ORACLE®

GRID INFRASTRUCTURE

An Oracle Reference Guide
March, 2013

Oracle Grid Infrastructure Bundled Agents *v2*

11.2.0.3+

Table of Contents

ORACLE

ORACLE®

High Availability Components for Applications

The Oracle Grid Infrastructure provides the necessary components to manage high availability (HA) for any business critical application. Oracle Grid Infrastructure Bundled Agents (XAG) are Oracle Grid Infrastructure components that provide the HA framework to application resources and resource types managed through the bundled agent management interface, AGCTL. This framework provides a complete, ready to use application HA solution that contains pre-defined Oracle Grid Infrastructure resource configurations and agents to integrate applications for complete application HA.

The Oracle Clusterware[1], a component of the Oracle Grid Infrastructure, provides a pre-configured public core network resource, ora.net1.network, to which applications can bind application Virtual IPs (APPVIPs) for network connectivity HA. Administrators can easily create additional network resources on distinct network interfaces, as required, for consolidation, network isolation and scalability. Oracle Grid Infrastructure will manage the network stack availability as well as all Oracle Grid Infrastructure resources with defined dependencies on the root network resource.

Applications require shared storage for binary, trace, metadata and state files. Shared storage enables the application to run on any node in the cluster by maintaining a single copy of binaries & configuration metadata and provides a persistent state tracking and restoration mechanism in the event of failover. Any Oracle certified shared or clustered file system can be used to satisfy the application shared storage requirements. This includes clustered file systems and failover file systems. To manage failover of OS native file systems, administrators will have to build the necessary script agent using the Oracle Clusterware script agent API. The Oracle Grid Infrastructure ships with the Automatic Storage Management Clustered File System (ACFS) which administrators can easily configure and present to the cluster for these purposes. An ACFS file system will be mounted on all user defined nodes and file system availability is managed entirely by the Oracle Grid Infrastructure. To enhance and automate application availability, applications can define start/stop dependencies on the ACFS configured file system resource, 'ora.<diskgroup>.<volume>.acfs'[2], so that the application is started, stopped or relocated as the ACFS resource comes online or goes offline. ACFS is the recommended HA file system for applications configured for Oracle Grid Infrastructure HA.

The Oracle Grid Infrastructure Bundled Agents leverage these infrastructure components and allow for a simple integration which eliminates the need for supplementary infrastructure agents for network, IP and file system HA. In addition, the Oracle Grid Infrastructure provides rich resource modeling which applications simply plug into. Each application entity is represented and managed as a resource within the Oracle Grid Infrastructure. This resource modeling provides fine-grained resource state reporting, inherent placement policies, and pre-configured resource dependencies for the bundled agents.

Bundled agents will also leverage existing databases and database services for a tighter dependency between the application and the database providing an even greater end-to-end availability model. Applications that connect to an Oracle Database and run in the same cluster can define dependencies on the database. If you configure a database dependency, the database will start up before the application so that it is open and available to the application when needed. For applications that connect to the database using the database SID, these applications

---

[1] Please see http://oracle.com/goto/clusterware for more details and links to specific Oracle Clusterware documentation

[2] Please see http://docs.oracle.com/cd/E11882_01/server.112/e10500/asmfilesystem.htm

ORACLE

should configure the '--*databases'* dependency. This will ensure that the application will run on the same node as the database. For applications that use TNS services to connect to the database, and thus can run on any node in the cluster and do not have the co-location requirement with the database or database instance, these applications should define their dependency on a *'--db_ services'*. This will allow, if so configured, the application and database to run on separate nodes in the cluster. This deployment model offers the greatest flexibility and ease of management. In both cases, the database will be started before the application starts. As with the ACFS file system dependency, the application should use the 'ora.database.db' , or ora.database.service.svc' Oracle Clusterware resource naming convention for the dependency definition. Please see the examples in the respective 'Syntax' sections.

AGCTL

AGCTL, Agent Control, is the agent command line utility to manage bundled agents (XAG) for application HA using Oracle Grid Infrastructure. AGCTL is the interface to add an application to the Oracle Grid Infrastructure as a clustered resource under agent control. Once configured, AGCTL is the management interface to bring the application online, offline or relocate, as well as to check the application state and modify the agent configuration. There are AGCTL options to disable and remove the application from the Oracle Grid Infrastructure agent control.

Installation and Configuration

The Oracle Grid Infrastructure bundled agents require an operational installation of the Oracle Grid Infrastructure version 11.2.0.3 or later on all nodes where the application is targeted to run. The bundled agents for 11.2.0.3 are available for download on the http://oracle.com/goto/clusterware download tab. Oracle Grid Infrastructure releases after 11.2.0.3 will include bundled agents as part of the standard distribution and installed by the Oracle Universal Installer in the Grid Infrastructure home directory.

The bundled application agents and the AGCTL management interface are packaged for release 11.2.0.3 in the zip file xagpack.zip. To begin installation, this zip file needs to be downloaded and expanded in a temporary directory. The $XAG_HOME and sub-directories must be owned by Oracle Grid Infrastructure install owner. The setup script, xagsetup.sh, must be run as the Oracle Grid Infrastructure install owner. The xagsetup.sh script offers the option to install local only (the default) or to install on all nodes in the cluster or, a subset of nodes in the cluster where the application is targeted to run. When deploying the bundled agents to multiple nodes in the cluster, the xagsetup.sh is executed on only one node of the cluster, and the xagsetup.sh script will attempt to create $XAG_HOME directory on all remote nodes, with an identical absolute path and the correct ownership and permissions. Please review the readme.txt file in the $XAG_HOME directory for details.

An identical, operational installation of the application must be available on all nodes in the cluster targeted to support application failover. Application version, directory location, ownership and permissions must be the same on all nodes and the application must be configured conforming to the application documentation.

The bundled agents may be deployed as a new install where the agents are installed when the Oracle Grid Infrastructure is installed and configured. As many applications depend on application VIPs, the Oracle Grid Infrastructure should be configured and functional prior to installing the application. The Oracle Grid Infrastructure and bundled agent installs are orthogonal to an existing application deployment. This allows installation of the Oracle Grid Infrastructure and the bundled agents to be in separate directory structures and configuration of the Oracle Grid Infrastructure stack in an existing application environment. In this situation, typically the application stack will need to be stopped, the correct communication endpoint for the application to use will be configured for the respective VIP and the application will be restarted under agent control using the AGCTL interface.

Requirements for Running AGCTL

AGCTL can be run as the Oracle Grid Infrastructure owner or a defined application administrator. The application administrator must be in the primary OS group of the Oracle Grid Infrastructure owner. For example, the Oracle Grid Infrastructure is installed by the oracle user in the group oinstall. The application administrator must also be in the oinstall group.

All AGCTL commands can be run by an application administrator. When creating application virtual IPs, administrators can either use the Oracle Grid Infrastructure utility appvipcfg, located in the Oracle Grid Infrastructure bin directory, or the VIP can be created using the AGCTL utility. In both cases, as the VIP is considered a super-user (root) owned resource, root access is required to create the VIP. If an application VIP is pre-created by the Oracle Grid Infrastructure administrator (as root), the application admin need only pass in the *--vip_name* on the AGCTL command line. This does not require super-user (root) privileges.

The Oracle Grid Infrastructure bundled agents and the AGCTL management interface support multiple application administrators for distinct applications. Application resource ownership and execution is controlled by UNIX like ACL definitions for a given resource and guarantees secure role separation. Role separation, execution privileges and ownership for an application are defined when the management of the application is presented to the Oracle Grid Infrastructure using the AGCTL management interface.

AGCTL Syntax and Use

```
$ ./agctl

 Manages Apache Tomcat, Apache Webserver, Goldengate, Siebel Gateway,
Siebel Server as Oracle Clusterware Resources

 Usage: agctl <verb> <object> [<options>]

    verbs:
add|check|config|disable|enable|modify|relocate|remove|start|status|stop

    objects:
apache_tomcat|apache_webserver|goldengate|siebel_gateway|siebel_server

 For detailed help on each verb and object and its options use:

    agctl <verb> --help or
```

AGCTL is the command to be used for application resource management. It is not supported to use the Oracle Grid Infrastructure utility CRSCTL to manage application resource of type xag, however, CRSCTL status calls for resource status, such as `crsctl status resource −t`, are supported.

ORACLE®

Oracle Grid Infrastructure Bundled Agents for Applications

Oracle GoldenGate

Oracle GoldenGate is a real-time database replication and data integration product. An Oracle GoldenGate instance consists of the **manager** process and its child processes, related programs, configuration files, checkpoint files, trail files and other logs/traces/templates/scripts. The child processes are named: *extract*, which captures change data, and *replicat*, which applies change data. There is exactly one running GoldenGate instance per installation of GoldenGate. A single **manager** process runs per GoldenGate instance/installation and it can support multiple **extract** and **replicat** processes.

