ICMLIVE
Innovyze Live Server Administration

Innovyze®

Empowering water experts

Overview

Note: This document is a guide for systems administrators of the Innovyze Live Server. The Innovyze Live Server must be installed and configured if using an ICMLive Operator client. The Innovyze Live Server is not required if using an ICMLive Configuration Manager.

The Innovyze Live Server is a component that provides the following functions:

- Schedule, monitor and manage automatic simulations.
- Provides real-time services and data to ICMLive Operator clients.
- Performs simulation maintenance.

For the server to carry out these functions, it requires connectivity to a database (usually a WorkGroup Data Server, but the database can also be standalone) and an InfoWorks Agent. The InfoWorks Agent does not have to be configured as the coordinator, but if it is not the coordinator then the agent must connect to a coordinator in order to run simulations.

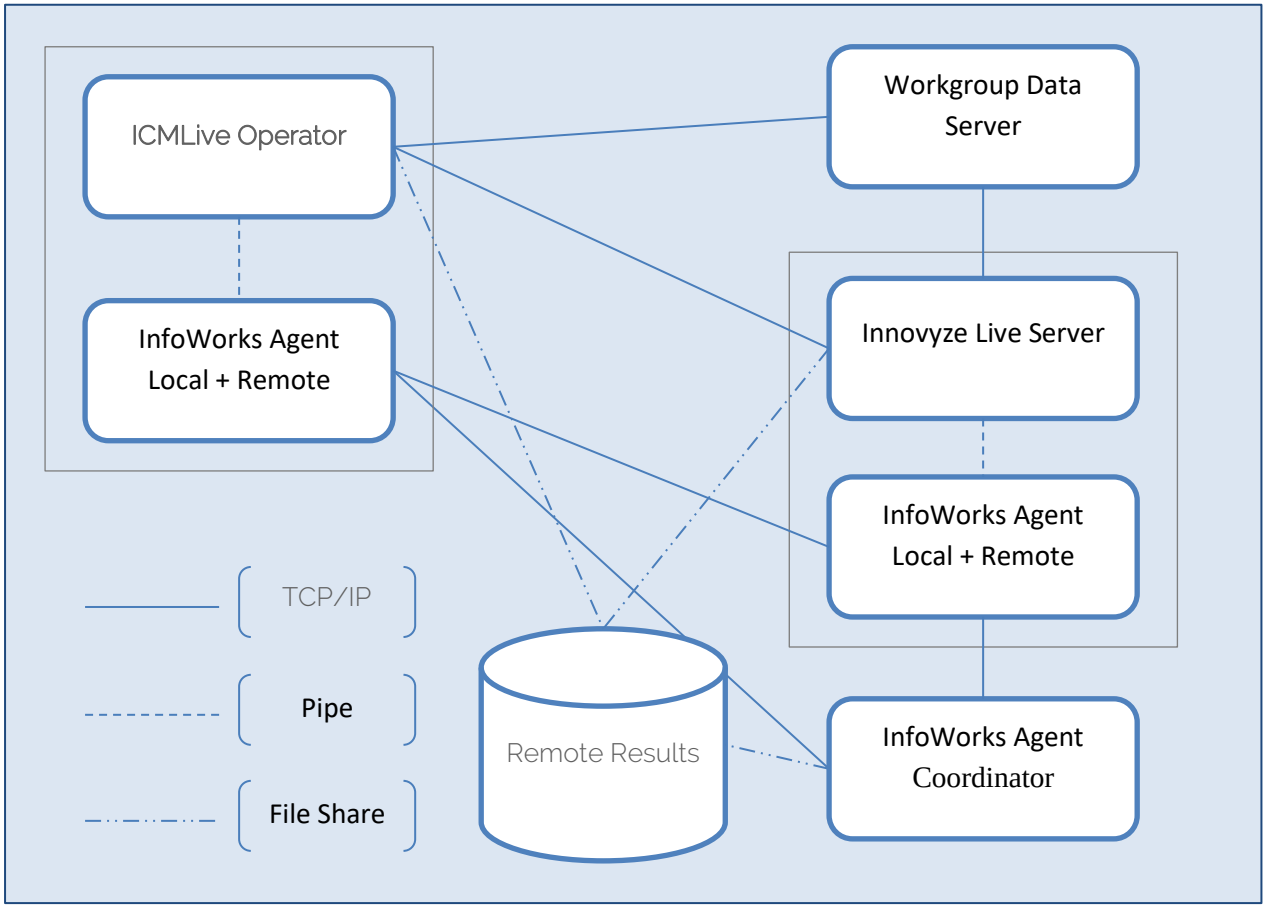


Figure 1Typical client-server architecture

The server is

normally installed on a different computer to those used to run the Live Operator client software. The same server is used by all clients.

The Live Operator client software communicates with the Innovyze Live Server TCP/IP. By default, the port used is 35001. The port number may be changed.

Both the Live Operator and Live Server connect to a local InfoWorks Agent. The Live Operator may also connect via TCP/IP to the local agent used by the Live Server.

The Remote Results file share is accessed by the Live Operator and Live Server for various reasons, including but not limited to the determining the status of results stored on the remote results file share. Most accesses to the Remote Results file share are in read-only mode; however, the first time a set of results are opened a cache file is created and stored on the Remote Results file share, so all applications require read and write access to the Remote Results file share.

