ARGENT

Argent Architecture Guide

ARGENT

Argent Architecture Guide

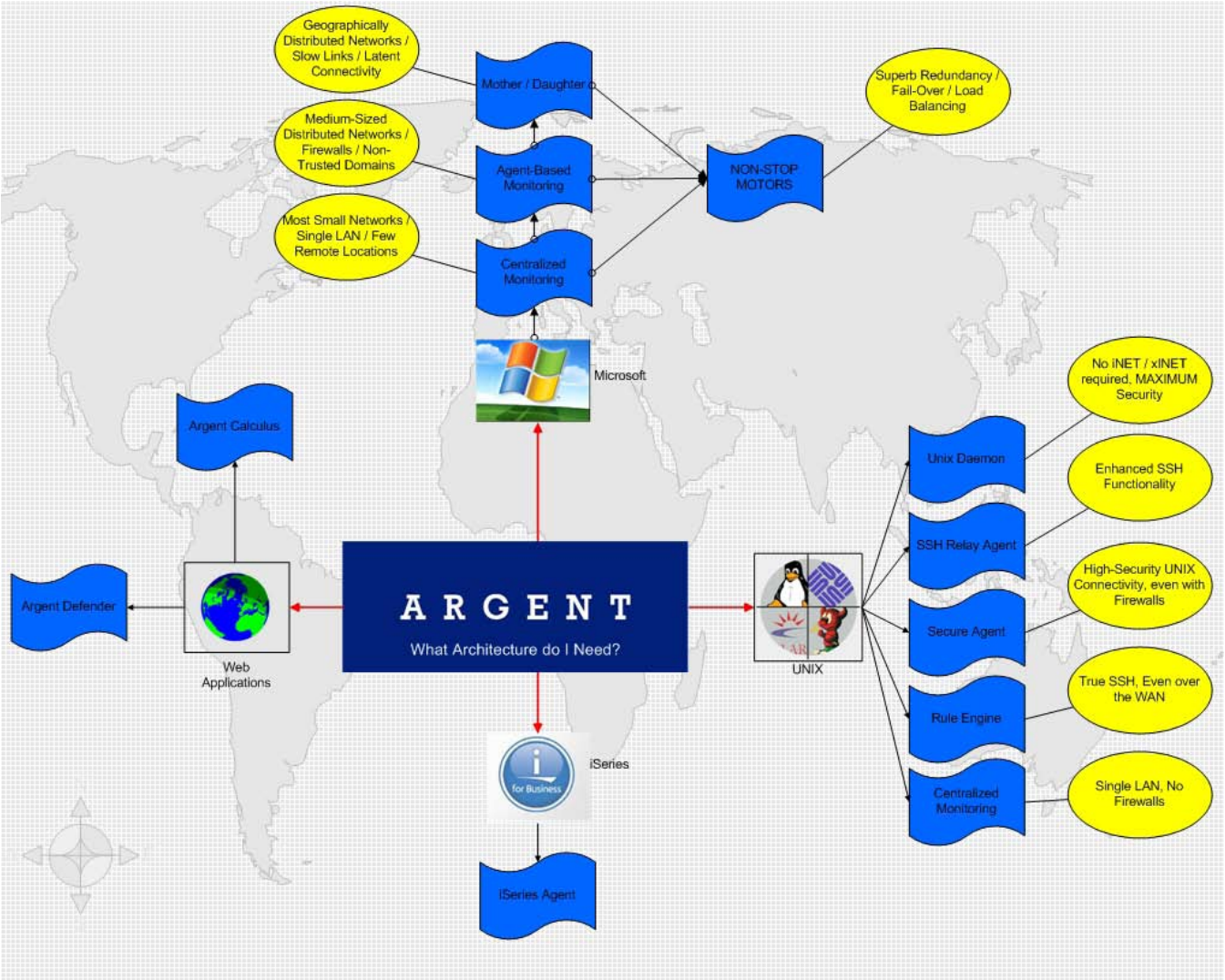
Contents

The Argent Architecture Decision Tree

Argent offers several options with regard to Architecture. The one thing that sets Argent apart from other vendors is our ability to scale so easily. Argent has an architecture option for ANY situation, for example…

Looking at the decision tree below, there is an option for every enterprise.

