



IBM Software Group

# IBM Presentations: Discovery and Synchronization Design for WebSphere Application Server



 business on demand software

# Agenda

- Overview
- Repository
- Distributed Environment
- Discovery
  - ▶ Overview
  - ▶ Messages
- Synchronization
  - ▶ Overview
  - ▶ Synchronize Methods
  - ▶ File Transfer
- Problem Determination
- Question and Answer

# Overview

- Separated system administration functionality
- Maintain order and reduce single-point-of-failure
  - ▶ Discovery
  - ▶ Synchronization
- This session reviews
  - ▶ Design and implementation of each protocol
  - ▶ WSAS trace

# Overview: Skills

- Entry Skills:
  - ▶ General knowledge of WebSphere administrative tools
  - ▶ Ability to read an xml file
- Exit Skills:
  - ▶ Understand the design and need for discovery and synchronization
  - ▶ Ability to perform problem determination for synchronization and discovery issues

# Repository

# Repository: Overview

- Repository contains configuration documents
- Under configuration root directory
  - ▶ <WAS\_HOME>/config
- Cell repository and Node repositories
  - ▶ Cell repository is the master configuration
  - ▶ Node repositories contain applicable subset of the configuration in a local copy
  - ▶ Updates are **one-way**: from master to nodes
- Repository service exists in every server

# Repository: Configuration Documents

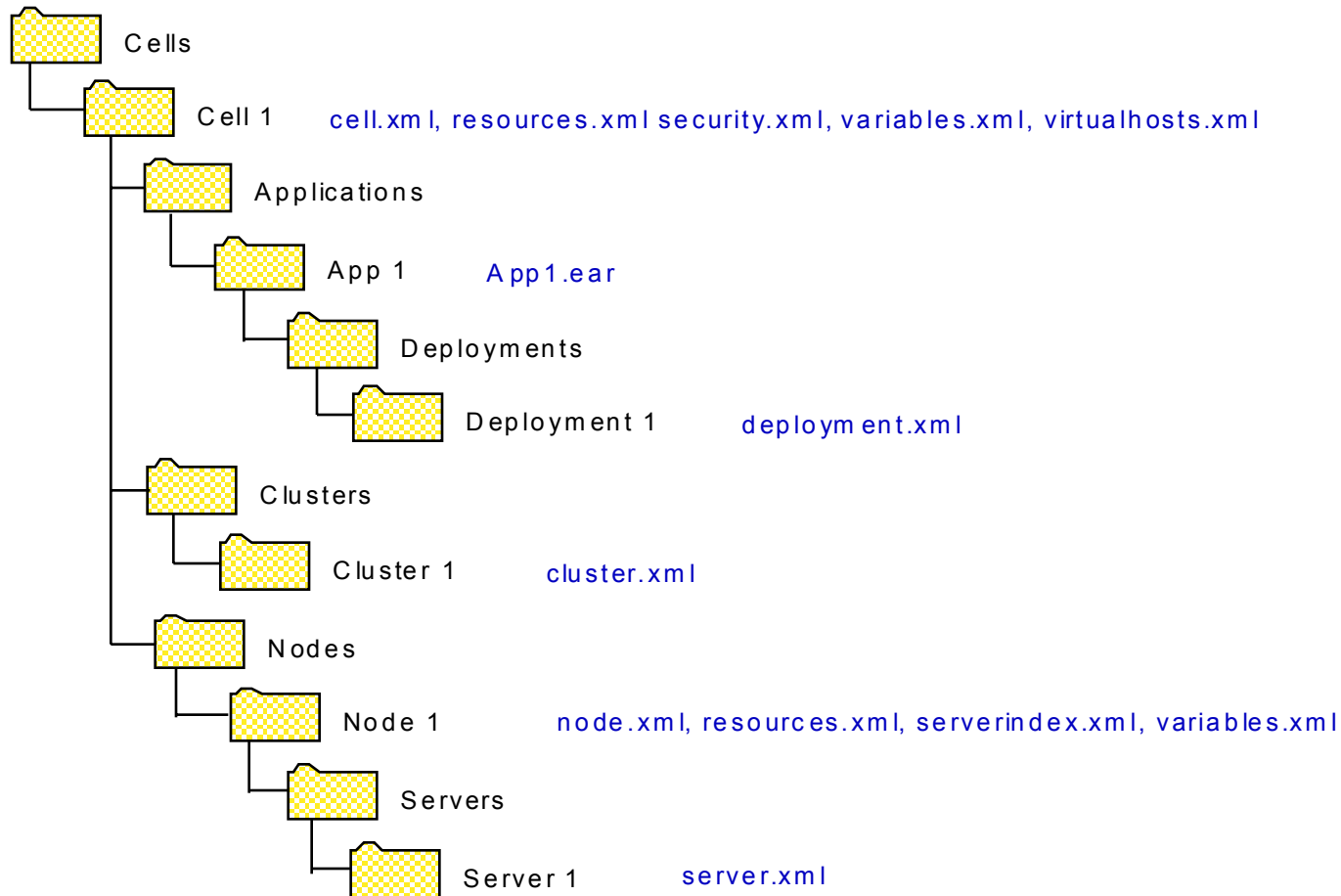
- WebSphere Application Server (WSAS) Configuration Documents
  - ▶ File types
    - Most have XML content
    - Application binary files are also managed
    - Other files, e.g. client files (not recommended)
  - ▶ Changes to Deployment Manager (dmgr) configuration documents are replicated to the node during a synchronization
    - Changes to node configuration will be temporary
  - ▶ Cascaded hierarchy of directories
    - Lower level configuration documents override higher level configuration documents
    - e.g. Node documents override cell documents if there is a conflict of interest