Version Support Matrix

The following combination of Clusterware/GoldenGate/Database releases is supported.

Grid Infrastructure (GI)	GoldenGate	Database
11.2.0.3.+	11.2.1.+/11.1.1.+	10.2.0.5.0, 11.1.+, 11.2.+

*GI 11.2.0.3.+ release implies the same Oracle Automatic Storage Manager (ASM) release.

Agent Functions

- Manage GoldenGate application failover

- Start the GoldenGate instance manager process

- Monitor the GoldenGate instance extract process

- Monitor the GoldenGate instance replicat process

- Monitor the GoldenGate instance manager process

- Stop the GoldenGate instance and relevant dependencies

- Relocate the GoldenGate instance and relevant dependencies

- Clean the GoldenGate instance and relevant dependencies after a non-recoverable failure event
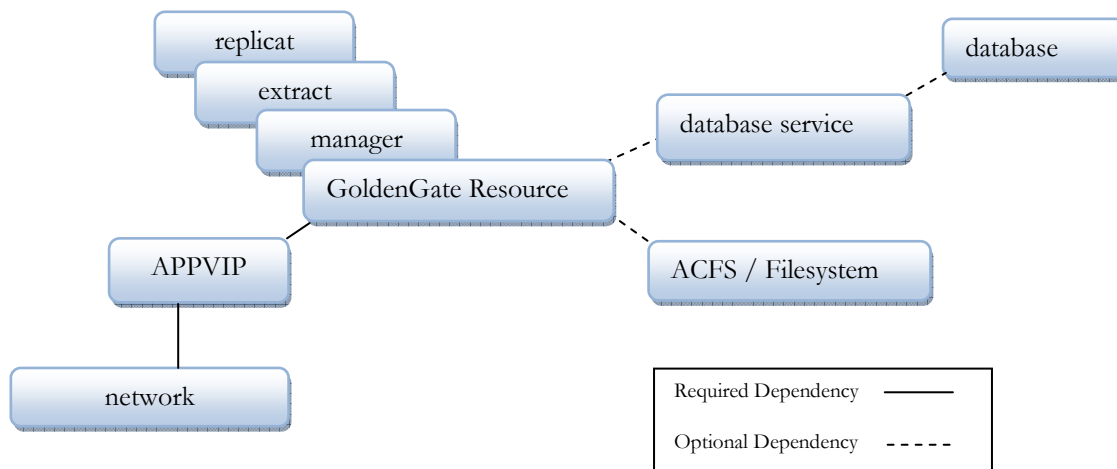
Resource Dependency Options

Figure 1

As noted in **Figure 1** above, the GoldenGate instance resource requires an application VIP (APPVIP in the example above) for manager connectivity by remote GoldenGate instances and for management and monitoring software components. An appvip requires a network resource net.net1.network, where net1 is the network number of the resource. Alternative network resources can be used if they exist. Other optional resource dependencies include ACFS or other supported file systems, and database connectivity dependencies either on the database service or the database directly.

State Definitions

ONLINE – The GoldenGate instance is online

OFFLINE – The GoldenGate instance is offline

INTERMEDIATE – The GoldenGate manager is online, however some or all extract and replicate processes are offline or have timed out when attempting to start

UNKNOWN - The state when Oracle Clusterware is unable to manage the resource and manual Oracle Clusterware intervention is required to stop it fix the root cause. Once corrected agctl start/stop commands should be used

The GoldenGate instance resource will transition to states : ONLINE, OFFLINE based on the operations and state of the *manager*. This resource will transition to state INTERMEDIATE if any of the specified *extract* or *replicat* processes are detected not to be running.

Resource Type Definitions

Oracle Clusterware uses resource types to organize and manage resources with common or similar attributes. The Oracle GoldenGate resource will have an application specific resource type, xag.goldengate.type, as defined internally by the Oracle Clusterware.

GoldenGate Agent AGCTL Syntax

Complete AGCTL usage for the goldengate resource is exposed using agctl –h. The following are common agctl operations for the GoldenGate application.

AGCTL command to register and configure a GoldenGate resource for a GoldenGate instance:

```
agctl [add | modify] goldengate instance_name
      --gg_home <GoldenGate installation directory>
      --serverpool <serverpool name> | --nodes <node1,node2,...>
      --instance_type <source|target|dual>
      --oracle_home <path>
      --db_services <associated database services>
      --databases <associated database resources>
      --environment_vars <name1=value1,name2=value2,...>
      --monitor_extracts <ext1,ext2,ext3,...>
      --monitor_replicats <rep1,rep2,rep3,...>
      --network <network_number>
      --ip <new VIP address>
      --vip_name <VIP resource name>
      --filesystems <acfs1,acfs2,...>
      --attribute <name1=value1,name2=value2,...>
```

Where the options for *agctl add* and *modify* commands for GoldenGate are:

instance_name	The name of the GoldenGate instance/resource. The instance_name must be unique and cannot be changed after the resource is registered. **Required**
gg_home	The GoldenGate installation directory. **Required**
serverpool	The name of the server pool in which this GoldenGate instance should be started
nodes	A list of nodes where the GoldenGate instance can be run
instance_type	Indicates whether this is a source or target instance. For bi-directional replication both source and target are required definitions. **Required**
oracle_home	The ORACLE_HOME location. **Required**
databases	The Oracle Database resources if GoldenGate instance has to be co-located with a Database instance and ORACLE_SID based connection is used by the extract/replicat processes. The ora.<database>.db resource name from `crsctl status resource –t` must be used for the databases definition.
db_services	The Database services (TNS), if the extract/replicat processes use service-based connection strings. The ora.<database>.<service_name>.svc name from `crsctl status resource –t` must be used for db_services definition.
monitor_extracts	An optional list of *extracts* to monitor. If any of the *extract* is not running, the GoldenGate instance resource state will transition to INTERMEDIATE.
monitor_replicats	An optional list of *replicats* to monitor. If any of the *replicat* is not running, the GoldenGate instance resource state will transition to INTERMEDIATE.
environment_vars	An optional list of environment variables to be passed when the GoldenGate instance is started/stopped/monitored. This is useful for setting up the environment when GoldenGate is operating on non-Oracle datastores.
network	The network number if a new VIP resource is to be created **Required**
ip	The VIP address if a new VIP resource is to be created **Required** if not using pre-created VIP
user	The name of the OS user. **Required** if not using pre-configured VIP
filesystems	The filesystem resource dependency name for the GoldenGate instance. The dependency name must be the resource name known to the Oracle Clusterware, for example ora.data.ggacfs.acfs, where **data** is the ASM diskgroup and **ggacfs** is the volume.

ORACLE

Note: The application VIP can be created in advance by the Grid Admin as root using the *appvipcfg* command. If the application VIP is created in advance only the *--vip_name* needs to be defined in the *agctl add* command and the command can be executed as the GoldenGate admin.. If you are creating the application VIP with *agctl*, the command must be run as root, the *--vip_name* should not be defined and the *--network*, *--ip* and *--user* flags must be defined.

The following are examples of common *agctl* commands for GoldenGate.

This command lists the current configuration of the GoldenGate instance. This command will also print a brief descriptive message listing the resource attributes:

```
agctl config goldengate <instance_name>
```

When the resource is first registered, it is enabled by default. Otherwise, to enable the GoldenGate resource so that Clusterware can start the GoldenGate instance:

```
agctl enable goldengate <instance_name>
```

The following command disables the GoldenGate resource after which Clusterware cannot start the instance. This command can be used if the GoldenGate instance should not be run for a period of time:

```
agctl disable goldengate <instance_name>
```

The relocate command relocates the running GoldenGate instance from one serverpool to another or from one node to another:

```
agctl relocate goldengate instance_name
      [--serverpool serverpool_name | --node node_name]
```

To delete the GoldenGate resource from the Clusterware registry:

```
agctl remove goldengate instance_name [--force]
```

This command starts the GoldenGate instance on the specified node or serverpool:

```
agctl start goldengate instance_name
      [--serverpool serverpool_name | --node node_name]
```

To get the current known state of the GoldenGate instance:

```
agctl status goldengate instance_name [-node node_name]
```

To stop the running GoldenGate instance:

```
agctl stop goldengate instance_name
```

ORACLE

Sample Configuration

The following is an example of a GoldenGate extract and replicat configuration on 2 4-node clusters where the Oracle Grid Infrastructure is installed and configured for both clusters.