The InfoWorks Agent Coordinator is responsible for writing results directly to the Remote Results file share, therefore it is recommended that the file share reside on the same physical device as the InfoWorks Agent Coordinator.

Depending on manifest configuration the Live Server launches one or more of the processing jobs that are managed by the Local InfoWorks Agent. If the Data Loader service is also running in support of the Live Server, TSDB update jobs may also be handled by the Local InfoWorks Agent. See Appendix B: Job Diagrams for a diagram of each job type that is run in support of the Live system.

Installing the server

Innovyze Live Server can only be installed on a Windows x64 platform.

The server is installed by running the setup program found in the \innovyzeliveserver directory on the installation media or downloaded from our website. This setup will install a Windows service and requires Microsoft Windows Vista or later. Once the server is installed it must be configured and the service started before use.

Configuring the server

When installing the Innovyze Live Server for the first time, the server must be configured using the **Server Configuration Manager**, which is installed with the server. The Innovyze Live Server configuration may be changed at any time using the **Server Configuration Manager**, which is the recommended method for making all configuration changes to the server.

The configuration file

There are several parameters that may be configured for the server. These options are set in a simple text file named **liveserver.cfg**.

The Innovyze Live Server installer installs a default version of this file, but the database parameter must be set after the first install.

The configuration file is stored in the following location:

```
%ProgramData%/Innovyze/LiveServerData


example: C:/ProgramData/Innovyze/LiveServerData
```

Manual editing of the configuration file should not be performed without advice from the Innovyze Support team. In the event that the configuration file must be manually edited, the following constraints apply:

Any user editing the configuration file must have administrator permissions on the local machine and the file must be edited using a text editor that has been run as an administrator.

The file should use UTF-8 encoding if it contains non-ASCII characters.

All parameters have the format keyword=value, but must be located in the correct section. Any line that starts with # in the first column is a *comment* line and is ignored.

For example:

```
# comment
[Section1]
Keyword1=value
Keyword2=value
[Section2]
Keyword1=value
```

Parameters

The following parameters may be configured to meet specific user requirements. Each parameter has a default value that will be used when the user does not provide the keyword value pair.

Some parameter values are restricted to a given range, using the following rules: 1) If the configured value is less than the lower bound value in the range, then the parameter value is clamped to the lower bound value; 2) If the configured value is greater than the upper bound value in the range, then the parameter value is clamped to the upper bound value. For example, assume the given range is 1 to 365 (#range: 1..365). If the configured value is 0, then the value is clamped to 1. If the configured value is 400, then the value is clamped to 365.

Section	Keyword	Value
Database	ConnectString	Database connection string. The value is in the same format as abbreviated in ICM most recent databases. #default:ConnectString=

Database	AgentPort	Database agent handling client connections. Specifies the port that the agent listens to that handles the database defined by the Database.ConnectString parameter. #default: AgentPort=35001 #range: 1..65535
License	DongleType	Type of dongle that contains the server license. 0 = Auto-detect local 1 = Reserved (do not use) 2 = Old style local (Sentinel) 3 = Old style network (Net Hasp) 4 = New style local (Hasp HL Max) 5 = New style network (Hasp HL Net) 6 = Local soft license 8 = Network soft license (on-demand) #default: DongleType=0
License	NetworkDongleAddress	Location of the network license - either a computer name or an IP address. Only required when using a network license (DongleType equals 4, 5, 7 or 8). #default: NetworkDongleAddress=
License	NetworkDongleKeyID	Hasp Key ID. Only optionally required when using a Hasp HL Net network license or an on-demand network soft license (DongleType equals 5 or 8). #default: NetworkDongleKeyID=
SimControl	Hostname	Sim agent configuration for the server local InfoWorks Agent when running on a remote computer. Hostname = the name of the computer that will run simulations on behalf of the server #default: Hostname=
SimControl	SimPort	Sim server configuration for the server local InfoWorks Agent when running on a remote computer. SimPort = the port that the sim agent is listening to run simulations on behalf of the server. #default: SimPort=40001 #range: 1..65535

SimControl	UseLocalPipe	Sim agent configuration for the server local InfoWorks Agent when running on the local host. The value, either 0 or 1, indicates whether or not to use the InfoWorks Agent running on the local machine. UseLocalPipe=1 - ignores Hostname and SimPort and the server connects to the InfoWorks Agent running on the localhost using pipes. UseLocalPipe=0 - Hostname and SimPort must be configured to a valid remote host / port and the server connects to the InfoWorks Agent running on the remote host using TCP/IP. #default: UseLocalPipe=1

SimControl	RunOn	The computer where simulations are run.
		
		`.  = this computer only`
		
		`*  = any computer`
		
		`<group> = any computer that is a member of the named group`
		
		#default: RunOn=*

SimControl	RunAsLogonUser	Specifies the username credentials used for simulation jobs. RunAsLogonUser=0 – simulations run as the original pseudo-username (InnovyzeLiveServer) that was used as the username at the inception of ICMLive. RunAsLogonUser=1 – simulations run with the username credentials that the Live Server service is configured to logon as. #default: RunAsLogonUser=1
SimControl	MaxSimEngineThreads	Sim engine multi-threading parameter to control the maximum number of simulation engine threads. Any value up to a maximum value of 99 overrides the InfoWorks Agent setting, but a value of 0 indicates to use the InfoWorks Agent configured value. #default: MaxSimEngineThreads=0 #range: 0..99