The Argent Architecture Decision Tree

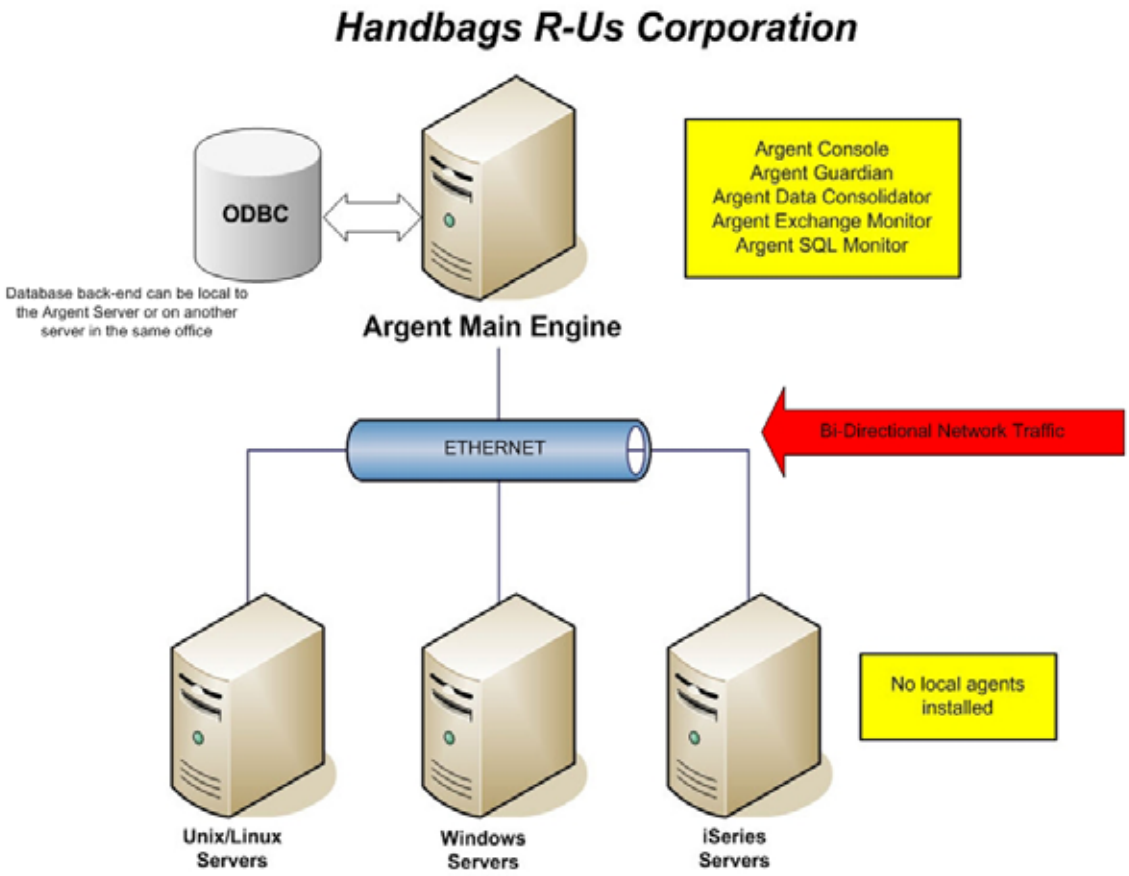


Centralized Monitoring

Centralized monitoring is the most commonly used approach for smaller customers with most of their monitored devices in a single geographic location or subnet.

This approach utilizes a single Argent main engine with a back-end database, either installed locally or on another server in the same location / subnet.

Note: This approach does not provide disaster recovery functionality.



For more information on centralized monitoring, please visit help.Argent.com.

Agent Types

A Daughter installation on a server consists of both a regional Supervising Engine as well as a Monitoring Engine.

In contrast, a Monitoring Engine installation needs to be driven by the Supervising Engine (either Mother or Daughter) on another machine.

If the connection from the Main Engine to the Daughter Engine is lost, the Daughter will still be able to continue scheduling the tasks previously requested from the Main Engine.

In addition, Daughter Engines, just like Mother Engines, store Argent Predictor data locally until the connection is restored to the Main Engine.

In the same way as Mother Engines operate, Daughter Engines can cache Alerts if the Argent Console Main Engine server is offline or the network segment is down. When the Argent Console Main Engine server returns online, the Alerts cached on the Daughter Engine are sent to the main Argent Console.

If an Alert Executor or an Argent Console Backup Engine is installed on the Daughter Engine, Alerts can be sent directly from the Daughter Engine. This avoids possible delays in notification or corrective action.

The Argent Console Backup Engine records Events and fires Alerts if the main Argent Console server is unreachable, while an Alert Executor simply fire Alerts.

In contrast, if the connection from the Main Engine to the Monitoring Engine is lost, the Monitoring Engine will stop monitoring as it cannot receive instructions from the Main Engine.