A GoldenGate instance containing 2 extracts is registered with Clusterware using agctl, as root:

```
# agctl add goldengate gg_1 --gg_home /myacfs/ogg  --instance_type source \
--nodes ost144-31,ost144-32,ost144-33,ost144-34 \
--network 1 --ip 10.10.120.149 --user oracle  --group oinstall \
--filesystems ora.data.ggacfs.acfs  --databases ora.orcl1.db \
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 --monitor_extracts etest,ptest
```

Where:
   -the Goldengate instance is gg_1
   -the GoldenGate home is /myacfs/ogg
   -the GoldenGate instance type is *source* for extract
   -the nodes in the cluster participating in GoldenGate HA are
         ost144-31
         ost144-32
         ost144-33
         ost144-34
   -the network is the default ora.net1.network and the VIP is 10.10.120.149
   -the GoldenGate user is oracle in the group oinstall
   -the file system dependencies are on the acfs resource ora.data.ggacfs.acfs as defined by the grid admin
   -the database dependency is on the database resource ora.orcl1.db
   -the database home is /u01/app/oracle/product/11.2.0./dbhome_1
   -and the extracts to monitor are etest for primary change data capture and ptest for sending trail files to remote systems.

On the replication target cluster, the GoldenGate instance with a single replicat is registered using *agctl*, as root:

```
# agctl add goldengate gg_2 --gg_home /myacfs/ogg  --instance_type target \
--nodes ost144-100,ost144-101,ost144-102,ost144-103 \
--network 1 --ip 10.10.120.150 --user oracle  --group oinstall \
--filesystems ora.data.ggacfs.acfs  --databases ora.orcl2.db\
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 --monitor_replicats rtest
```

ORACLE

Where:
    -the GoldenGate instance is gg_2
    -the GoldenGate home is /myacfs/ogg
    -the GoldenGate instance type is *target* for replicat
    -the participating nodes are
        ost144-100
        ost144-101
        ost144-102
        ost144-103
    -the network is the default ora.net1.network and the VIP is 10.10.120.150
    -the GoldenGate user is oracle in the group oinstall
    -the file system dependencies are on the acfs resource ora.data.ggacfs.acfs as defined by the grid admin
    -the database dependency is on the database ora.orcl2.db
    -the database home is /u01/app/oracle/product/11.2.0./dbhome_1
    -and the replicat to monitor is *rtest*.

Oracle GoldenGate Agent Notes

1. Extracts with remote trail file writes must use the target VIP in the RMTHOST setting.
2. Control of the GoldenGate application will be through *agctl start/stop/status/relocate* commands. Do not stop the manager process via the GoldenGate command line interface GGSCI or via the GoldenGate GUI products as this will initiate a component failover.
3. If there are multiple GoldenGate instances corresponding to separate GoldenGate installation paths, a Clusterware GoldenGate resource needs to be registered for each clustered instance.
4. Oracle Clusterware will start, stop, monitor, restart, and failover the GoldenGate **manager** process. Clusterware will not start, stop or restart individual *extract* and *replicat* processes.
5. The Oracle Grid Infrastructure Bundled Agent for GoldenGate leverages the GoldenGate native AUTOSTART and AUTORESTART parameters in the *manager* parameter file for **extract** and **replicat** HA. Therefore, these parameters must be setup appropriately so that the **extract** and **replicat** processes are started and kept running. Clusterware can optionally monitor specified **extract**/**replicat** processes if configured to do so (see add command line syntax above) but their status will not initiate a failover in the current version.
6. After Goldengate integration with Oracle Clusterware, only the agctl tool should be used to start, stop and relocate the GoldenGate instance. Clusterware will implement pre-execution checks and ensure dependent infrastructure resources like databases, filesystems, etc, are running before the GoldenGate processes are started.

NOTE: *agctl* is run as root in the configuration example because the application vip (appvip) is created on the *agctl* command line and not pre-created using *appvipcfg*. When creating the appvip using *agctl*, you must define the user who will manage the VIP.

GoldenGate administrators must ensure the required **extracts** and **replicats** (or both) are configured for AUTOSTART and AUTORESTART. If the GoldenGate resource transitions to INTERMEDIATE state, Administrators need to take corrective actions for the failed **extract(s)/replicat(s)** and then restart them.

Oracle GoldenGate Integrated Capture is fully supported and requires the 11.2.0.3 database specific bundle patch for Integrated Extract 11.2.x (see Oracle Support MOS note, Doc ID 1411356.1).

For more detail on Oracle GoldenGate and bundled agent deployments please see the "Oracle GoldenGate Best Practices: Configuring Oracle GoldenGate with Oracle Grid Infrastructure Bundled Agents (XAG)" available on http://oracle.com/goto/clusterware.

ORACLE

Oracle Siebel Customer Relationship Management (CRM)

Oracle Siebel CRM is a suite of Customer Relationship Management applications of which the Siebel Gateway Server, Siebel Servers and the Siebel Web Server Extensions are critical server components for application availability. Each of these components maintain essential configuration, logging and application metadata. The Grid Infrastructure Bundle Agents for Oracle Siebel CRM provide high availability for the Siebel Gateway Server and configured Siebel Servers.

Version Support Matrix

The Oracle Grid Infrastructure for Siebel CRM supports the following combination of Oracle Grid Infrastructure / Oracle Siebel releases. The respective Database versions must be compatible with the Grid Infrastructure and Siebel versions.

Grid Infrastructure	Siebel
11.2.0.3.+	8.1+

Oracle Siebel CRM Supported Deployment Models

The Oracle Grid Infrastructure Bundled Agents for Siebel CRM High Availability support the following three deployment models depicted in **Figure 2**:

- Model 1: Standalone Singleton Siebel Gateway Server with failover
- Model 2: Standalone Siebel Server with failover
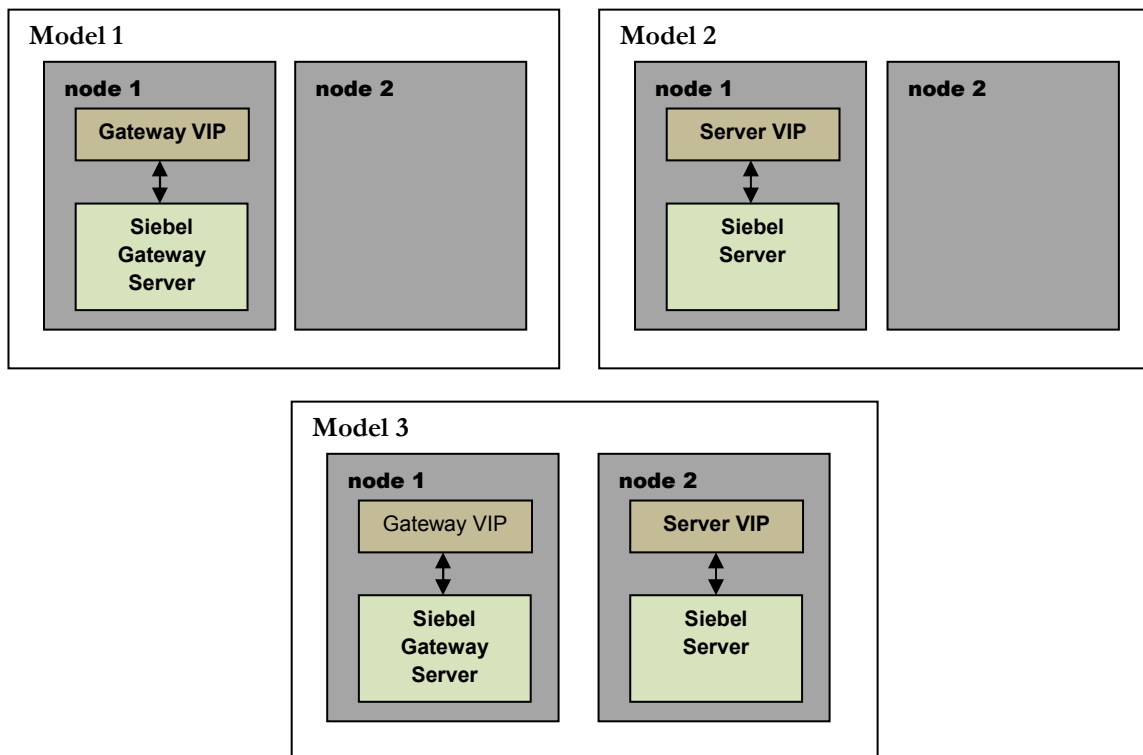- Model 3: Siebel Gateway Server with Siebel Server with failover

Figure 2

14

These deployment models will permit co-location of the Siebel Gateway Server with the Siebel Server on the same physical server. And, if configured, each Siebel Server must be bound to a unique application Siebel Server VIP to preserve the Siebel Server HA model.

Siebel Gateway Server

The Siebel Gateway Name Server is a required singleton server process which coordinates access of Siebel Enterprise Server and Siebel Servers to configuration data and connectivity information. When the Siebel Gateway Name Server is down, no new Siebel Server components can be started or added and server administration functions are severely limited.

Siebel Server

The Siebel Server is the middle-tier platform that supports both back-end and interactive processes for every Siebel client. These processes are components within the Siebel Server architecture and support functions such as:

- Mobile Web client synchronization
- Operation of business logic for Siebel Web clients, as well as connectivity and access to the Siebel Database and Siebel File System
- Integration with legacy or third-party data
- Automatic assignment of new accounts, opportunities, service requests, and other records
- Workflow management

The Siebel Server supports both multi-process and multithreaded components, and can operate components in background, batch, and interactive modes.