Log	EventSourceLevel	Event source logging level. 2 = general information about a large variety of events 3 = status events such as initialization, shutdown etc. 4 = warnings 5 = errors #default: EventSourceLevel=2
Log	LogFileEnabled	Enable the application text log file. When the log is enabled, the file will be written to %LocalAppData%/Innovyze/LiveServerData/LiveServer.log. #default:LogFileEnabled=0
Log	LogFileLevel	Text log file logging level. 1 = debug information (everything) 2 = general information about a large variety of events 3 = status events such as initialization, shutdown etc. 4 = warnings 5 = errors #default: LogFileLevel=1
Log	LogFileLocalTime	Use local time in the text log file. If false or not defined, then UTC time is used in the log (only applies to the file log). #default: LogFileLocalTime=1
Log	LogFilePath	Text log file path. Specifies a directory that will be written to as the server log that overrides the default text log file location. The log file is <LogFilePath>/LiveServer.log. #default: LogFilePath=
Log	LogFileMaintMinutes	Text log file maintenance is run once per day; this parameter specifies how many minutes past midnight to run text log maintenance. #default: LogFileMaintMinutes=15 #range: 0..1439

Log	LogFilePurgeDays	Text log file purge days: Purge old text log files after the number of days specified. #default: LogFilePurgeDays=30 #range: 1..365
Monitor	ConnectFrequencyMinutes	How often to retry the database connection if it cannot be verified at startup. #default: ConnectFrequencyMinutes=10 #range: 1..1440
Monitor	RetryInitStateMinutes	How often to retry initializing state when a database connection fails after it has been verified when the service is starting. #default: RetryInitStateMinutes=1 #range: 1..30
Monitor	ScanFrequencyMinutes	How often to scan the database for changes, such as new manifests, modified run schedules, etc. #default: ScanFrequencyMinutes=6 #range: 1..43800
Factory	CancelJobMinutes	The number of minutes after a job is submitted to the InfoWorks ICM Agent that it will be cancelled by the Live Server if the job is still incomplete. If the value is 0, then submitted jobs will never be cancelled by the Live Server, but by the InfoWorks ICM Agent per its 'cancel inactive jobs after days' configuration. #default: CancelJobMinutes=60 #range: 0..1440
Factory	RetryFrequencyMinutes	For purposes of creating and dispatching simulations, the frequency to retry database and sim agent connections when previous connection attempts failed. #default: RetryFrequencyMinutes=10 #range: 1..60
Factory	DeleteStateHours	The number of hours that must elapse before current or previous simulation data can be deleted from persisted state. #default: DeleteStateHours=72 #range: 1..720

Factory	CleanupFrequencyMinutes	How often to clean-up state information. #default: CleanupFrequencyMinutes=360 #range: 1..1440
Messenger	BufferDays	The number of days from the current date that the server will buffer messages that are provided to the Live Operator client. This controls the number of server messages that a user may view in the Live Operator client. #default: BufferDays=3 #range: 1..30
Messenger	MaxMessages	The maximum number of messages that the server will buffer within the BufferDays period. The maximum allowed value is 2000. #default: MaxMessages=500 #range: 1..5000
Messenger	PurgePastMidnightMinutes	Old server messages are purged from memory once per day; this parameter specifies the number of minutes past midnight to purge the previous day messages. Note: these messages are only purged from the server buffer, not from the system event log. #default: PurgePastMidnightMinutes=3 #range: 0..1439
Maintenance	SimMaintMinutes	Sim maintenance is run once per day; this parameter specifies how many minutes past midnight to run sim maintenance. #default: SimMaintMinutes=30 #range: 0..1439
Maintenance	RetryFrequencyMinutes	For purposes of running sim maintenance, the frequency to retry database and sim agent connections when previous connection attempts failed. #default: RetryFrequencyMinutes=15 #range: 1..60

Maintenance	RetryCount	For purposes of running sim maintenance, the number of times to retry failed database and sim agent connections before terminating the daily sim maintenance task. #default: RetryCount=12 #range: 1..24
Maintenance	LogMaintMinutes	Sim log maintenance is run once per day; this parameter specifies how many minutes past midnight to run sim log maintenance. #default: LogMaintMinutes=10 #range: 0..1439
Maintenance	PurgeLogFileDays	Sim maintenance log file purge days: Purge old sim maintenance log files after the number of days specified. #default: PurgeLogFileDays=60 #range: 1..365
Run	RunOriginNoDST	Run origin times with no daylight savings time. The value, either 0 or 1, indicates whether or not to ignore daylight savings time when calculating run origin dates. RunOriginNoDST =1 – The run origin uses the local time, ignoring daylight savings time. RunOriginNoDST =0 – The run origin uses the local time only. #default: RunOriginNoDST=1
Scheduler	NumLateMinutes	Do not initiate late simulations if the next scheduled simulation starts within the specified time period. #default: NumLateMinutes=15 #range: 1..1440
Sentinel	LastFileWriteToleranceSeconds	Do not initiate an externally triggered run if the last triggered file write has occurred within the tolerance specified since the last file write. #default: LastFileWriteToleranceSeconds=5 #range: 0..299
Sentinel	NumStaleMinutes	Do not initiate an externally triggered run if the last trigger file write occurred earlier than the current time minus the number of minutes specified. #default: NumStaleMinutes=15 #range: 1..1440