# Repository: Configuration

- Servers read their configurations directly from the configuration files
- Edit configuration documents
  - ▶ Script
  - ▶ Wsadmin commands
  - ▶ Admin Console
  - ▶ WSAS API Program
  - ▶ Manual edit files (not recommended)



# Repository: Config Structure

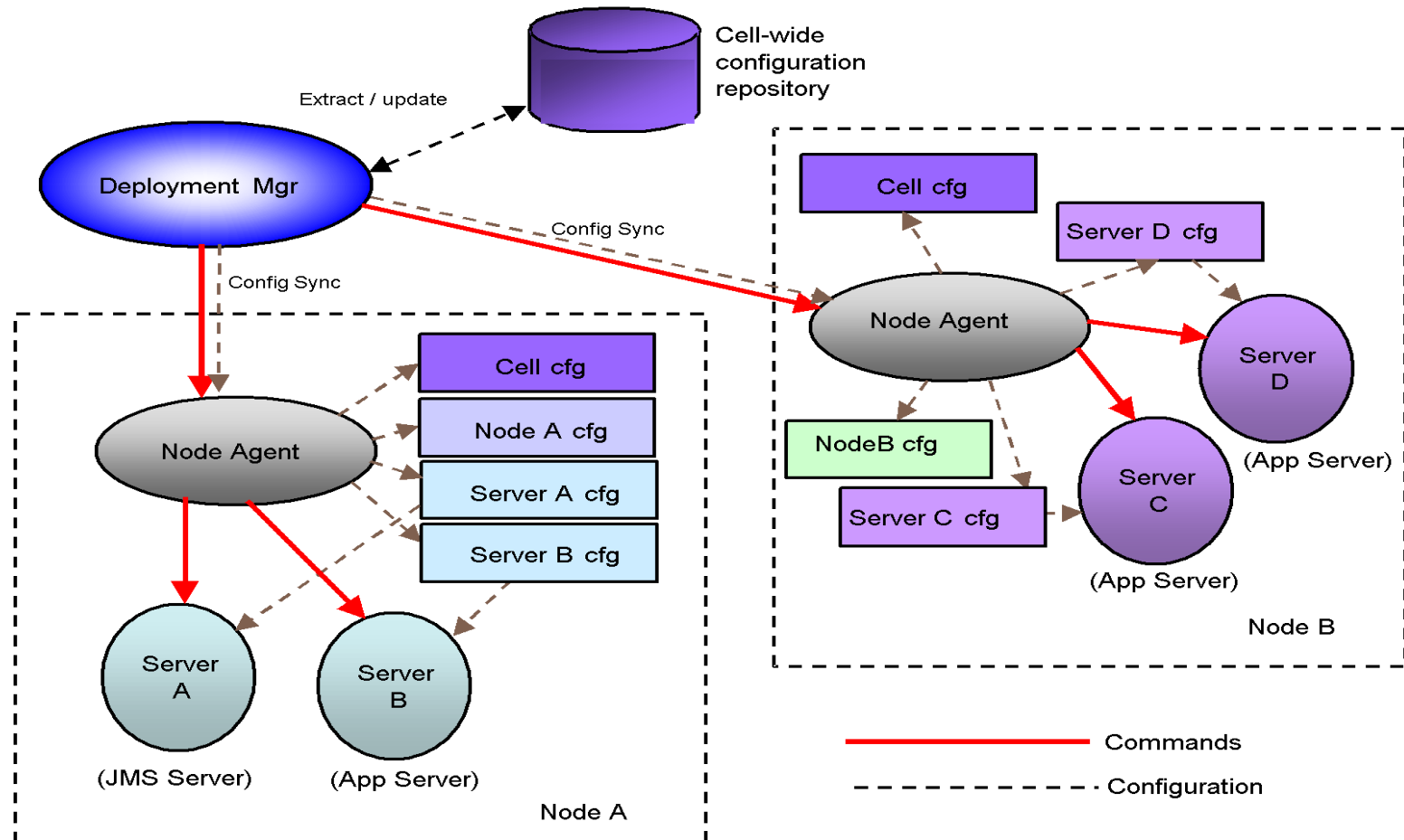


# Distributed Environment

# Distributed Environment: Federation

- Base Application Server installation can be converted into a member of a cell (federate)
- Procedure
  - ▶ Install and run the Deployment Manager in the network
  - ▶ Federate the node
    - Run addNode script on the BASE node (point to dmgr)
    - Use Admin Console, “Nodes” -> “Add Node”, on dmgr (point to server1)
  - ▶ Node configuration is added into the cell-wide configuration store
  - ▶ Configuration information is downloaded to the node so that it is synchronized with the cell
  - ▶ By default, the NodeAgent is started on the node
- No installation of new software is required on the BASE server's machine