Agent Functions

- Manage the Siebel application failover for the Gateway and Siebel Servers

- Start the Siebel Gateway Server or Siebel Server and relevant dependencies

- Monitor the Siebel Gateway Server or Siebel Server and relevant dependencies

- Stop the Siebel Gateway Server or Siebel Server and relevant dependencies

- Relocate the Siebel Gateway Server or Siebel Server and relevant dependencies

- Clean the Siebel Gateway Server or Siebel Server and relevant dependencies after a non-recoverable

  failure event

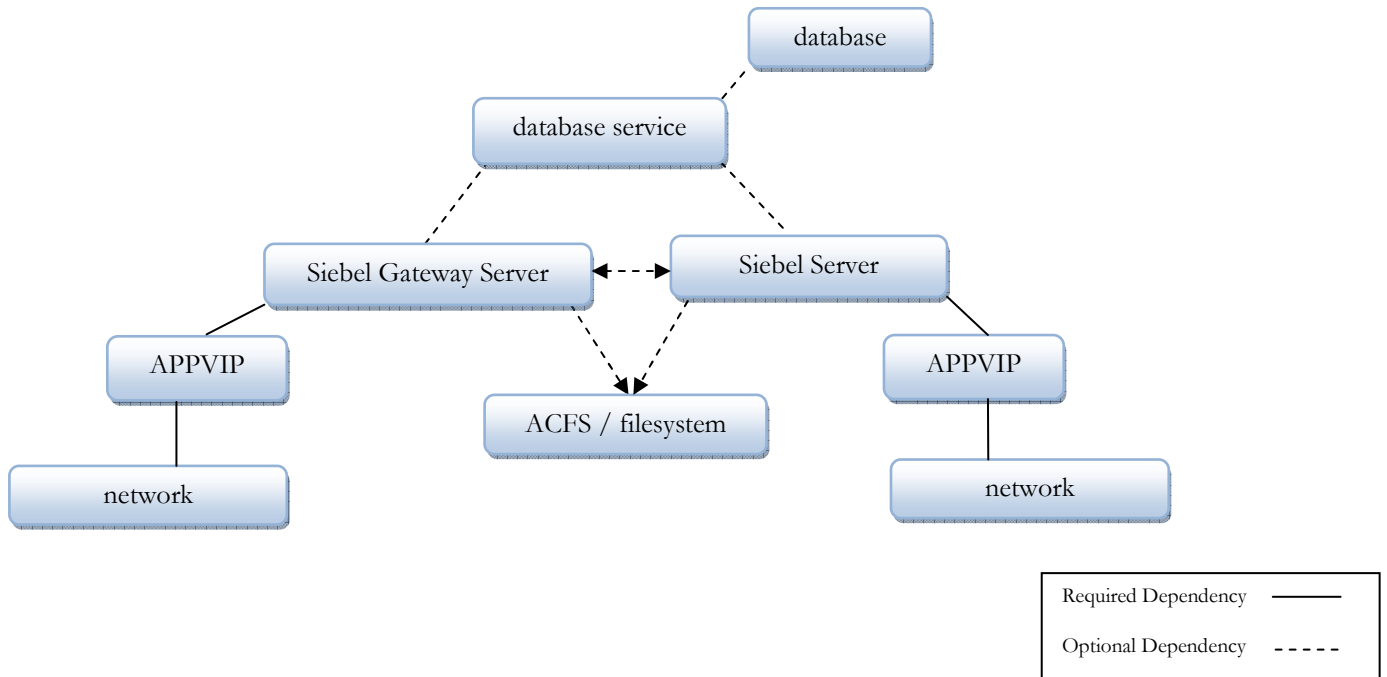**ORACLE**

Resource Dependency Options

Figure 3

The Siebel Gateway Server, as depicted in **Figure 3**, requires an application VIP (APPVIP) and supporting network resource. The Siebel Server and any additional configured Siebel Servers each require a unique application VIP to preserve individual Siebel Server HA. All application VIPs may share the same network resource (the net1 network resource by default) or they can configure unique network resource dependencies on unique network adapters. Optional start and stop dependencies can be configured for the Siebel Gateway Server and configured Siebel Servers where starting one of these resources will start the other. In addition, start and stop dependencies may be configured for a supporting shared file system such as ACFS. The agent supports start and stop dependencies configured for either the database or configured services depending on how the application connects to the database.

State Definitions

ONLINE – The Siebel Gateway Server or Siebel Server is online

OFFLINE – The Siebel Gateway Server or Siebel Server is offline

INTERMEDIATE – The Siebel Gateway Server or Siebel Server is online, however inaccessible

UNKNOWN - The state when Oracle Clusterware is unable to manage the resource and manual Oracle Clusterware intervention is required to stop it fix the root cause. Once corrected agctl start/stop commands should be used

ORACLE®

Resource Type Definitions

Oracle Clusterware uses resource types to organize and manage resources with common or similar attributes. The Oracle Siebel CRM resource will have an application specific resource type, xag.siebel.type, as defined internally by the Oracle Clusterware.

Seibel Agent AGCTL Syntax

The *agctl add* command adds Siebel Gateway Server or Siebel Server instance to Oracle Clusterware.   The Siebel user can add the Siebel Gateway Server or Siebel Server and all of its options when adding the resources with pre-created VIP., The options *network_number, ip_address,* must be executed as root in order to create the VIP and the user and group options must be defined.
The following are common *agctl* commands for the Siebel agent.

This command registers the Siebel Gateway Server to the Oracle Clusterware for agent control:

```
agctl add siebel_gateway <instance_name>
        --siebel_home <siebel_home>
        --oracle_home <oracle_home>
        [--oracle_client_home <oracle_client_home>]
        [ { --serverpool <serverpool> | --nodes <node1[,…]> } ]
        [ { --vip_resource <vip_resource> |
            --network=<network_number> --ip=<ip_address>
            --user=<user> --group=<group> } ]
        [ --attribute "<attribute_name=value,attribute_name=value,…>" ]
        [ --filesystems <filesystem1[,…]> ]
        [ --databases <database1[,…]> ]
        [ --db_services <db_service1[,…]> ]
```

This command registers the Siebel Server to the Oracle Clusterware for agent control:

```
agctl add siebel_server <instance_name>
        --siebel_home <siebel_home>
        --oracle_home <oracle_home>
        --enterprise_name <enterprise_name>
        --server_name < server_name>
        [--oracle_client_home <oracle_client_home>]
        [ --gateway_resource <gateway_resource> ]
        [ { --serverpool <serverpool> | --nodes <node1[,…]> } ]
        [ { --vip_resource <vip_resource> |
            --network=<network_number> --ip=<ip_address>
            --user=<user> --group=<group> } ]
        [ --attribute "<attribute_name=value,attribute_name=value,…>" ]
        [ --filesystems <filesystem1[,…]> ]
        [ --databases <database1[,…]> ]
        [ --db_services <db_service1[,…]> ]
```

Where the options for *agctl add* and *modify* commands for Siebel are:

instance_name	The name of the Siebel Gateway instance/resource. The *instance_name* must be unique and cannot be changed after the resource is registered. **Required**
siebel_home	The Siebel Gateway installation directory. **Required**

ORACLE®

oracle_client_home	The location of the Oracle 32-bit Client Libraries for client connectivity to the database
enterprise_name	The Enterprise Server Name **Required**
serverpool	The name of the server pool in which this Siebel instance/resource should be started
nodes	A list of nodes where the Siebel Gateway instance/resource can be run
environment_vars	An optional list of environment variables to be passed when the Siebel Gateway or Server instance is started/stopped/monitored.
network	The network number if a new VIP resource is to be created **Required**
ip	The VIP address if a new VIP resource is to be created **Required** if not using pre-created VIP
user	The name of the OS Siebel user. **Required** if not using pre-configured VIP
group	The name of the Siebel group to which the Siebel user belongs.
attribute	Sets/overrides default values for standard Clusterware attributes of the GoldenGate resource (e.g. CHECK_INTERVAL, AUTO_START)
filesystems	List of file system resource dependencies for this Siebel instance. The filesystem resource name must be the resource name known to the Oracle Clusterware, for example ora.<diskgroup>.<volume>.acfs, for ACFS filesystems.
databases	Oracle Database resource dependency for the Siebel instance. The ora.<database>.db resource name from `crsctl status resource –t` must be used for the **databases** definition.
db_services	Database services dependency name. The ora.<database_name>.<service_name>.svc resource name from `crsctl stat resource –t` must be used for the **db_services** definition.