TaskScheduler	NumLateMinutes	Do not initiate late ICM Live tasks if the next scheduled task starts within the specified time period. #default: NumLateMinutes=30 #range: 1..1440
TransLog	StartLogMaintMinues	When to initiate the maintenance job to purge transaction logs. Transaction log maintenance is run once per day. This parameter specifies how many minutes past midnight to initiate transaction log maintenance. #default: StartLogMaintMinues=20 #range: 0..1439
TransLog	PurgeLogFileDays	When to purge an old transaction log. Specifies the number of elapsed days before the transaction log is deleted by transaction log maintenance. #default: PurgeLogFileDays=60 #range: 1..365

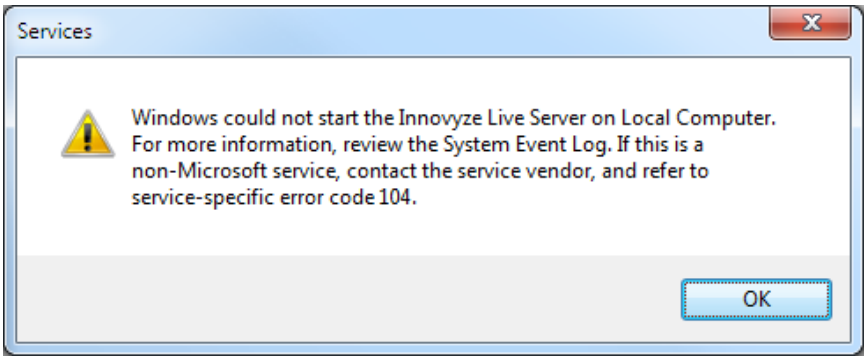
Starting the server

Before starting the server for the first time, or following an upgrade, it is recommended that dependencies on other required services be established. See Appendix A: Innovyze Live Server dependencies for information on how to establish service dependencies for Innovyze Live Server.

The service is started by opening the Control Panel and browsing to Local Services and starting the **Innovyze Live Server** service.

Troubleshooting when the server fails to start

If the service fails to start due to a **service-specific** error, a dialog similar to the following will be displayed along with an error code. Additional detailed information may exist in the event log. This information can be found by starting opening the Control Panel and browsing to the Event Viewer. When the Event Viewer is open look for errors under Application logs with a source name of InnovyzeLiveServer.



The following table describes the service-specific error codes.

Error code	Description	Possible resolution
100	An internal program error has occurred.	Contact the Innovyze support team. Detailed information about the internal error should exist in the event log.
101	A database connection string was not found in the configuration file.	Run the Server Configuration Manager to configure a database for the server.
102	The configuration file does not exist.	Run the Server Configuration Manager to configure the server.
103	The server is not licensed. Additional information is provided in the event log.	If using a local dongle, verify that the dongle is inserted. If using a license manager, verify that the service is running. Verify that the license is correctly specified in the configuration file.
104	A database error occurred. Additional information is provided in the event log.	Run the Server Configuration Manager to verify that the Database is correct. If connecting to a Workgroup Data Server, verify that it is running. Verify that the database does not require a sub-version update.

Server data

The server produces data that is for internal server use, but some of the data may indirectly be provided to Live Operator clients. To maintain correct operational behaviour, this server data should not be edited or deleted.

Data location

The configuration file is stored in the following directory:

```
%ProgramData%/Innovyze/LiveServerData/<databaseGuid>


where databaseGuid = the server database guid


example: C:/ProgramData/Innovyze/LiveServerData/EF736CC3-73EC-F031-0952-C4E38E3E1AD2
```

Data anatomy

The data for each master database exists at a separate location, as shown by the data location directory. Within each of these directories are local working directories for internal use by various server thread components.

The following server files contain server state information. To maintain correct operational behaviour, this server data should never be edited or deleted.

- FileWatch.xml
- Maintenance.xml
- SimFactory.xml
- TaskFactory.xml
- TaskScheduler.xml

The SERVER service may produce daily text log files for each day that the server is run, but only when the text log file is enabled per the Log.LogFileEnabled parameter. These files are stored in the server_logs directory. It is OK to manually delete these files, but it is recommended to allow the server to maintain these files per the Log.LogFilePurgeDays parameter.

The MAINTENANCE service produces a log file for each day that simulation maintenance is run. These files are stored in the maintenance_logs directory. It is OK to manually delete these files, but it is recommended to allow the server to maintain these files per the Maintenance.PurgeLogFileDays parameter.

The TRANSLOG service produces log files in the run_logs and task_logs directories. These files are for internal and third-party use. Therefore, these files should never be manually deleted. It is recommended to allow the server to maintain these files per the TransLog.PurgeLogFileDays parameter.