# Distributed Environment: Picture



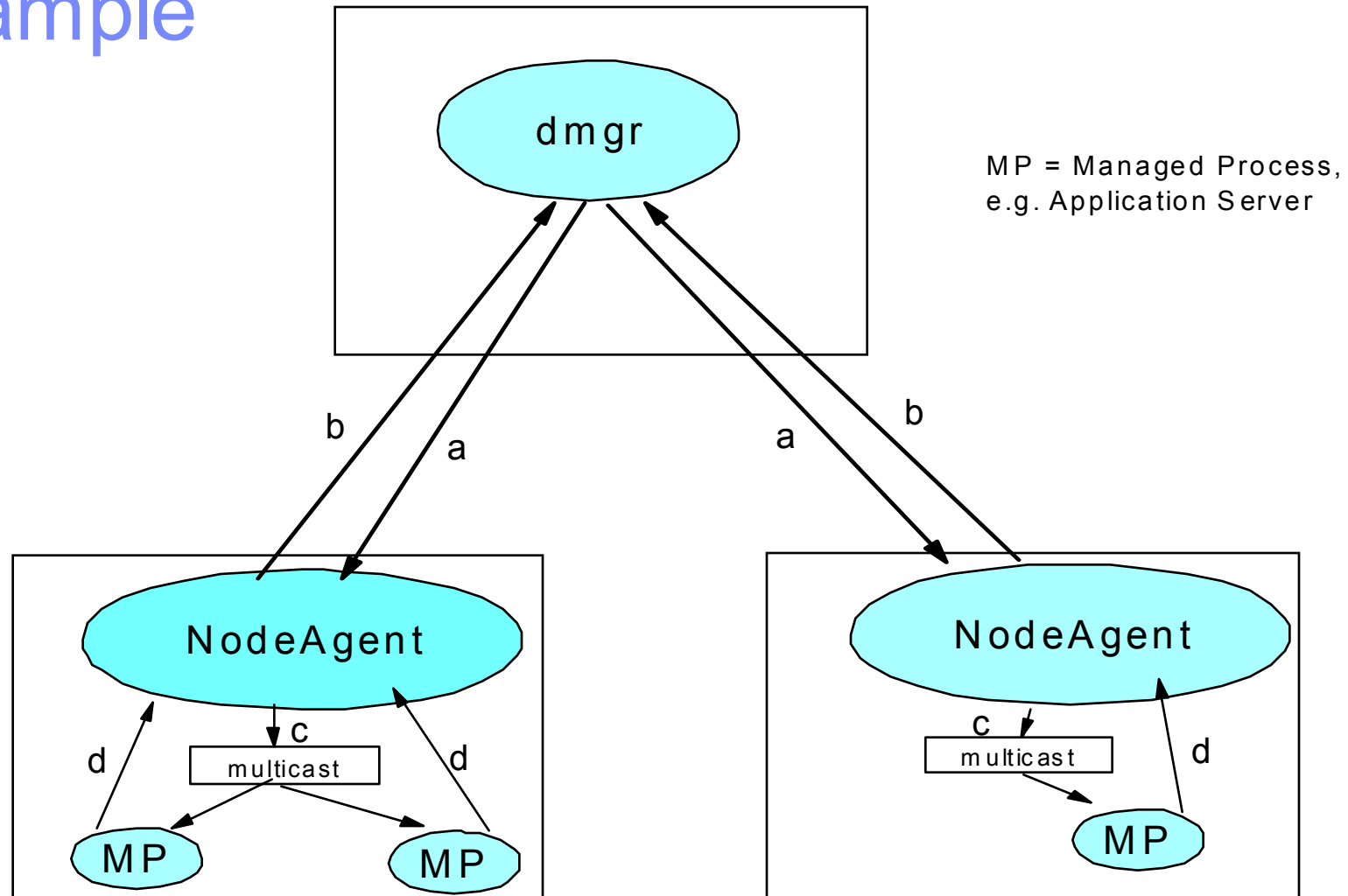
# Discovery

# Discovery: Overview

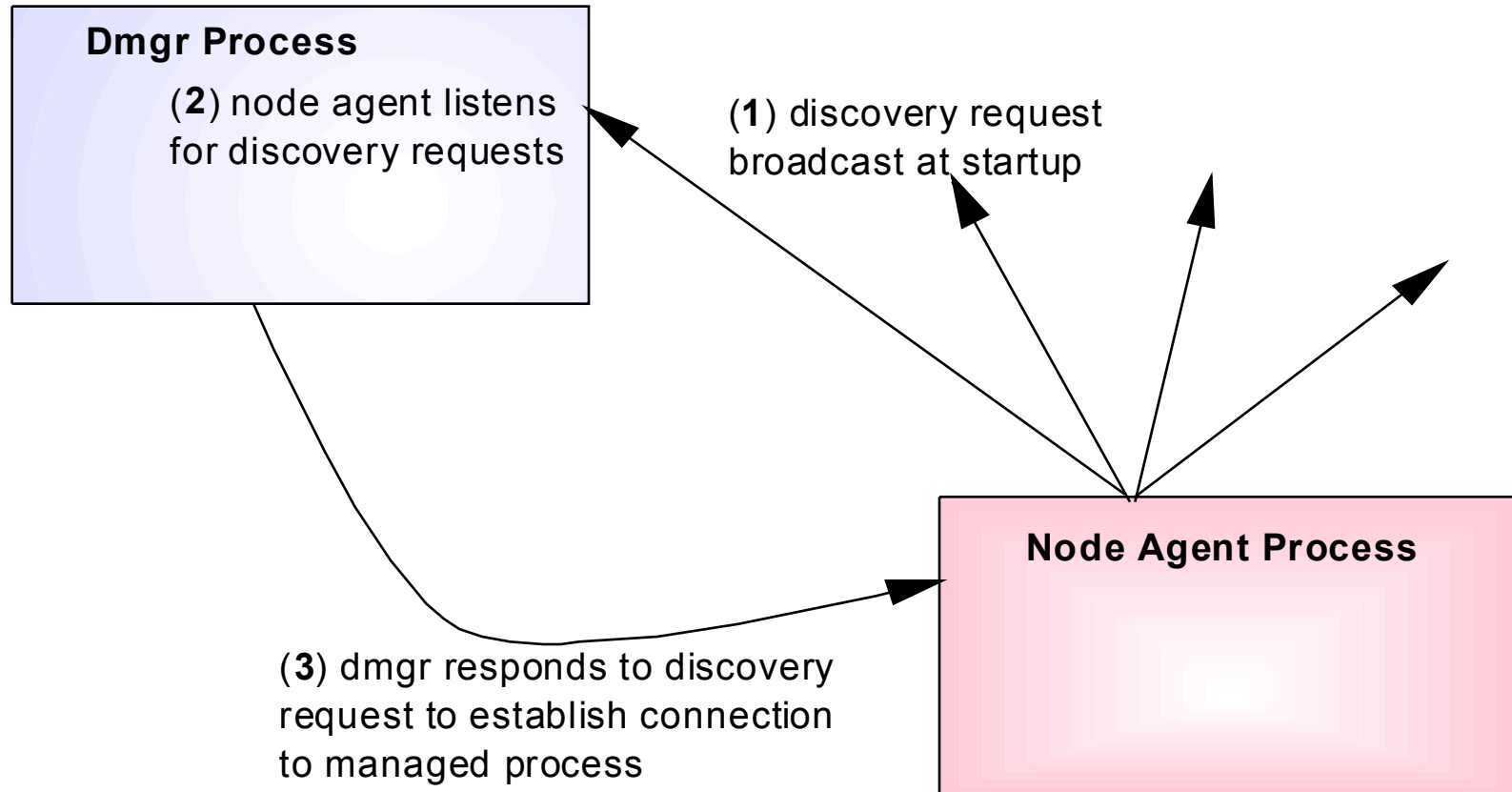
- Discovery Protocol
  - ▶ TCP (recommend), UDP and Multicast (Only Application Server)
- Sender
  - ▶ Listen on discovery port
  - ▶ Send Discovery Query Message to peer's discovery port:
    - Retry interval: 1, 2, 4, 8 minutes, etc.
  - ▶ Peer sends back Discovery Response Message to sender's discovery port
  - ▶ Process Discovery Response Message
    - Update routing table
      - Start process monitoring
- Receiver
  - ▶ Listen on discovery port
  - ▶ Process Discovery Query Message
    - Update routing table
      - Start process monitoring
  - ▶ Send Discovery Response Message to sender's discovery port

Note: If server enables “Standalone” under “Administration Services”, this disables discovery.

# Discovery: Distributed Environment Example



# Discovery: Detailed Example





# Discovery: MBean

- Supports query and respond methods as backup to multicast
- Registered as RoutingListener to report discovery of parent/children in the log
  - ▶ SystemOut.log: [6/28/04 11:58:54:903 EDT] b875025  
DiscoveryMBea I ADMD0023I: Process discovered (name: dmgr, type: DeploymentManager, pid: 4344)
- Emit Notifications
  - ▶ Child discovered:  
TYPE\_DISCOVERY\_PROCESS\_FOUND
  - ▶ Child lost: TYPE\_DISCOVERY\_PROCESS\_LOST
  - ▶ Parent discovered: TYPE\_DISCOVERY\_AGENT\_FOUND
  - ▶ No notification for parent lost

# Discovery: Message Breakdown

- Discovery Query Message and Discovery Response Message
  - ▶ Query id
    - Sequence number used by sender to keep track of messages
  - ▶ Peer advertisement
    - Cell, node
    - Role: Whether it's cell manager, node agent, or managed process
    - Services: Info about connectors to access AdminService - RMI, SOAP
    - EndPoints: Location of discovery port
  - ▶ Target
    - Cell, node, role
- Representation
  - ▶ In memory as ServerInfo
  - ▶ On the wire as XML

# Synchronization

# Synchronization: Overview

- To move or remove configuration files
- Dmgr (CellSync MBean) cooperates with NodeAgent process to perform sync operation
- NodeAgent (NodeSync MBean) drives operation to compare repositories and get updates
- Synchronization and Full Resynchronization