The following are further examples of common *agctl* commands for Siebel:

To list the current configuration of the requested Siebel instance and print a brief descriptive message listing the resource attributes:

```
agctl config { siebel_gateway | siebel_server } [ <instance_name> ]
```

This command enables the requested Siebel resource so that Clusterware can start the instance. When the resource is first registered, it is enabled by default:

```
agctl enable { siebel_gateway | siebel_server } <instance_name> [ --node
                <node_name> ]
```

ORACLE

This command disables the requested Siebel instance after which Clusterware cannot start the instance. This command can be used if the required Siebel instance should not be run for a period of time:

```
agctl disable { siebel_gateway | siebel_server } <instance_name> [ --node
<node_name> ]
```

This command modifies the configuration of the Siebel Gateway instance. Options are same as the *agctl add siebel_gateway* command:

```
agctl modify siebel_gateway <instance_name>
          [ --siebel_home <siebel_home>]
          [ --oracle_home <oracle_home>]
          [ --oracle_client_home <oracle_client_home>]
          [ { --serverpool <serverpool> | --nodes <node1[,…]> } ]
          [ { --vip_name <vip_resource> |
              --network=<network_number> --ip=<ip_address>
              --user=<user> --group=<group> } ]
          [ --attribute "<attribute_name=value,attribute_name=value,…>" ]
          [ --filesystems <filesystem1[,…]> ]
          [ --databases <database1[,…]> ]
          [ --db_services <db_service1[,…]> ]
```

This command modifies the configuration of the Siebel Server instance. Options are same as the *agctl add siebel_server* command:

```
agctl modify siebel_server <instance_name>
          [ --siebel_home <siebel_home>]
          [--enterprise_name <enterprise_name>]
          [--server_name < server_name>]
          [ --oracle_home <oracle_home>]
          [--oracle_client_home <oracle_client_home>]
          [ --gateway_resource <gateway_resource> ]
          [ { --serverpool <serverpool> | --nodes <node1[,…]> } ]
          [ { --vip_name <vip_resource> |
              --network=<network_number> --ip=<ip_address>
              --user=<user> --group=<group> } ]
          [ --attribute "<attribute_name=value,attribute_name=value,…>"]
          [ --filesystems <filesystem1[,…]> ]
          [ --databases <database1[,…]> ]
          [ --db_services <db_service1[,…]> ]
```

To relocate the running Siebel instance (*siebel_gateway* or *siebel_server*) from one serverpool to another or from one node to another:

```
agctl relocate { siebel_gateway | siebel_server } <instance_name>
                 [ --server <serverpool> | --node <node_name> ]
```

This command deletes the requested Siebel resource. The force option will be required if there are running dependencies on the resource:

```
agctl remove { siebel_gateway | siebel_server } <instance_name> [ --force ]
```

19

ORACLE®

To start the requested Siebel instance on the specified node or serverpool:

```
agctl start { siebel_gateway | siebel_server } <instance_name>
                  [ --server <serverpool> | --node <node_name> ]
```

This command returns the current known state of the requested Siebel resource:

```
agctl status { Siebel_gateway | Siebel_server } <instance_name> [-node
<node name> ]
```

This command stops the running Siebel resource where *instance_name* is either the Siebel Gateway Server or the Siebel Server instance. The force flag may be required if there are running dependencies on the requested resource:

```
agctl stop { siebel_gateway | siebel_server } <instance_name> [ --force ]
```

Sample Configurations

The following is an example of a Siebel Gateway Server HA configuration on a 2-node cluster where the Oracle Grid Infrastructure is installed and configured prior to the Siebel Gateway Server and the VIP and the VIP hostname is used as the Siebel Gateway address rather than the physical hostname.

To create the Siebel Gateway Server execute as root:

```
# agctl add siebel_gateway sieb_gtwy --siebel_home /myacfs/gtwy  \
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 \
--oracle_client_home /u01/app/oracle/product/11.2.0/client
--nodes ost144-41,ost144-42 \
--network 1 --ip 10.10.121.141 --user siebel  --group oinstall \
--filesystems ora.data.sieb.acfs  --databases ora.orcl4.db
```

Where:
    -the Siebel Gateway Server instance is sieb_gtwy
    -the Siebel Gateway Server home is /myacfs/gtwy
    -the Oracle Home is /u01/app/oracle/product/11.2.0/dbhome_1
    -the Oracle Client Library home is /u01/app/oracle/product/11.2.0/client
    -the nodes in the cluster participating in Siebel Gateway Server HA are
        ost144-41
        ost144-42
    -the network used is the default ora.net1.network and the VIP is 10.10.121.141
    -the Siebel user is siebel in the group oinstall
    -the file system dependency is on the resource ora.data.sieb.acfs as defined by the grid admin
    -the database dependency is on the database resource ora.orcl4.db

ORACLE

To create the Siebel Server HA for a single server, execute as root:

```
# agctl add siebel_server  sieb_srvr1  --siebel_home /myacfs/sieb_srvr1  \
--oracle_home /u01/app/oracle/product/11.2.0/dbhome_1 \
--client_home /u01/app/oracle/product/11.2.0/client \
--nodes ost144-41,ost144-42 \
--network 1 --ip  10.10.121.142 --user  siebel  --group  oinstall \
--filesystems  ora.data.sieb.acfs  --databases  ora.orcl4.db
```

Where:
   -the Siebel Server instance is sieb_srvr1
   -the Siebel Server home is /myacfs/sieb_srvr1
   -the Oracle Home is /u01/app/oracle/product/11.2.0/dbhome_1
   -the Enterprise Server name is SIEB
   -the Oracle Client Library home is /u01/app/oracle/product/11.2.0/client
   -the nodes in the cluster participating in Siebel   Server HA are
        ost144-41
        ost144-42
   -the network used is the default ora.net1.network and the VIP is 10.10.121.142
   -the Siebel user is siebel in the group oinstall
   -the file system dependency is on the resource ora.data.sieb.acfs as defined by the grid admin
   -the database dependency is on the database ora.orcl4.db

Siebel CRM Agent Notes

There are three deployment models that influence the Siebel Server to Siebel Gateway communication. For new installs, configure the application VIP and use that as the Siebel Gateway Name Server host and IP for the Siebel Gateway and Siebel Server installations. If you are installing the Oracle Grid Infrastructure Bundled Agents in a pre-existing Siebel environment you will need to execute the following steps:

1. Create the Siebel Gateway VIP on the *agctl* command line or using *appvipcfg*
2. Stop the Siebel Gateway Server
3. Execute the `agctl add siebel_gateway …` per instructions above where *--vip_name or --network, --ip, --user* are defined.
4. Start the Siebel Gateway instance using `agctl start siebel_gateway <siebel_instance_name>`

When step 3 is executed, the agctl utility will modify the Siebel Servers dependent on that Gateway Server by executing, internal to the `agctl add siebel_gateway`, the following steps:

a) source siebenv.sh
b) Stop Siebel Server service (stop_server ALL)
c) Remove *.osdf in the $SIEBEL_ROOT/sys
d) Backup the SVC file with the format svc.siebsrvr.siebel:<Siebel Server>
e) Remove the svc.* files in the $SIEBEL_ROOT/sys
f) Recreate SVC file by executing siebctl command below to bind the VIP to the Siebel Servers:

```
$ siebctl -S siebsrvr -i $enterprise_name:$server_name -a -g "-g $ip:$gateway_port -e $enterprise_name
-s $server_name" (executed by agctl)
```

ORACLE

where:

enterprise_name = The name of the Siebel Enterprise

siebsrvrname = The name of the Siebel Server

ip:gtwyport = The agent VIP assigned to the Siebel Gateway Server:port number

server_name=The name of the Siebel Server

This ensures that the clustered Siebel Servers associated with the defined Siebel Gateway are updated automatically. Whenever the *agctl add* or *modify* Siebel_gateway command is run and the *--vip_name* or *--network, --ip, --user* have been modified, agctl will run the above steps a-f automatically so that the Siebel Servers recognize the Siebel Gateway Server's new network identity. The changes take effect when Siebel Servers are started. For any Siebel Servers added subsequent to adding or modifying the Siebel Gateway address, the user will need to execute the steps a-f noted above manually.

Siebel Server Changes to Enable Siebel Enterprise Server Communication

When the Siebel Server is clustered you need to ensure that other non-clustered Siebel Servers in the Siebel Enterprise can communicate with the clustered Siebel Server. The *agctl* command will not modify non-clustered Siebel Servers. Please follow the next set of steps to manually change the hostname of the clustered Siebel Server to the Cluster Network Name:

1. Using the srvrmgr command in order to bind to the Siebel Server cluster hostname to the Siebel Server change the configuration of Siebel server using srvrmgr command below:
    a. If the Siebel version is 7.7 or later, change the ServerHostAddress parameter to the IP address of the Siebel server's logical host name resource using the following syntax:

```
$ srvrmgr> change param ServerHostAddress=$ip for server $server_name
```

    b. If the Siebel version is earlier than 7.7, change the Host parameter to the logical hostname for the Siebel server using the following syntax:

```
$ srvrmgr:hasiebel> change param Host=$ip for server $server_name
```

NOTE: The above can either be IP or Hostname as long as they map to the cluster's logical IP/logical Hostname. The changes take effect when Siebel Servers are started.