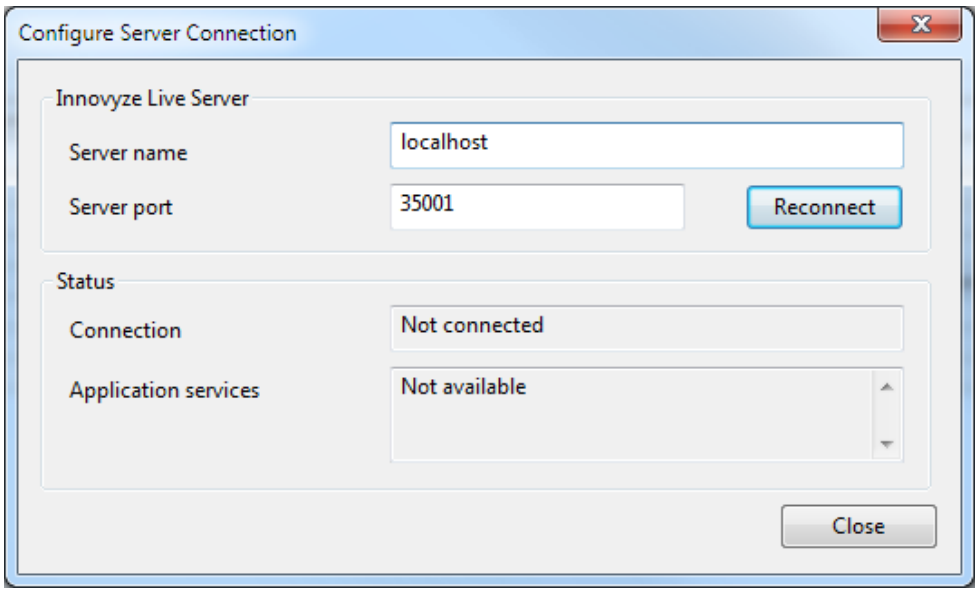
Other files or directories that may appear in the server data directory should never be edited or deleted without advice from the Innovyze Support team.

Connecting Live Operator clients to the server

The Live Operator client connects to the server via the *Server connection* command located on the file menu or Overview toolbar.

The configure a server connection dialog

When the server connection command is invoked, the following dialog is launched:



The following entities are to be configured by the user:

Server name – this is the name of machine where the server is located.

Server port – this is the port on the machine where the server is located and the server is listening to for client connections.

Reconnect – Once the user has selected a server name and server port, pressing the reconnect command will change the new client-server connection configuration.

The following entities are status information regarding the current client-server connection:

Connection – The status of the connection may be one of the following:

- Connecting – The client has initiated an attempt to connect to the server. When the client is not connected to the server, it will automatically attempt to reconnect to the server until a connection has been successfully established.
- Not connected – This indicates that no connection can be established between the client and server. This can be for many reasons, including but not limited to the following:
  - The server is not running on the machine specified as server name.
  - The server is not listening on the port specified by server port.

    - The network is down.
    - Etc.
- **Connected** – This indicates that an instance of the server is running on the machine indicated by **server name** and **server port**. In other words, a basic connection has been successfully established between the client and server.

Applications services – The status of the application services may be one of the following:

- **Available** – All application services are available to the client.
- **Not available** – No application services are available to the client. The reason that application services are not available is included in the application services message. This can be for many reasons, including but not limited to the following:
    - The server is using a different database
    - The client and server versions do not match
    - Etc.

Upgrading the server

The existing older Innovyze Live Server service should be stopped before attempting to install a newer version. The service is stopped by opening the Control Panel and browsing to Local Services and stopping the **Innovyze Live Server** service.

Appendix A: Innovyze Live Server dependencies

If a Sentinel HASP License Manager service (hasplms), an InfoWorks ICM Agent service (infoworksagent) or an Innovyze Workgroup Data Server (snumbat) service used by Innovyze Live Server is running on the local machine, then a dependency should be created so that these services start before the server. It is recommended that dependencies be set through the **Server Configuration Manager**, but dependencies may also be set manually.

Manual configuration of service dependencies

Dependencies may be manually configured from a command prompt with administrator privileges:

```
sc config innovyzeliveserver depend= <services-string-slash-seperated>
```

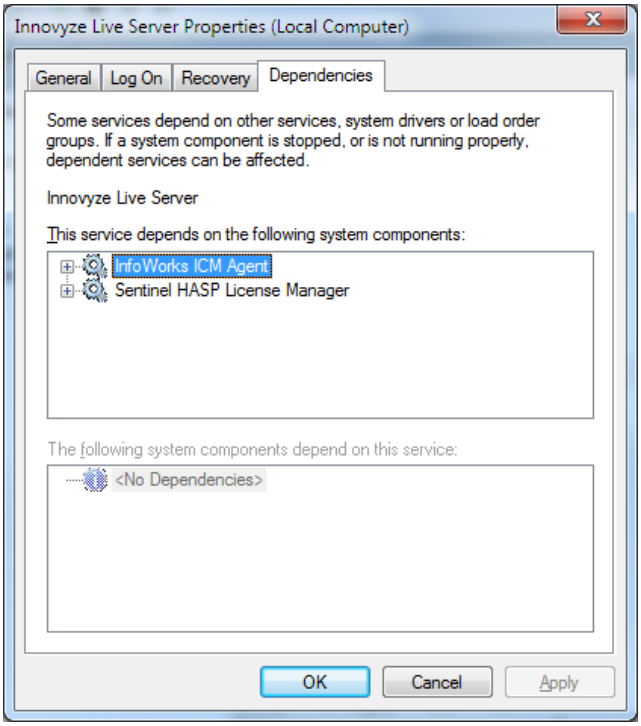
*Note: the space character following the **depend=** parameter is required. If more than one service dependency is required, then each dependent service name is separated by a slash ('/') character.*

Example: Set dependency for the Sentinel HASP License Manager and the InfoWorks ICM Agent services.