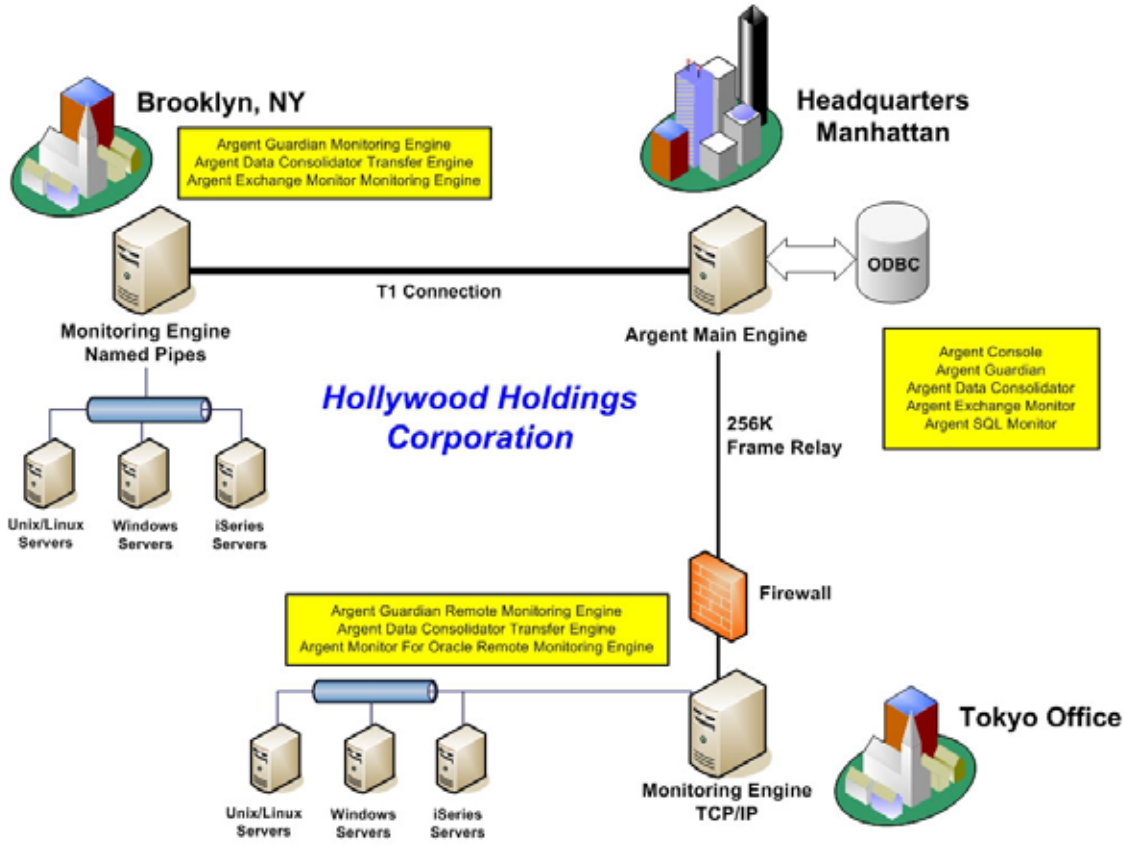
Clearly the Mother/Daughter option is more powerful, but does require more planning and takes longer to implement

A Monitoring Engine is quick and easy to install and is a good choice if the network connection between the Main Engine and the Monitoring Engine is stable.

Agent-Based Monitoring

Agent-based monitoring is very useful for customers with remote locations connected by DEPENDABLE, high-bandwidth links.

This approach is also useful for situations where firewalls are involved OR there are multiple domains resulting in a situation where ONE Argent Service Account does not have administrative connectivity to all machines involved.



Remote Monitoring Engines, Transfer Engines, and Database Engines in Argent communicate with the Main Engine using one of two protocols:

Named Pipes

This is similar to the same communication used when a server is viewed via My Network Places or Network Neighborhood.

Agent-Based Monitoring

TCP/IP Only

This option uses less bandwidth and is limited to one particular IP port per service. This is the option that should be used when monitoring devices through a firewall.

1- To install the remote engines in Argent, select the Administration button.

2- From this location, select Monitoring Engines (or Transfer Engines/Database Engines in the Argent Data Consolidator).

3- Right-click here and select the Install Monitoring Engine option.

4- Select the option Selected Servers Are Located Remotely to use TCP/IP communication only.

When Named Pipes are used, the Monitoring Engine services will be pushed out to the target server.

When TCP/IP is used, the installation files must be physically copied or downloaded to the target server and the Argent Setup program run with the appropriate command-line switches.

Alternatively, the Create Remote Engine Installation Package feature can be used to create an easy-to-use installation package for any number of engines.

When installing a remote engine using the Argent GUI, either of these options can be selected for remote engines.

In addition, these two options can be mixed and matched freely. It is perfectly acceptable to have one Monitoring Engine that has been deployed to use Named Pipes, and another using only TCP/IP.