Control of the Siebel components must be through *agctl start/stop/status/relocate* commands. Do not stop the Siebel Gateway or Siebel Server processes via the Siebel command line interface or via any other means as this will initiate a component failover.

ORACLE

Apache HTTP and Tomcat Server

The Oracle Grid Infrastructure Bundled Agents manage Apache Tomcat and HTTP Servers for high availability and scalability. The bundled agents provide HA for these Apache components which is unavailable by default. The ability to configure and control multiple Apache resources provides a highly scalable framework which administrators can manage through a simple, intuitive interface.

Apache Tomcat

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the Java Community Process. Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation (see tomcat.apache.org)

Apache HTTP Server

Apache HTTP Server is a public-domain open source Web server (see httpd.apache.org).

Version Support Matrix

The Oracle Grid Infrastructure for Apache Tomcat and HTTP Servers supports the following combination of Oracle Grid Infrastructure / Apache releases.

Supported Apache Versions

Grid Infrastructure (GI)	HTTP Server	Tomcat Server
11.2.0.3+	2.2.x, 2.0.x	7.0.x, 6.0.x, 5.5.x

Apache Tomcat and Web Server Supported Deployment Models

Oracle Grid Infrastructure bundled agents for Apache supports four deployment models depicted in **Figure 4** below. The first model (model 1) is a standalone Apache HTTP Server. The second model (model 2) is a standalone Apache Tomcat Server without the HTTP Server. The third model (model 3) is the Apache HTTP Server co-located with the Apache Tomcat Server on a single node in the cluster. The fourth model (model 4) is the HTTP Server and the Apache Tomcat Server deployed on different nodes in the cluster.
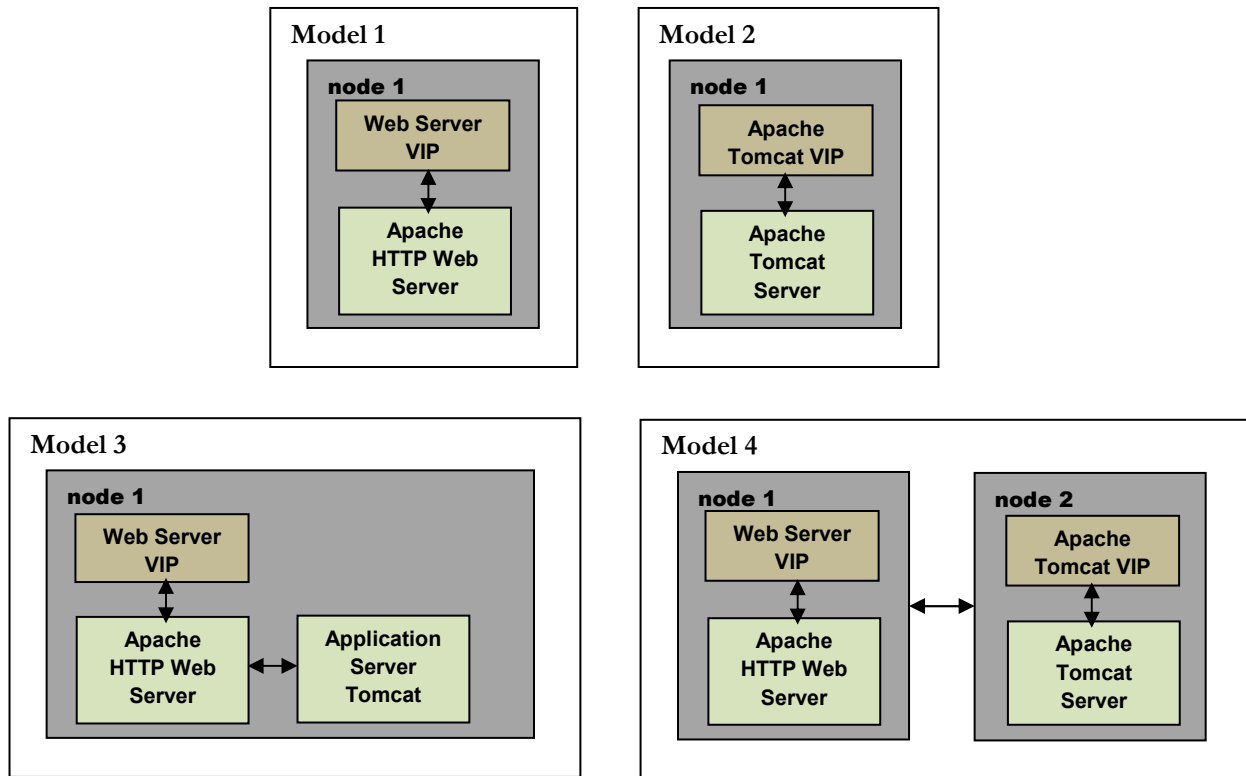
Figure 4

Apache Agent Functions

- Manage the Apache Tomcat and and/or Apache HTTP Server failover

- Manage the Apache Tomcat and/or Apache HTTP Server scalability

- Start the Apache Tomcat and/or Apache HTTP Server and relevant dependencies

- Monitor the Apache Tomcat and/or Apache HTTP Server and relevant dependencies

- Stop the Apache Tomcat and/or Apache HTTP Server and relevant dependencies

- Relocate the Apache Tomcat and/or Apache HTTP Server and relevant dependencies

- Clean the Apache Tomcat and/or Apache HTTP Server and relevant dependencies after a non-recoverable failure event

ORACLE

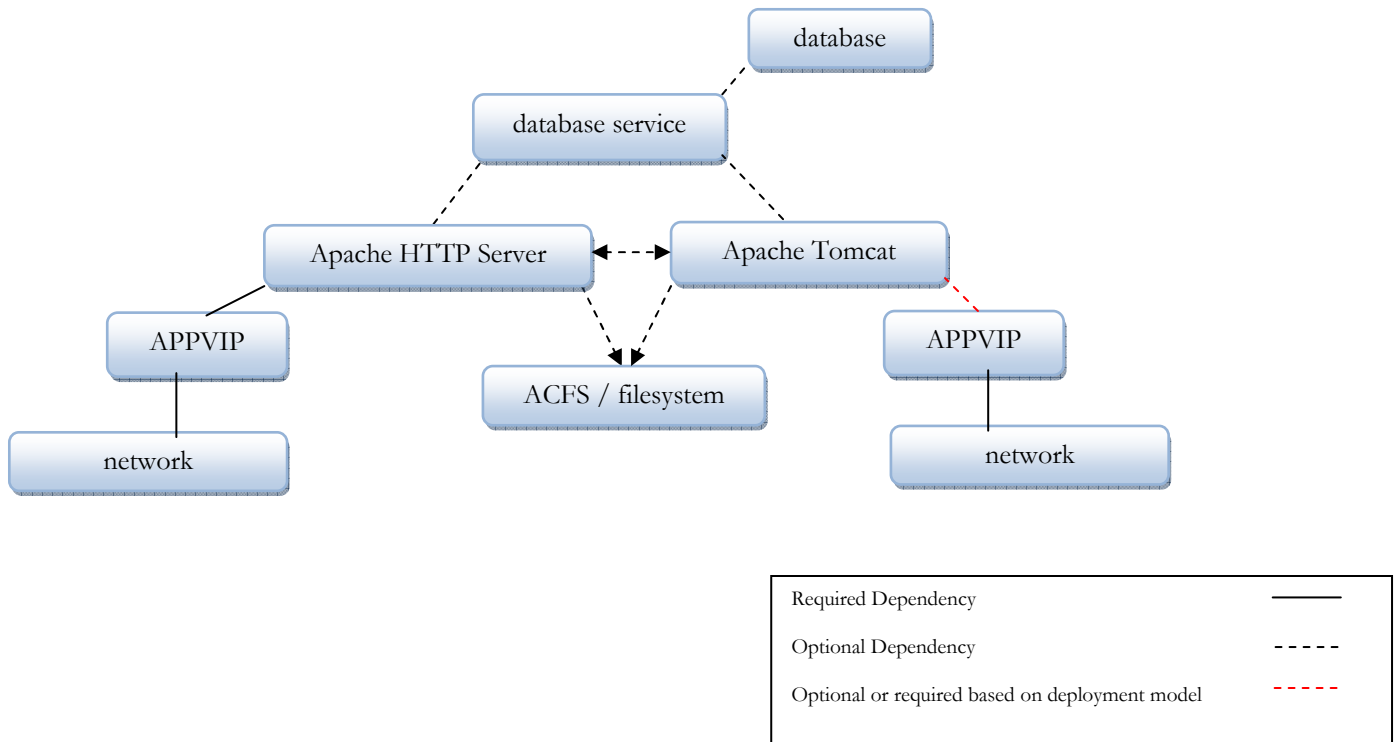**Resource Dependency Options**

Figure 5

The chosen deployment model will dictate required dependencies. **Figure 5** illustrates that Models 1-3 have a single network resource stack defined for either the Apache HTTP Server or Tomcat Server. If the Apache HTTP Server and the Tomcat Server are co-located on a single node, the Apache HTTP Server will have a dependency on the network stack and the Apache Tomcat Server will have a dependency on the Apache HTTP Server. In deployment Model 4, the Apache HTTP Server and Apache Tomcat are running on separate nodes in the cluster. In this model, they would both need network stack (network, VIP) dependencies. How the Apache Tomcat Server is deployed will determine whether there is a required dependency on the network stack (see **red** dependency link above). Supplementary dependencies may be defined between the HTTP server and Tomcat even though they are running on separate nodes.