From a command prompt with administrator privileges, enter the following:

```
sc config innovyzeliveserver depend= hasplms/infoworksagent
```

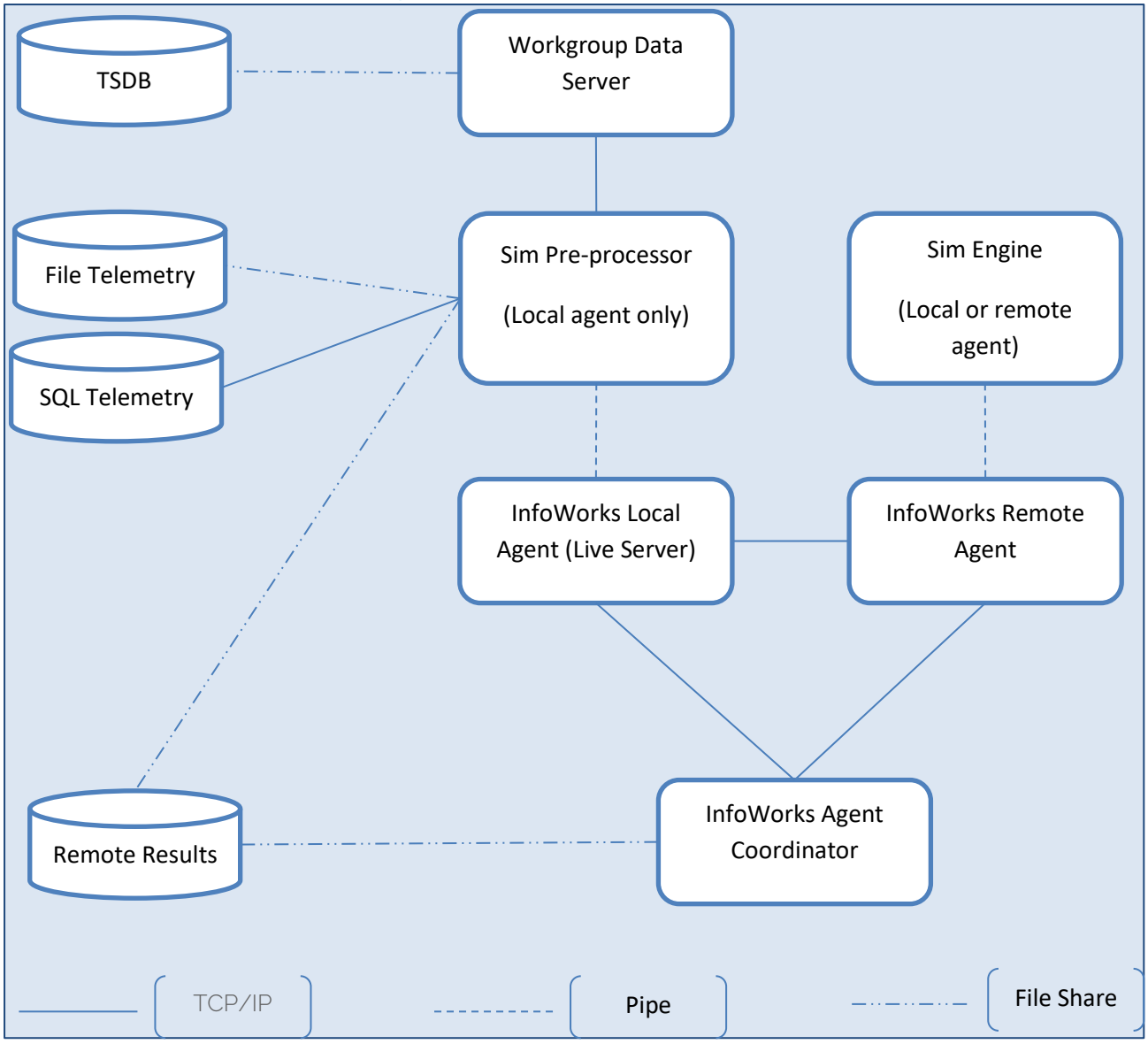
Go to the Control Panel Services window and verify that the dependencies have been added.