# Synchronization: Concepts

- Auto Sync: Automatic, periodic synchronization
- Startup Sync: Synchronization on Application server startup
- Manual Sync: Explicit invocation
- Configurable properties
  - ▶ `autoSyncEnabled`
  - ▶ `syncInterval`
  - ▶ `syncOnServerStartup`
- Synchronization uses an optimized algorithm that monitors the repository

# Synchronization: Epoch/Digest

- Epoch
  - ▶ Epoch is a value WSAS assigns to the configuration overall and folders
  - ▶ Determines if synchronization should compare file digest values
  - ▶ Types of Epoch:
    - Repository
      - Master cell repository epoch
      - Node repository epoch
    - Folder
- Digest
  - ▶ Calculated value for each configuration file in local configuration
  - ▶ Determines specifically what files changed inside the config directory
- Deployment Manager and NodeAgent store in memory the values of Epoch
- Any differences in the repositories will be picked up and brought down to the node

# Synchronization: Installed Applications

- Installed applications are treated like the other configuration files in the repository and are synchronized
  - ▶ EAR file and configuration files associated to application
  - ▶ Normal synchronization operations will synchronize the modified application files through the Administration programs
  - ▶ After the EAR configuration files are moved, WSAS notifies the NodeAgent to expand the ear file (Outside the synchronization process)
- Every time the EAR file is synchronized, this notifies the NodeAgent to expand in the appropriate location on that node

# Synchronization: Status

- Synchronization status based on
  - ▶ Startup Sync
    - If enabled, synchronizes every time the NodeAgent starts an application server
    - May cause startup delays
  - ▶ Auto Sync
    - Synchronizes occur at designated intervals (Synchronization Interval)
    - Status of synchronization reported in the Admin Console
    - Recommended for production environments to either set the sync interval to a high value or disable auto sync
  - ▶ When Auto Sync is disabled
    - Status of synchronization will always be "Not Synchronized" on a node restart
    - Value is changed to synchronized only when a manual synchronize is forced
- Status displayed on the console is strictly dependent on the values set between dmgr and NodeAgent synchronization



# Synchronization: Synchronization

- Synchronization only moves marked files
- Determined by checking changed epoch folder values
  - ▶ Four phase operation
    - Check global epoch of the cell repository
    - Check folder level epochs
    - Compare document digests in changed folders
    - Move only changed files
  - ▶ Execution
    - Auto sync (enabled by default)
    - Manually
      - During save
      - “Synchronize” button
- During configuration synchronization operations, if the repository epoch has changed since the previous synchronization operation, then individual folder epochs will be compared

# Synchronization: Full Synchronization

- Configuration modifications outside the WSAS APIs, don't change epoch values
- Full Synchronization reset the overall epoch so all file digests are recalculated and compared
  - ▶ Three phase operation
    - Resets all epoch values
    - Recalculates all file digest
    - Move all files that have different digest values
  - ▶ Execution
    - Manually Induced (syncNode, Full Resynchronization)
    - On dmgr and NodeAgent restarts

# Synchronization: SyncNode script

- SyncNode.sh/bat command
  - ▶ Forces a full synchronization to occur between the node and the dmgr for the cell
  - ▶ Only use when NodeAgent fail to work because the node configuration does not match the cell configuration
  - ▶ Run from <WAS\_HOME>/bin directory
- SyncNode command does not change the status of the node
- Note: NodeAgent cannot be running for this script to work.

# Synchronization: Administrative Tools

- Wsadmin commands

1. Identify the ConfigRepository MBean and assign it to variable 'a':

```
wsadmin>set a [$AdminControl completeObjectName  
type=ConfigRepository,process=nodeagent,node=nodename,*]
```

=>Returns a MBean.

2. Refresh the node repository epoch to force a "Full synchronization":

```
wsadmin>$AdminControl invoke $a refreshRepositoryEpoch
```

=>Returns an integer

3. Set the variable 'b' for node synchronize:

```
wsadmin>set b [$AdminControl completeObjectName  
type=NodeSync,node=nodename,*]
```

=>Returns a MBean.

