even” database (Extra Credit)
- Lab 6-7: Perform ongoing DCM receive in “odd” database (Extra Credit)

## Scenario Two: Peer-to-Peer

- Lab 6-8: Prepare “odd” database for DCM



# Starting the tool

- How to start
- `ccm start -m -d /opt/synergy/ccmdb/odd_ws<nr> -s tornado`
- `ccm start -m -d /opt/synergy/ccmdb/even_ws<nr> -s tornado`