Apache HTTP Server Dependency Options

If the *--databases*, *--db_services,* and/or *--filesystems* option are specified, the Apache Web Server will have a start dependency on databases, database service, and file system resources. This dependency requires that the database, database service, and file system resources must be running before the Apache HTTP Server can start. The Apache HTTP Server can start on any node in the cluster as long as the database, database service, and file system resources are running. The agent dependency model guarantees that these resources will be started in the correct order, if configured. The resource dependency definition must be the resource names as exposed in `crsctl stat res –t`. For example database, database service and filesystem (for ACFS) would be represented as ora.<database_name>.db, ora.<database_name>.<service_name>.svc and ora.<diskgroup>.<volume>.acfs respectively.

Apache Tomcat Dependency Options

If the *--webserver*, *--databases*, *--db_services,* and/or *--filesystems* option is specified, the Apache Tomcat Server will have a start dependency on the Apache HTTP Server, database, database service, and file system resources. This dependency means that the Apache HTTP Server, database, database service, and file system resources must be running before the Apache Tomcat Server can start. The Apache Tomcat Server can start on any defined node in the cluster as long as the Apache HTTP Server, database, or database service are running on any node in the cluster. The agent dependency model guarantees these resources will be started in the correct order, if the dependency is configured.

Apache Tomcat will also have a start dependency on the Apache HTTP Server, database, database service, and/or file system. This dependency means that Apache Tomcat Server will automatically start whenever the Apache HTTP Server, database, database Service, and file system resources are started. As with the Apache HTTP Server, the resource dependency definition must be the resource names as exposed in `crsctl stat res –t`. For example database, database service and filesystem (for ACFS) would be represented as ora.<database_name>.db, ora.<database_name>.<service_name>.svc and ora.<diskgroup>.<volume>.acfs respectively.

Apache Agent State Definitions

ONLINE – The Apache HTTP Server or Apache Tomcat is online

OFFLINE – The Siebel Apache HTTP Server or Apache Tomcat is offline

INTERMEDIATE – The Apache HTTP Server or Apache Tomcat is online, however inaccessible

UNKNOWN - The state when Oracle Clusterware is unable to manage the resource and manual Oracle Clusterware intervention is required to stop it fix the root cause. Once corrected agctl start/stop commands should be used

Resource Type Definitions

Oracle Clusterware uses resource types to organize and manage resources with common or similar attributes. The Apache HTTP Server and Apache Tomcat resources will have application specific resource type, xag.httpd.type and xag.tomcat.type   as defined and used internally by the Oracle Clusterware.

Apache Agent AGCTL Syntax

Complete AGCTL usage for the Apache resources is exposed using agctl –h. The following are common agctl operations for the Apache HTTP Server and Apache Tomcat.

AGCTL command to register and configure an Apache HTTP Server resource:

```
agctl add apache_webserver <instance_name>
        --apache_home <webserver_home>
        [--config_file <configuration_file>]
        [{--serverpool <serverpool> | --nodes <node1[,…]> }]
        [{--vip_name <vip_resource> |
            --network=<network_number> --ip=<ip_address>
            --user=<user> --group=<group> }]
        [--attribute "<attribute_name=value,attribute_name=value,…>"]
        [--filesystems <filesystem1[,…]>]
        [--databases <database1[,…]>]
        [--db_services <db_service1[,…]>]
```

ORACLE

Similarly, to register and configure Apache Tomcat to execute under agent control:

```
agctl add apache_tomcat <instance_name>
        --catalina_home <catalina_home>
        --java_home <java_home>
        [--catalina_base <catalina_base>]
        [--jre_home <jre_home>]
        [--webserver <webserver_instance> ]
        [{--serverpool <serverpool> | --nodes <node1,node2[,…]>}]
        [{ --vip_name <vip_resource> |
            --network=<network_number> --ip=<ip_address>
            --user=<user> --group=<group>}]
        [--attribute "<attribute_name=value,attribute_name=value,…>"]
        [--filesystems <filesystem1[,…]>]
        [--databases <database1[,…]>]
        [--db_services <db_service1[,…]>]
```

Where the options for *agctl add* and *modify* commands for Apache HTTP and Tomcat Server are:

instance_name	Name of the Apache HTTP server or Tomcat server instance/resource. The instance_name must be unique and cannot be changed after the resource is registered. **Required**
apache_home	Web Server installed directory. **Required**
config_file	Name of the configuration file. It's used for multiple instances.
catalina_home	Application Server installed directory **Required**
catalina_base	Application Server configuration directory. It's used for multiple instances.
java_home	JAVA Development installed directory for Application Server. **Required**
jre_home	JRE Development installed directory for Application Server. Defaults to java_home if it's empty.
webserver	Name of the Web Server instance. This option is only used for adding Application Server(s) for model 3 & 4 only. Otherwise, it returns an error.
serverpool	Name of the server pool in which this Siebel instance/resource should be started
environment_vars	An optional list of environment variables to be passed when the Apache HTTP or Tomcat instances are started/stopped/monitored. This is useful for setting up the environment when GoldenGate is operating on non-Oracle datastores.
vip_name	The virtual IP address of the node or the application VIP resource that was pre-created by Oracle Clusterware Application VIP utility (*appvipcfg*).
network	The network number if a new VIP resource is to be created **Required**

ORACLE®

ip	The VIP address if a new VIP resource is to be created **Required** if not using pre-created VIP
user	The name of the OS Apache user. **Required** if not using pre-configured VIP
group	The name of the Apache OS group to which the Apache user belongs.
attribute	Set/override default values for standard Clusterware attributes of the Apache resource (e.g. CHECK_INTERVAL, AUTO_START)
filesystems	List of file system resource dependencies for this Apache instance. The filesystem resource name must be the resource name known to the Oracle Clusterware, for example ora.<diskgroup>.<volume>.acfs, for ACFS filesystems.
databases	Oracle Database resource Apache instance dependency. The database resource name must be the resource name known to the Oracle Clusterware, for example ora.<database_name>.db.
db_services	Database services resource Apache dependency. The db_services resource name must be the resource name known to the Oracle Clusterware, for example ora.<database_name>.<service_name>.svc.

The following are other common Apache *agctl* commands.

This command lists the current configuration of the requested Apache instance. This command will also print a brief descriptive message listing the resource attributes:

```
agctl config { apache_webserver | apache_tomcat } <instance_name>
```

This command enables the requested Apache resource so that agent can start the instance. When the resource is first registered, it is enabled by default:

```
agctl enable { apache_webserver | apache_tomcat } <instance_name> [--node
                <node_name>]
```

This command disables the requested Apache instance after which agent cannot start the instance. This command can be used if the required Apache instance should not be run for a period of time:

```
agctl disable { apache_webserver | apache_tomcat } <instance_name> [--node
                <node_name>]
```

This command modifies the configuration of the Apache HTTP Server instance. Options are same as the *agctl add apache_webserver* command:

```
agctl modify apache_webserver <instance_name>
          [--apache_home <webserver_home>]
          [--config_file <configuration_file>]
          [{ --serverpool <serverpool> | --nodes <node1[,…]>}]
          [{ --vip_name <vip_resource> |
              --network=<network_number> --ip=<ip_address>
              --user=<user> --group=<group>}]
          [--attribute "<attribute_name=value,attribute_name=value,…>"]
          [--filesystems <filesystem1[,…]>]
          [--databases <database1[,…]>]
          [--db_services <db_service1[,…]>]
```

This command modifies the configuration of the Apache Tomcat Server instance. Options are same as the *agctl add apache_tomcat* command:

```
agctl modify apache_tomcat <instance_name>
          [--catalina_home <catalina_home>]
          [--catalina_base <catalina_base>]
          [--java_home <java_home>]
          [--jre_home <jre_home>]
          [--webserver <webserver_instance>]
          [{ --serverpool <serverpool> | --nodes <node1[,…]>}]
          [{ --vip_name <vip_resource> |
              --network=<network_number> --ip=<ip_address>
              --user=<user> --group=<group>}]
          [--attribute "<attribute_name=value,attribute_name=value,…>"]
          [--filesystems <filesystem1[,…]>]
          [--databases <database1[,…]>]
          [--db_services <db_service1[,…]>]
```