4. Synchronize by issuing the following command:

```
wsadmin>$AdminControl invoke $b sync
```

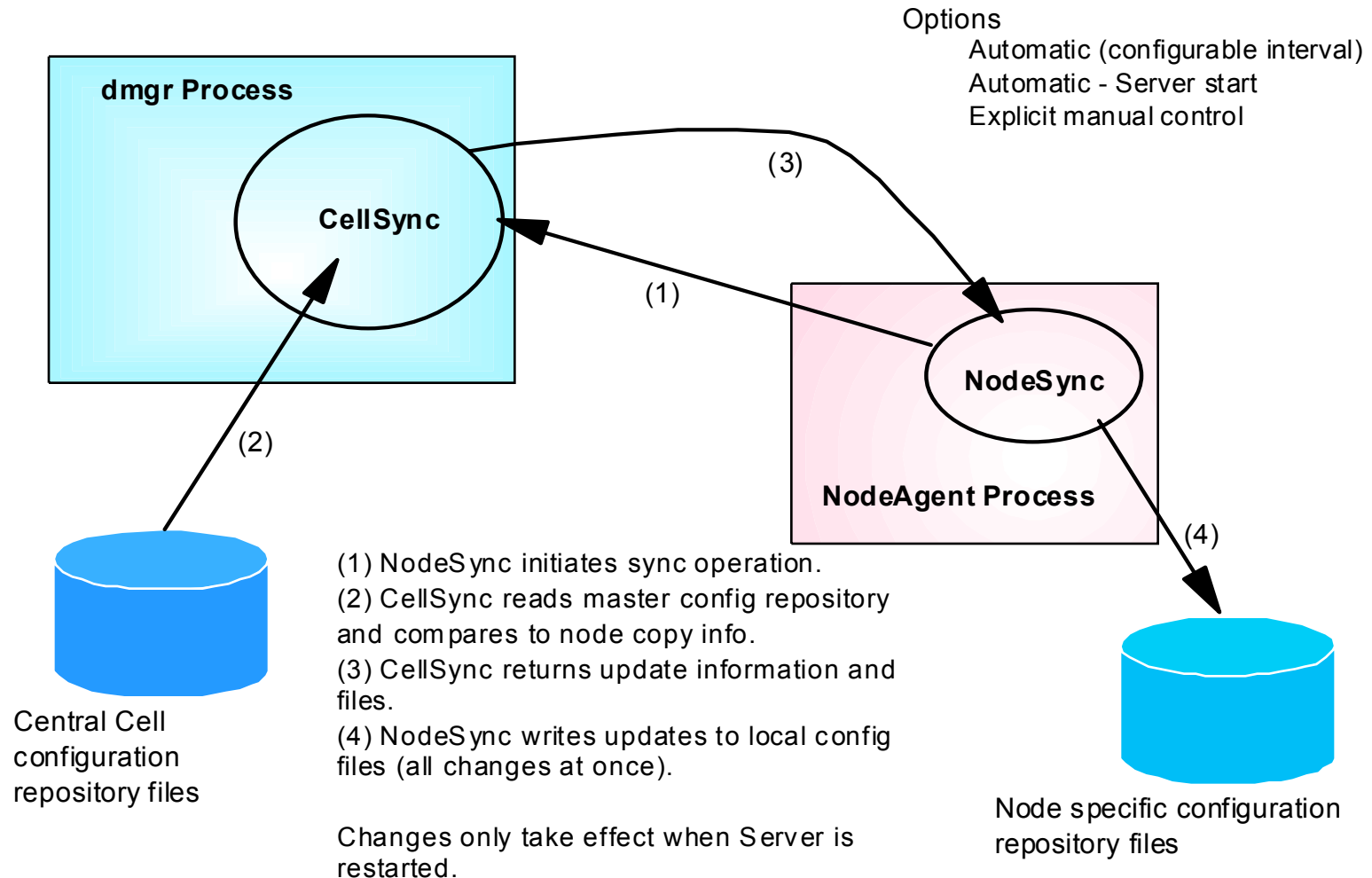
- Administrative Console

1. Expand the **System Administration** section and select **Nodes**
2. Check the box next to the nodes that are not synchronized
3. Click **Synchronization** or **Full Resynchronize**

# Synchronization: JMX MBean

- CellSync MBean ("type=CellSync,\*")
  - ▶ syncNode
- NodeSync MBean ("type=NodeSync,node=node 1,\*")
  - ▶ requestSync
  - ▶ sync
  - ▶ isNodeSynchronized
  - ▶ getSyncResult
- Notifications
  - ▶ websphere.nodesync.initiated
  - ▶ websphere.nodesync.complete