Appendix B: Job Diagrams

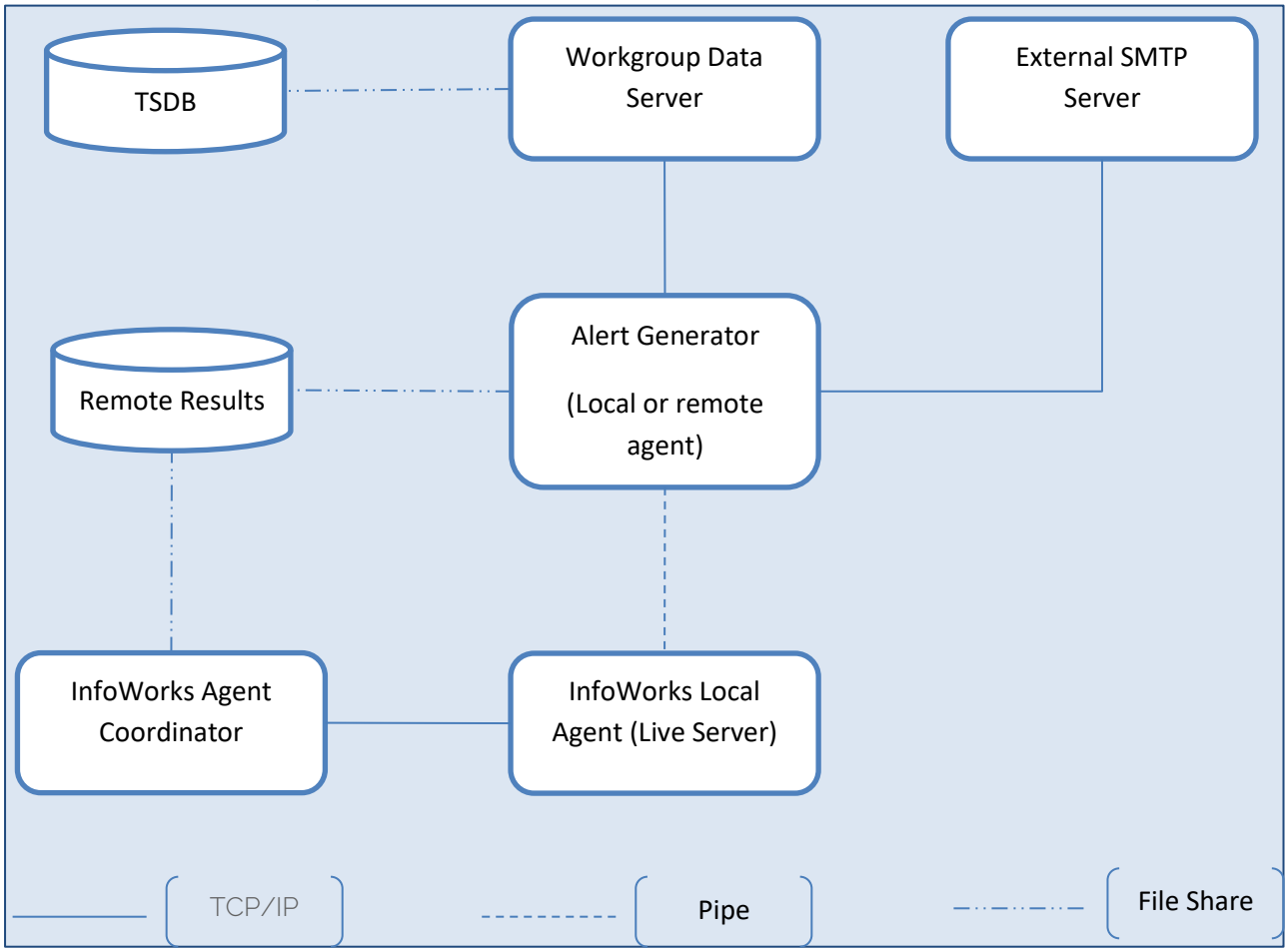
The following list of job diagrams show the interaction of the InfoWorks Agent with various software components and databases.

Simulation jobs initiated by the Live Server local InfoWorks Agent



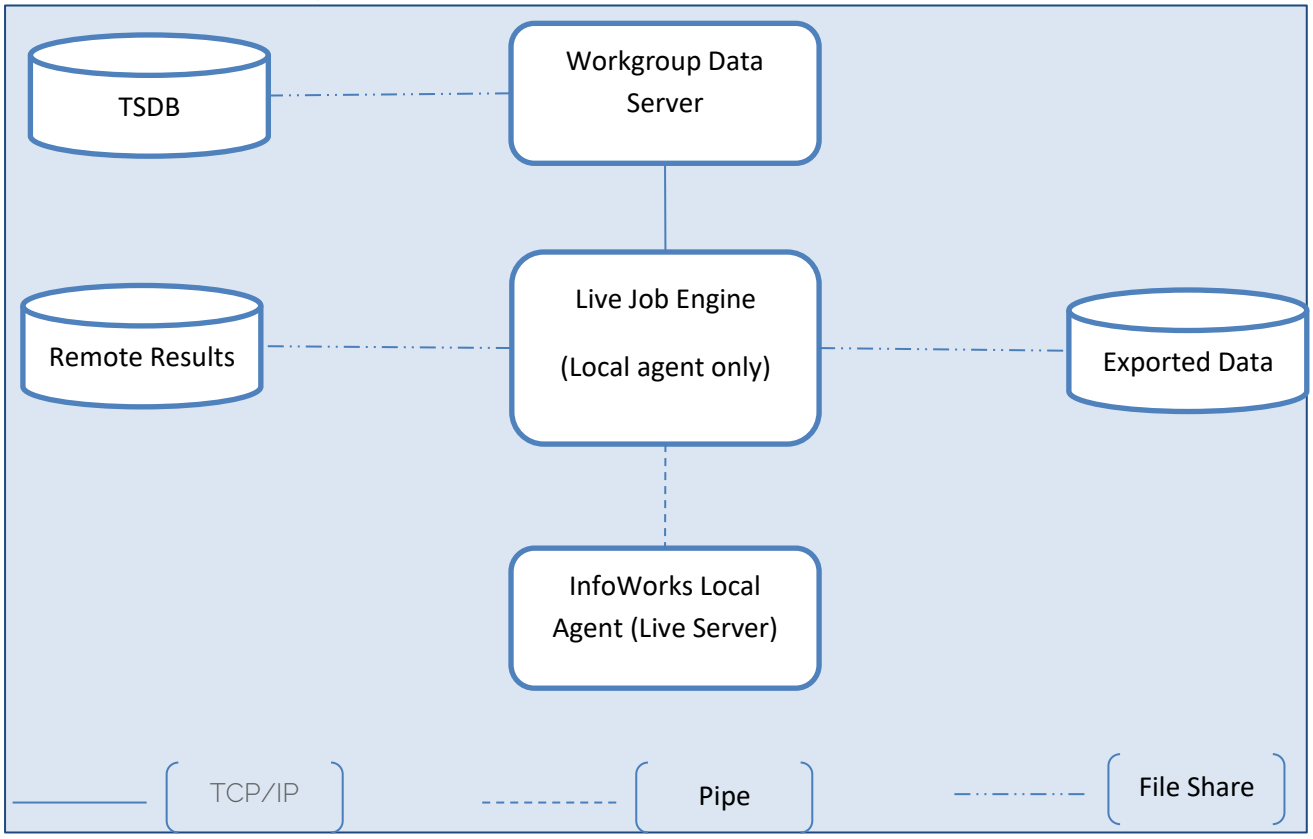
Simulations are always pre-processed by the local InfoWorks Agent, but the simulation engine may run on any available remote InfoWorks Agent.