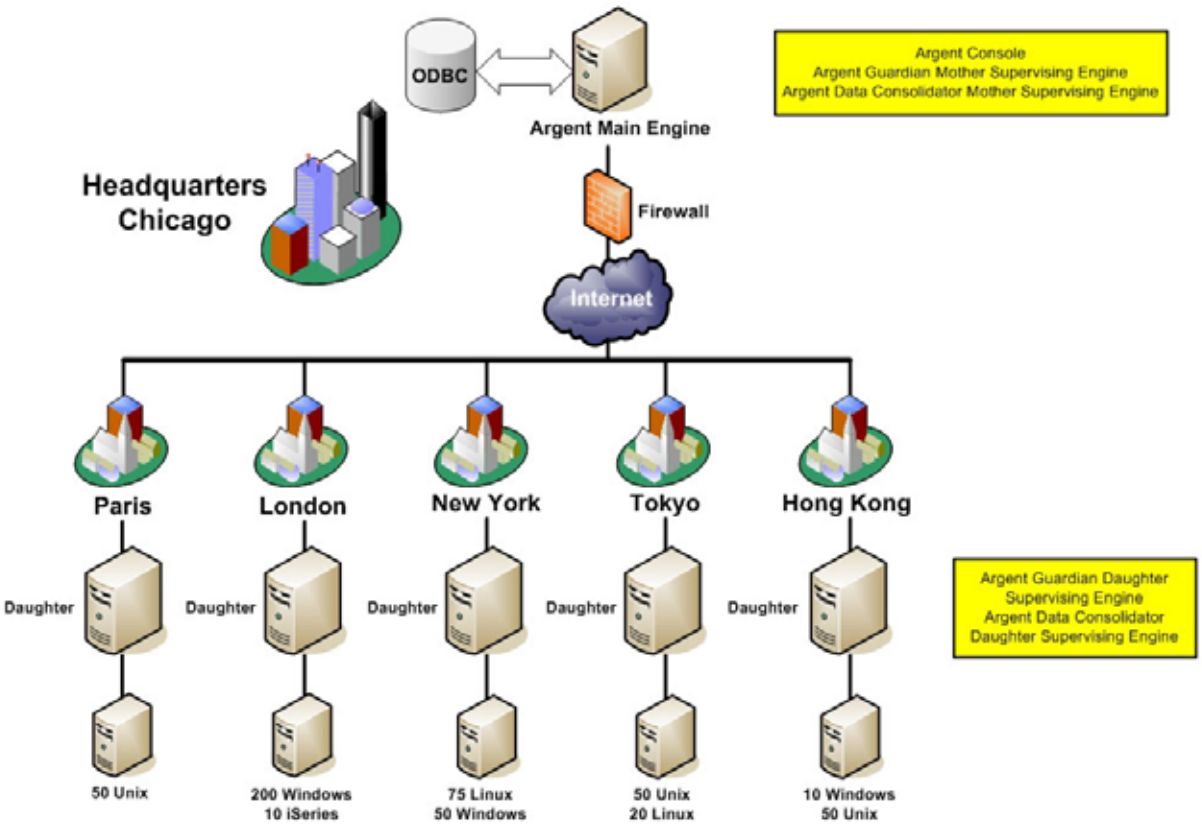
The decision should be made based on whether the server housing the remote engine is over a slow link, is protected by a firewall, or is in another domain.

Mother-Daughter Architecture

Mother-Daughter architecture is the optimal solution for customers with geographically distributed networks, slow links, firewalls, or other complicating factors in their enterprise layout.

Each network location or subnet contains a Daughter Engine pointing back to a central "Mother" Engine. A Daughter engine is more than just a monitoring engine – it contains its own local scheduling engine as well.