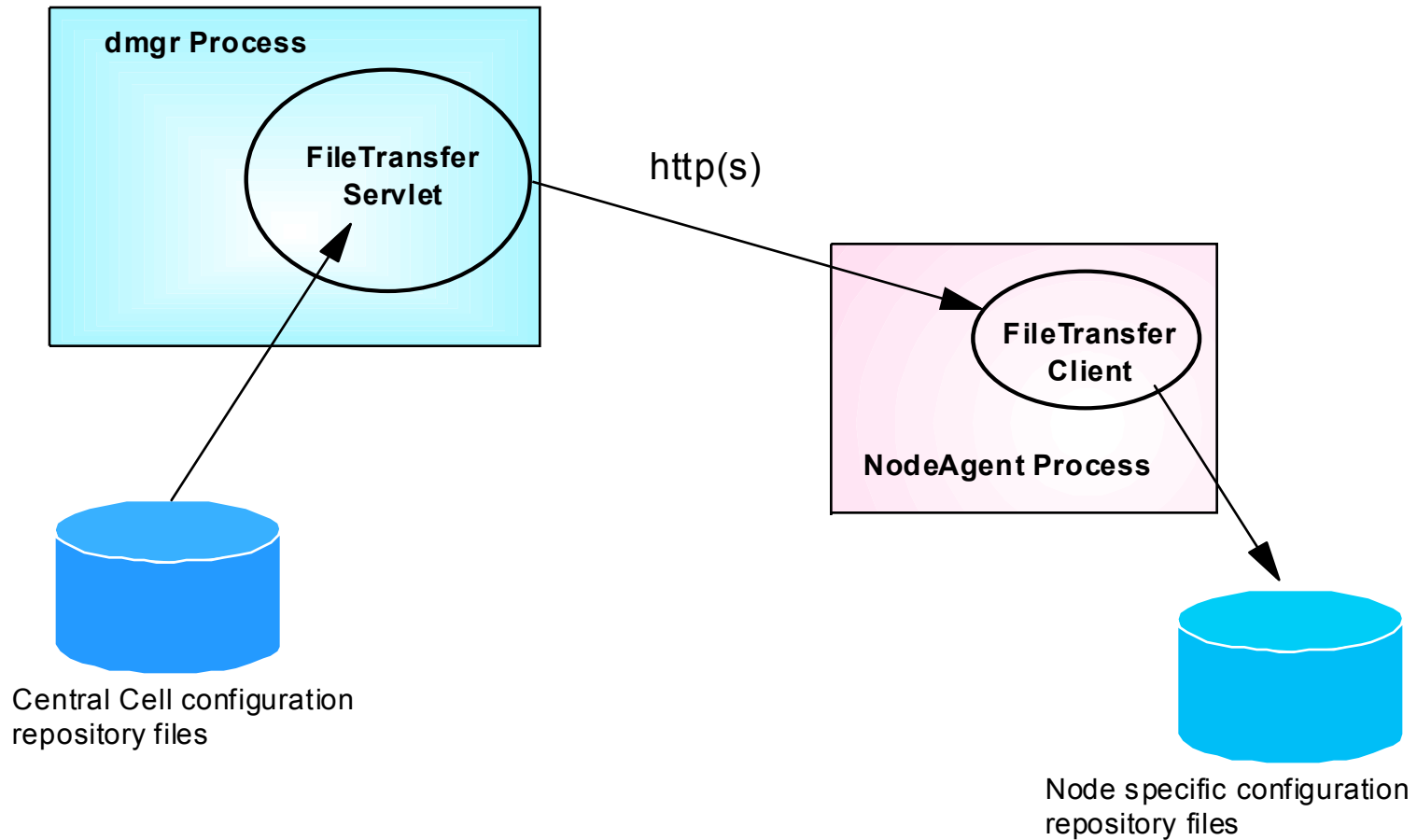
# Synchronization: Picture



# Synchronization: File Transfer

- File synchronization service keep the configuration documents across the cell up-to-date
- This service runs in the dmgr and NodeAgents
- Ensures that the configuration changes made to the cell repository are propagated to the appropriate node repositories

# Synchronization: File Transfer Service





# Problem Determination

# PD: Overview

- Determine if this is a discovery issue or synchronization issue
  - ▶ Admin Console Status: “Not synchronized” vs. “Unknown”/“Unavailable”
- Discovery MustGather Documents
  - ▶ WSAS Discovery Trace
    - Enable tracing on all processes in question
    - TraceString:  
`com.ibm.ws.management.discovery.*=all=enabled`
  - ▶ Network/IP tracing
- Synchronization MustGather Documents
  - ▶ WSAS Synchronization Tracing

# PD: Discovery Issues

- Review the trace files
- Did the discovery broadcast a message and get a connection?
  - ▶ trace.log: [6/28/04 11:58:55:384 EDT]  
604a9020 TcpMessenger < sendMessage -  
successful
- What did the Discovery Query and Response state?
- If the receiver never received the message, review possible network issues.

# PD: Broadcast Discovery

**[6/28/04 11:58:55:374 EDT] 604a9020 DiscoveryServ d message[src, dest]  
tcp://9.37.37.37:7272/DiscoveryService  
tcp://9.37.37.37:7277/DiscoveryService**

**...**

**[6/28/04 11:58:55:374 EDT] 604a9020 TcpMessenger > openSocket  
[6/28/04 11:58:55:374 EDT] 604a9020 TcpMessenger d InetAddress, port:  
9.37.37.37/9.37.37.37  
7277**

**...**

**[6/28/04 11:58:55:384 EDT] 604a9020 TcpMessenger < openSocket - good**

**...**

**[6/28/04 11:58:55:384 EDT] 604a9020 TcpMessenger < sendMessage -  
successful**

**...**

**[6/28/04 11:58:55:384 EDT] 604a9020 TcpConnection d End of  
inputstream. Closing socket**

# PD: Discovery Query Message

[6/28/04 11:58:52:219 EDT] 604a9020 TcpConnection > run

...  
[6/28/04 11:58:52:239 EDT] 604a9020 DiscoveryServ d the incoming discovery message is:

```
<?xml version="1.0" encoding="UTF-8"?>
<jxta:DiscoveryQuery><Type>0</Type><QueryId>1</QueryId><PeerAdv><jxta:PeerA
dvertisement><Name>dmgr</Name><Pid>4344</Pid><Version>5.0.2.2</Version><
Cell>ibmNetwork</Cell><Node>ibmM anager</Node><Role>DeploymentM anager</
Role><Endpoint><jxta:EndpointAdvertisement><Name>tcp</Name><Address>tcp
://ibm:7277</Address><Transport><jxta:TCPTransportAdvertisement><Protocol>t
cp</Protocol><Port>7277</Port><PublicAddress>ibm</PublicAddress><LocalAdd
ress>ibm</LocalAddress></jxta:TCPTransportAdvertisement></Transport></jxta:
EndpointAdvertisement></Endpoint><Service><jxta:ServiceAdvertisement><Nam
e>AdminService</Name><AccessMethod>endpoint:name=BOOTSTRAP_ADDRES
S;host=ibm;port=9809</AccessMethod><AccessMethod>connector:type=SOAP;h
ost=ibm;port=8879;preferred=true;isInternal=true</AccessMethod><AccessMetho
d>connector:type=RMI;host=ibm;port=9809;isInternal=true</AccessMethod></jxta:
ServiceAdvertisement></Service></jxta:PeerAdvertisement></PeerAdv><Cell>ib
mNetwork</Cell><Node>ibm</Node><Role>Node Agent</Role></jxta:DiscoveryQu
ery>
```