Alert Jobs initiated by the Live Server local InfoWorks Agent



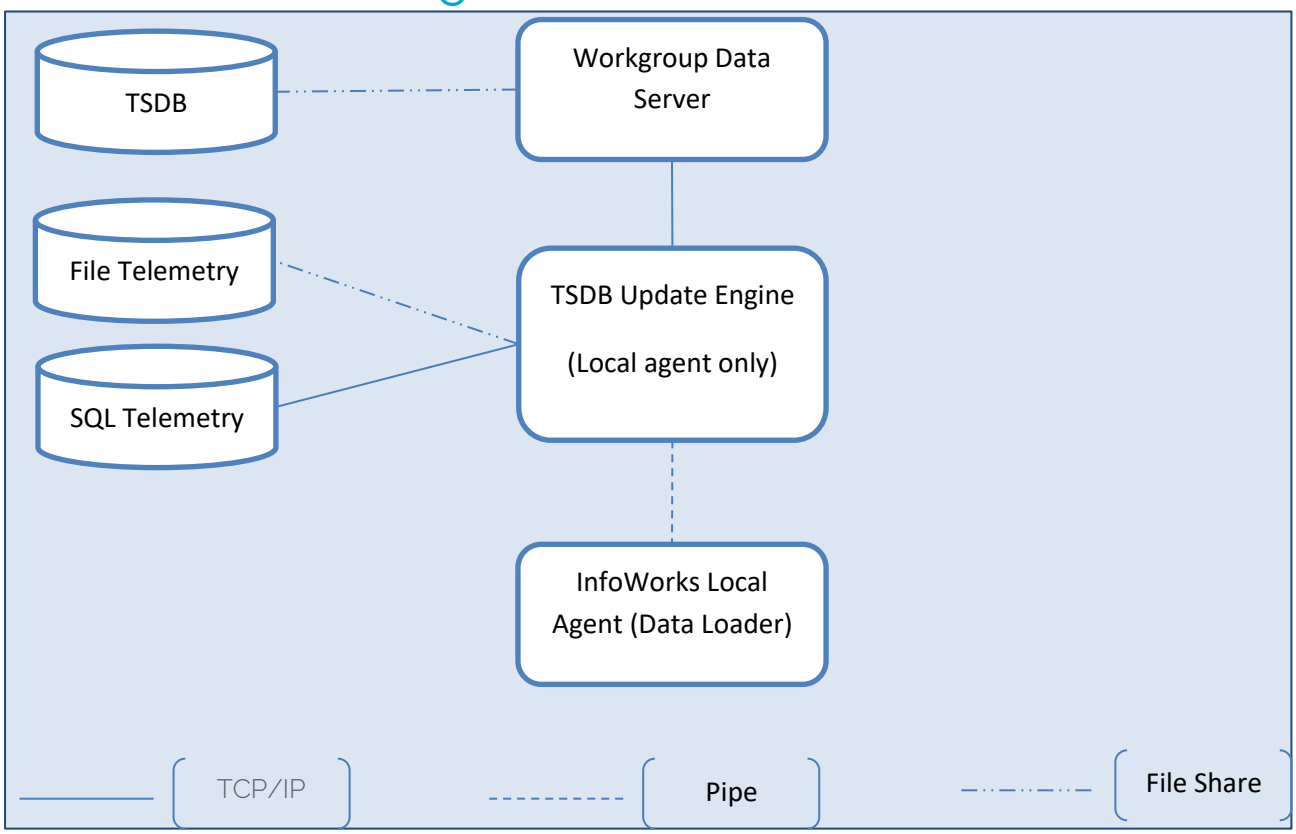
Observe that the Alert Generator is initiated by the Live Server local agent. It may then run on either the local agent or a remote agent. The Alert Generator may optionally access an external SMTP Server to send alert action emails, if configured and an appropriate alert action is raised.

Live jobs initiated by Live Server Local InfoWorks Agent



Live jobs include Results Export and Post Event Analysis jobs. Observe that Live jobs always run on the local InfoWorks Agent, only.

TSDB Update jobs initiated by the Data Loader local InfoWorks Agent



TSDB Update read telemetry and update the TSDB database. Observe that TSDB update jobs always run on the local InfoWorks Agent only, and is initiated by the Data Loader. The Data Loader is usually installed on the same machine as the Live Server for licensing purposes and to facilitate the updating of TSDB databases so that simulation jobs initiated by the Live Server run in a more efficient manner.