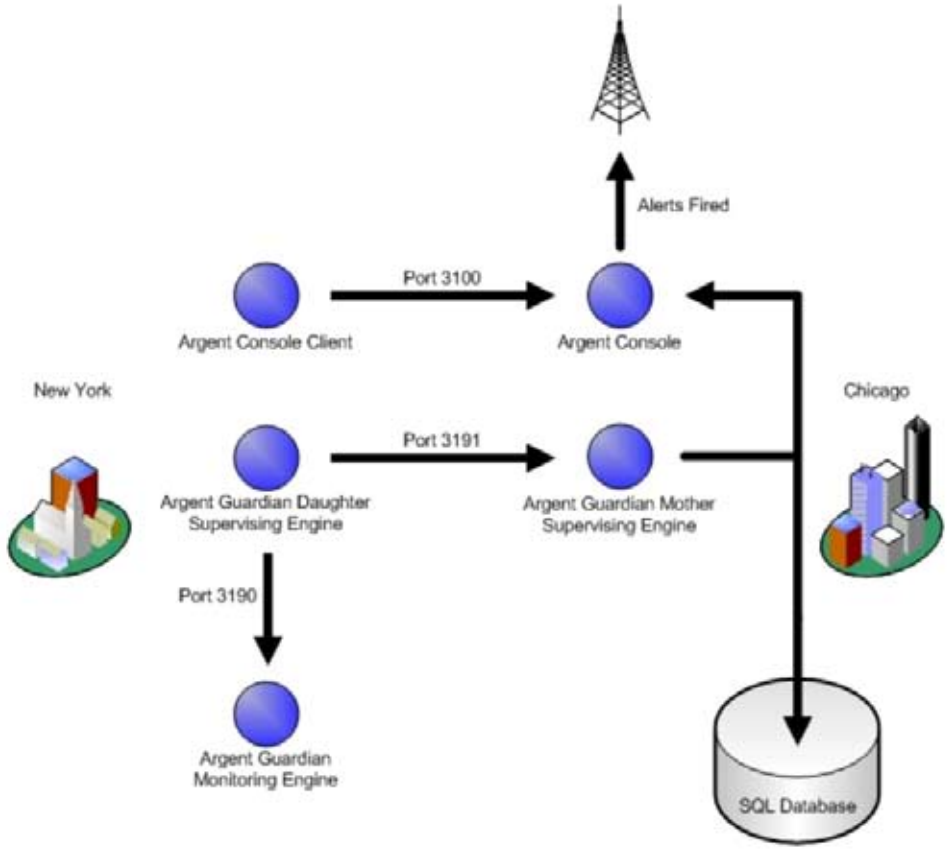
This facilitates continued monitoring when links are slow (or they fail) and introduces more redundancy. With the addition of an ALERT EXECUTOR, the alerting process can also continue when the link to the Mother Engine fails.



Bandwidth is at a premium. The links are of questionable quality and quite unstable. Connectivity is needed for monitoring, alerting, and trend-analysis whilst also handling link failure. The Mother/Daughter Architecture is ideal for this problematic connectivity.

This is what traffic would look like on the Mega-Corp network. (The example shown is traffic between headquarters in Chicago and the New York office):

Mother-Daughter Architecture



It is easy to see how the Mother Supervising Engine for Argent Guardian in Chicago drives the schedule that is picked up by the Daughter Supervising Engine in New York. The following shows the sequence of events:

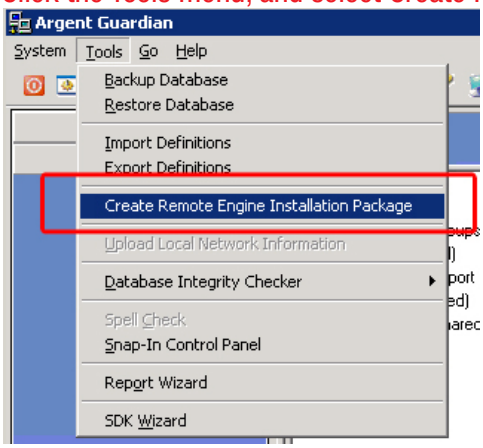
1. *When the Daughter Supervising Engine "phones home" at a set interval (every 60 seconds by default), it checks for updated schedules in the CB_DATA.BKU file.*
2. *If the Daughter Engine detects an updated schedule, it downloads the updated information and re-initializes its monitoring schedule. The Daughter Engine drives its own monitoring engine. The monitoring engine in New York is completely unaware that Chicago even exists.*
3. *The monitoring engine in New York begins monitoring devices in that location as per the schedule that has been initialized by its supervising engine.*
4. *The Daughter Engine periodically uploads trend-analysis data back to the Mother Engine. Daughter Engines always use Code-Base databases.*
5. *The Argent Console Client in New York sends any required alerts back to the Argent Console in Chicago.*