This command relocates the running Apache HTTP or Tomcat Server instance from one serverpool to another or from one node to another:

```
agctl relocate { apache_webserver | apache_tomcat } <instance_name>
                   [--server <serverpool> | --node <node_name>]
```

This command deletes the requested Apache resource from the Clusterware registry. The force option will be required if there are running dependencies on the resource. This command does not remove the apache_webserver or application server defined homes:

```
agctl remove { apache_webserver | apache_tomcat } <instance_name> [ --force ]
```

This command starts the requested Apache instance on the specified node or serverpool:

```
agctl start { webserver | tomcat } <instance_name>
                   [--server <serverpool> | --node <node_name>]
```

This command returns the current known state of the requested Apache resource:

```
agctl status [ webserver | tomcat ] <instance_name> [--node <node_name>]
```

29

ORACLE

This command stops the running Apache resource where instance_name is either the Apache webserver or the Apache tomcat instance. The force flag may be required if there are running dependencies on the requested resource:

```
agctl stop { webserver | tomcat } <instance_name> [ --force ]
```

Apache Agent Sample Configuration

The following is an example of an Apache HA configuration on a 2-node cluster where the Apache HTTP Server is configured to run in a defined serverpool and the database connection if via TNS services:

```
# agctl add apache_webserver webserver01 --apache_home /apache2.2 \
--configuration_file /apache2.2/myconf.xml --network 1 --ip 192.168.56.201 \
--user apache --group oinstall --filesystems ora.data.apache_ws.acfs \
--server_pool ora.pool1 --db_services ora.orcl5.dbserv1.svc
```

Where:
   -the Apache HTTP Server instance is webserver01
   -the Apache HTTP Server home is /apache2.2
   -the Apache configuration file location is /apache2.2/myconf.xml
   -the network used is the default ora.net1.network and the VIP is 192.168.56.201
   -the Apache user is apache in the group oinstall
   -the file system dependency is on the resource ora.data.apache_ws.acfs as defined by the Grid Admin
   -the serverpool is ora.pool1 (it is expected that the Grid Admin creates the serverpool in advance)
   -the database dependency is on the database TNS service ora.orcl5.dbserv1.svc

The following command adds the Apache Tomcat Server and restricts the application to run on specific nodes in the cluster with a dependency on the Apache HTTP Server defined above where the Apache HTTP Server and the Apache Tomcat Server are not co-located. Note that in the previous command, the Apache HTTP Server VIP was created as root on the *agctl* command line with the defined *--user* option. Below, the Apache Tomcat VIP has been pre-created. This command is run as the Apache Tomcat Admin:

```
$ agctl add apache_tomcat tomcat01  --catalina_home /tomcat7.8 --java_home \
/usr --catalina_base /tomcat7_base --webserver webserver01 --vip_name myvip \
--nodes node2,node4 --filesystems ora.data.apache_tc.acfs
```

Where:
   -the Apache Tomcat Server instance is tomcat01
   -the Apache Tomcat Server Catalina home is /tomcat7.8
   -the Jave home is /usr
   -the Apache Tomcat Server Catalina base is /tomcat7_base
   -the webserver dependency is on Apache HTTP Server instance webserver01
   -the APPVIP is pre-configured and is myvip

ORACLE

-the Apache Tomcat Server instance is restricted to run on nodes node2 and node4
-the file system dependency is on the resource ora.data.apache_tc.acfs as defined by the Grid Admin

Apache Agent Notes

For Apache HTTP Server:

1. During *agctl add* or *modify* operations where *--vip_name* option is added or modified or *--network, --ip, --user* options are added or modified, *agctl* will modify the *Listen* entry in $APACHE_HOME/conf/httpd.conf *w*ith the the new VIP address.
2. If $APACHE_HOME location is on a shared file system, *agctl* will modify the configuration file for all nodes.
3. If $APACHE_HOME location is not on a shared file system, a warning will be issued. The Apache HTTP admin is responsible for updating the *Listen* entry in $APACHE_HOME/conf/httpd.conf file on every node to use the VIP address.

   For example:

   > Listen <host>:80
   > Will be modified to:
   > Listen <ip address>:80
   > Where <ip address> is the IP address of the application VIP

   NOTE: Prior to any modification of the *httpd.conf* file, the original file will will be saved as *httpd.conf.xag*. If *vip_name* is modified, the *httpd.conf* file will also be modified to reflect the changes.
   If *vip_name* is removed or if Web Server instance is removed from Oracle Clusterware, the *httpd.conf* file will be restored from the backup file *httpd.conf.xag*. The backup file will be deleted if restore is successful.

For Apache Tomcat:

1. During *agctl add* or *modify* operations where the *--vip_name* option is added or modified or *--network, --ip,* or *–user* options are added or modified, *agctl* will internally modify the *<Host name= …>* entry in $CATALINA_BASE/conf/server.xml file with the the new VIP address.
2. If $CATALINA_BASE location is on a shared file system, *agctl* will automatically back up the server.xml file to server.xml.xag and modify the configuration file for all nodes.
3. If $ CATALINA _BASE location is not on a shared file system, a warning will be issued. The Apache Tomcat admin is responsible for updating the *Listen* entry in $ CATALINA_BASE/conf/server.xml file on every node to use the new VIP address.

   For example:

   > Host name=<host>:80
   > must be modified to:
   > Host name=<ip address>:80
   > Where <ip address> is the IP address of the application VIP

NOTE: Prior to any modification of the *server.xml* file by *agctl*, the *agctl* the original file will be saved as *server.xml.xag*. If the VIP resource is modified by changing*--vip_name* or, *--network*, *--ip* , *--user*, the *server.xml* file must also be modified to reflect the changes. If the VIP resource is removed my removing *--vip_name* or *--network, --ip, --user* or if the Apache Tomcat Server instance is removed using *agctl*, the server.xml the will be automatically restored from the backup file server.xml.xag. The backup file will be deleted if the restore is successful.

Control of the Apache components must be through *agctl start/stop/status/relocate* commands. Do not stop the Apache HTTP or Tomcat Server processes via the Apache command line interface or via any other means as this will initiate the component failover.

Diagnostics and Troubleshooting

Diagnostic logs exist in each tier of the environment and should be evaluated to correctly identify the fault and root cause. The Oracle Clusterware may fail an application resource when the underlying application has been modified outside of AGCTL control. Under such circumstances, the bundled agent may not have the modified application configuration and the start/stop or check actions would fail. The application configuration must be consistent with the AGCTL configuration of the resource. The Oracle Grid Infrastructure will report a fault attributed to another tier (application/network/OS) due to modifications that are inconsistent with the expected configuration.

Error Diagnostic Facility

The Oracle Grid Infrastructure Bundled Agents provides an `oerr xag <errno>` diagnostic facility for error analysis. To use this diagnostic facility, configure $XAGHOME/bin in the admin path. The Oracle RDBMS also uses the oerr facility for error diagnostics. The Oracle RDBMS `oerr` will only report on RDBMS errors. Please execute the oerr facility located in the $XAGHOME/bin directory for XAG related error messaging.

Log files

The xagsetup.sh will log local and remote installation execution to the log file : $XAG_HOME/log/<hostname>/xagsetup_<timestamp>.log. If xagsetup.sh terminates as a result of a failure in the initial stages of installation, this log file will be located in the /tmp directory

AGCTL and agent application execution is logged in separate files in the directory:

$XAG_HOME/log/<hostname>

The AGCTL log file will log all execution related to AGCTL commands for a given application. The xag_application.log will also log corresponding execution on a given Oracle Clusteware application resource.

Oracle Clusterware diagnostic logs are located in the $GRID_HOME/log/<hostname> directory. The most common logs to evaluate would be the alter<hostname>.log which is the overall status of the Clusterware and would include diagnostic data related to application agent activity. The other diagnostic data should be evaluated with guidance from Oracle Support.

Oracle Grid Infrastructure Cluster Health Monitor

The Oracle Grid Infrastructure ships with an integrated system health monitor that collects and reports on cluster wide system metrics that affect the performance and availability of the cluster. These metrics are captured and can be used for offline replay for performance diagnostics or root cause analysis. Please see the *Oracle Clusterware Administration and Deployment Guide 11g Release 2 (11.2) Appendix H Troubleshooting* for more details.

Resource profile

Application administrators have read/write access for application resources under their control. The output from `crsctl status resource <resource name> -t` would provide run time status. The complete profile for an application resource can be viewed with $GRID_HOME/bin/crsctl status resource<resource name> -f which would produce output similar to the attribute key/value pairs listed above for each application resource.

Support

The Oracle Grid Infrastructure Bundled Agents are fully supported by the Oracle Support organization. Supplementary documentation is available on My Oracle Support.

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together

ORACLE