...  
[6/28/04 11:58:54:903 EDT] b875025 DiscoveryM Bea l ADM D0023l: Process discovered (name: dmgr, type: DeploymentManager, pid: 4344)

...  
[6/28/04 11:58:55:384 EDT] 604a9020 TcpConnection d End of inputsteam. Closing socket

# PD: Discovery Response Message

[6/28/04 11:58:55:384 EDT] 7d9b9044 TcpConnection > run

...  
[6/28/04 11:58:55:384 EDT] 7d9b9044 DiscoveryServ d the incoming discovery message is:

```
<?xml version="1.0" encoding="UTF-8"?>
<jxta:DiscoveryResponse><Type>0</Type><QueryId>1</QueryId><Response><jxta:
PeerAdvertisement><Name>node agent</Name><Pid>3316</Pid><Version>5.0.2.2
</Version><Cell>ibmNetwork</Cell><Node>ibm</Node><Role>Node Agent</Role>
<Endpoint><jxta:EndpointAdvertisement><Name>tcp</Name><Address>tcp://ibm
:7272</Address><Transport><jxta:TCPTransportAdvertisement><Protocol>tcp</P
rotocol><Port>7272</Port><PublicAddress>ibm</PublicAddress><LocalAddress>
ibm</LocalAddress></jxta:TCPTransportAdvertisement></Transport></jxta:Endpo
intAdvertisement></Endpoint><Service><jxta:Service Advertisement><Name>Adm
inService</Name><AccessMethod>endpoint:name=BOOTSTRAP_ADDRESS;host=ib
m;port=2809</AccessMethod><AccessMethod>connector:type=SOAP;host=ib
m;port=8878;preferred=true;isInternal=true</AccessMethod><AccessMethod>con
nector:type=RMI;host=ibm;port=2809;isInternal=true</AccessMethod></jxta:Servi
ce Advertisement></Service></jxta:PeerAdvertisement></Response><Cell>ibmNet
work</Cell><Node>ibmManager</Node><Role>DeploymentManager</Role></jxta:
DiscoveryResponse>
```

...  
[6/28/04 11:58:57:937 EDT] 1888d04c DiscoveryMBea I ADMD0023I: Process discovered (name: ibm, type: NodeAgent, pid: 3316)

...  
[6/28/04 11:58:58:268 EDT] 7d9b9044 TcpConnection d End of inputsteam. Closing socket

# PD: Discovery initDiscovery Fails

- Discovery fails to initialize.
  - ▶ No discovery message is broadcasted
  - ▶ Discovery Port is not listening

- Error from NodeAgent's trace.log:

[10/28/03 13:57:32:844 EST] 42570e39 JMXConnectors d processing server:

**TestServer**

[10/28/03 13:57:32:844 EST] 42570e39 JMXConnectors d processing endPoint:  
SOAP\_CONNECTOR\_ADDRESS

...  
[10/28/03 13:57:33:125 EST] 42570e39 JMXConnectors d Got Exception initializing  
discovery:

**java.lang.NullPointerException**

at

com.ibm.ws.management.component.JMXConnectors\$EndPtCollector.<init>(JMXConnectors.java:957)

at

com.ibm.ws.management.component.JMXConnectors.initDiscovery(JMXConnectors.java:195)

## PD: Discovery initDiscovery (Answer)

- The "serverindex.xml" from above NodeAgent

```
<serverEntries xmi:id="ServerEntry_1065794240606" serverName="
TestServer " serverType="APPLICATION_SERVER">
  <specialEndpoints xmi:id="NamedEndPoint_1065794240871"
  endPointName="SOAP_CONNECTOR_ADDRESS">
    <endPoint xmi:id="EndPoint_1065794240621" host="ibm.com"
    port="8880"/>
  ...
</serverEntries>
```

- Conclusion/Mismatch:

This "TestServer " does not have a directory in the servers level where the server.xml file located so that causes the whole issue.



# PD: Synchronization Issues

- Understand the synchronization problem:
  - ▶ Verify the File Transfer enterprise application is installed and running on the dmgr
  - ▶ Execute a full resynchronization for the problem node to double check if synchronization initialized
- Execute syncNode script
  - ▶ syncNode.log: [8/3/04 20:51:31:848 EDT] d25caaf AdminTool A ADMU0402I: The configuration for node ibm has been synchronized with Deployment Manager ibm: 8879
  - ▶ Note: this will move files and not validate the information
- If admin console show “Not Synchronized” after “Full Resynchronize”
  - ▶ MustGather Website

<http://www-1.ibm.com/support/docview.wss?rs=180&context=SSCMPDZ&uid=swg21140941>

# Questions and Answers