Mother-Daughter Architecture

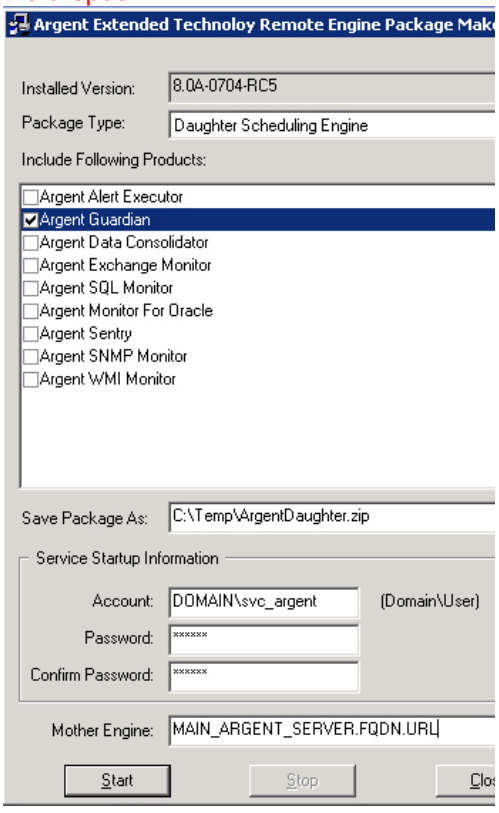
Installing a Daughter Engine is easy using the Argent integrated package creation tool. This approach is ideal for installing the Daughter Engine in a network with limited bandwidth.

From the Argent GUI:

1. Click the Tools menu, and select Create Remote Engine Installation Package.



2. On The Remote Engine Package Maker window (Screen P1), choose Daughter Scheduling Engine from the dropdown.



Mother-Daughter Architecture

3. Select the products to have Daughter Engine created.
4. Specify the path and filename for the package, including the extension.

   (A ZIP archive is created, so .ZIP is preferred. But if you are emailing the file and ZIP files are blocked, you can name it with any extension without affecting the integrity of the archive.)
5. Enter the service account information to use on the remote server, in the form Domain\User.
6. Enter the service account's password, and again to confirm.
7. Enter the name of the Mother Engine. This is the name the Daughter Engine uses to communicate with the Mother Engine, so the name entered here MUST resolve to the Main Argent server from the Daughter Engine.

   If the Main and Daughter are in different domains, the fully qualified domain name or IP address is recommended here.
8. Press Start to create the installation archive.
9. Transfer the completed file to the Daughter Engine server and extract to a temporary location.
10. On the Daughter Engine server, run the Setup.exe file.

By design, the client GUI for the Daughter Scheduling Engine is not included within the Remote Engine Installation Utility.
A command-line utility -- **XT_Daughter_Console.exe**, is located in the Daughter server's directory and it allows simplified administration.

In contrast to the command-line utility, if you wish to implement the full GUI for the Argent Daughter Scheduling Engine, you must first transfer the installation package you used to install the Mother Engine to the Daughter server.

Then do the following:

1. *Run **Setup.exe** on the Daughter server*
2. *Select Upgrade the Argent Management Console*
3. *Once upgraded, you can launch **AMC.exe** on the Daughter server*

This will open the Client GUI for the Daughter Scheduling Engine.

Non-Stop Motors

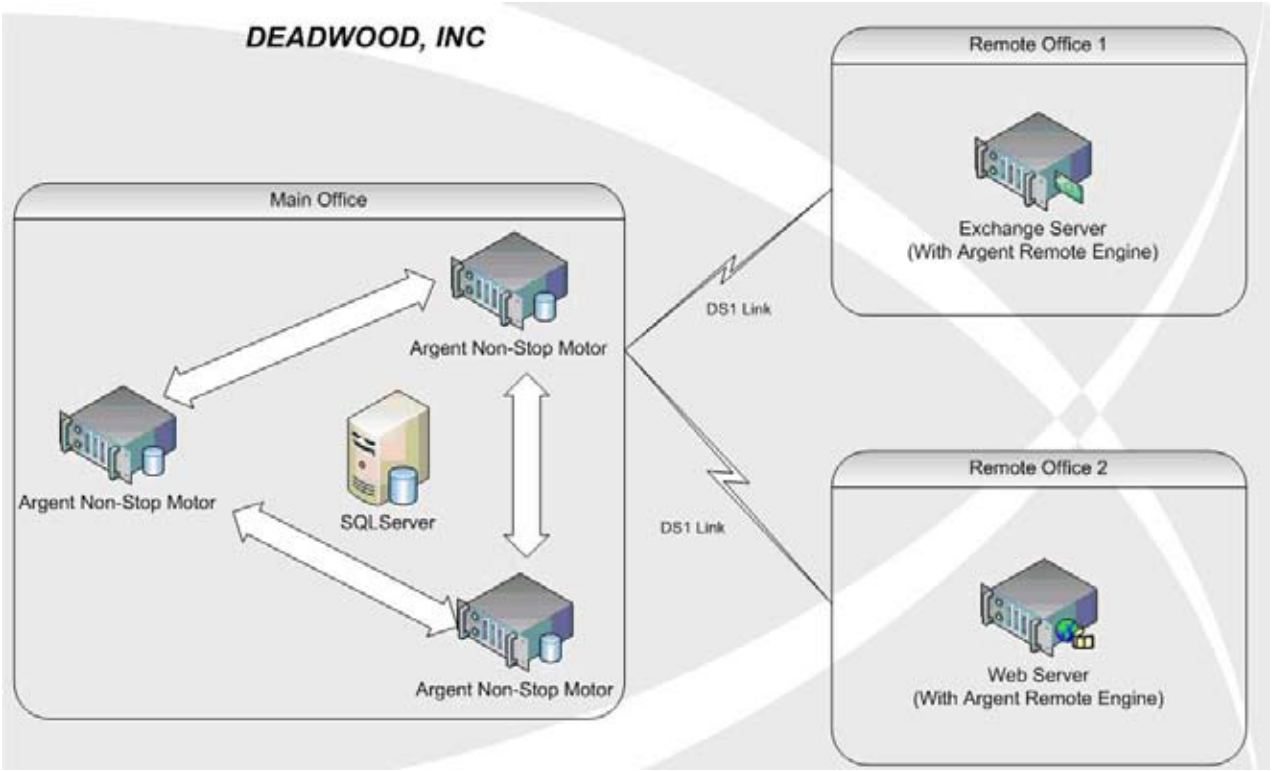
Argent's Non-Stop Monitoring is a significant enhancement to monitoring technology by providing increased load-balancing, expanded scalability, and improved reliability.

Non-Stop Monitoring has been implemented for the Argent Console and the Argent Guardian, and will be added to all Argent Extended Technology products.

Non-Stop Monitoring is essentially a pool of load-balanced Mother Engine servers all sharing the same ODBC back-end database.

The previous architecture only allowed for one Mother Engine server. Under the old scenario, the sole Mother Engine server was required to process all alerting, saving of Argent Predictor data into the ODBC database, communication with Daughter Scheduling Engines, etc. For some customers this could create a processing bottleneck on the Main Engine server.

With the implementation of Non-Stop Monitoring, the processing for all of these tasks can be shared among a pool of load-balanced Mother Engine servers, all sharing the same backend ODBC database. For clarity, the term "Argent Motor" has been introduced to describe a monitoring server that processes work from the common pool.

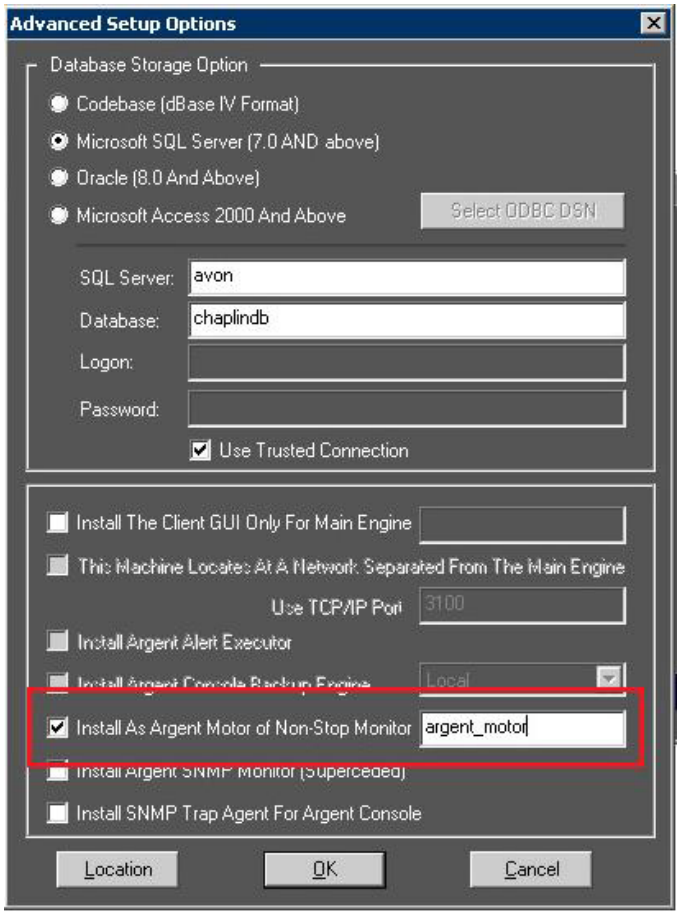


Non-Stop Motors

Deploying a Non-Stop motor is very simple. First, set up your main engine as you would normally, and then determine which other machines are to be a part of the Non-Stop Pool.

Execute SETUP on each member, and ensure you check the box for “install as Argent Motor of Non-Stop Monitor “XXXXX” where XXXXX is the name of an EXISTING member (main engine or otherwise).

ALL of the members must point to the same SQL back-end. In the case below, SQL server AVON is used, with database name “CHAPLINDB.”



Again, All Argent Motors share the same ODBC backend (SQL Server or Oracle). To gain the ultimate reliability, the ODBC backend should be a highly available implementation - database clustering is strongly recommended.

Non-Stop Motors

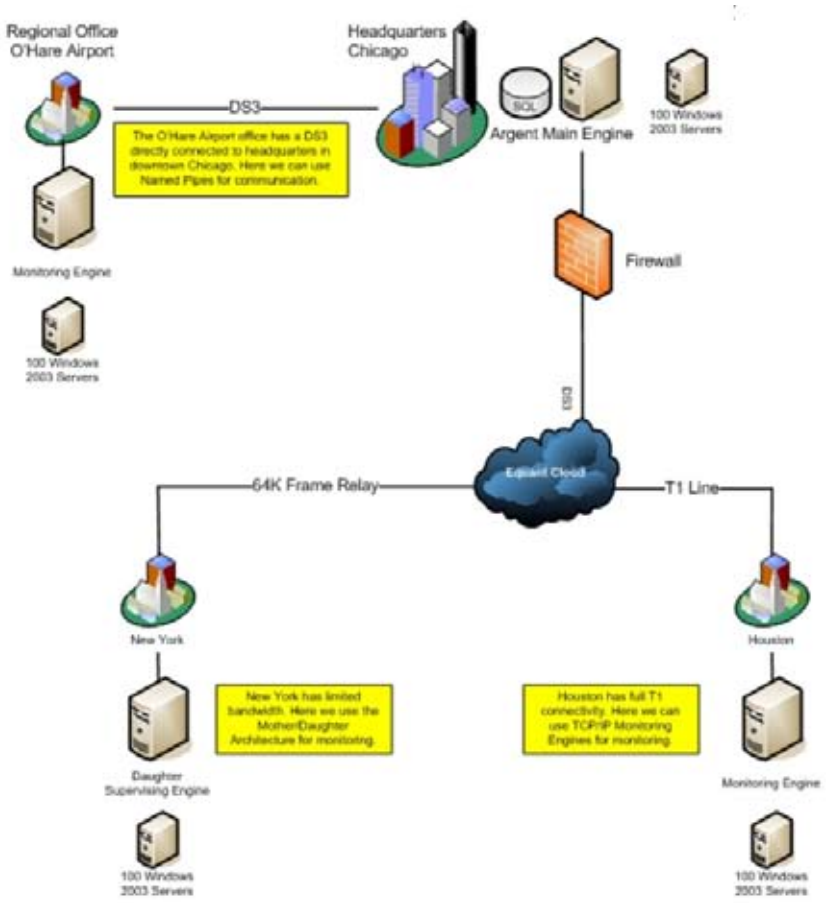
Argent's Non-Stop Monitoring is a completely distributed design. No central management is required. An individual Argent Motor can be dynamically added and removed without affecting the functioning of system. The load is automatically distributed among the functioning Argent Motors.

Note: Non-Stop Monitoring CAN also include remote Daughter Engines. When installing the Daughter Engine, simply select ONE of the Non-Stop Motors as the primary engine. When the Daughter connects for the first time, it will download configuration data and will self-configure itself to work with ANY of the Non-Stop motors, even if one of them fails.

Mixed Architecture

In some situations, the Network Administrator may need to deploy Argent in a mixed environment.

For example, this company has a main office in downtown Chicago, a regional office at O'Hare Airport (connected via a DS3 line), and a remote office in New York and Houston, connected via 64K and T1 lines, respectively.



Note the O'Hare office has an extremely stable method of connectivity. The easiest and most effective way to monitor this office is to deploy a Monitoring Engine using Named Pipes.

Houston also has a high level of bandwidth. However, there is a firewall between Houston and downtown Chicago, so TCP/IP is used.

Lastly, New York has a limited 64K frame-relay link. Here the Mother/Daughter Architecture is used. This is what the network traffic looks like in this situation:

UNIX Monitoring

Argent offers SEVEN options for monitoring Unix systems:

Telnet
SSH
Linux Secure Agent
Linux Shell Script Agent
SSH Relay Agent
Unix Rule Engine
Unix Daemon

Each of these options provides different features and benefits.

For example, Telnet and SSH are agentless options; nothing needs to be installed on the monitored server. Telnet communication, of course, is completely plain-text, making it not secure. SSH, on the other hand, is completely encrypted from end to end, making it secure.

The Linux Secure Agent and the Linux Shell Script Agent can be easily installed on each monitored server, providing a dedicated communication channel.

The Linux Secure Agent is a binary executable.

The Linux DAEMON is similar to the Secure Agent, but does not require INETD.

The Linux Shell Script Agent is just that -- a shell script.

This allows for user customization and for use on systems where a Linux Secure Agent binary does not yet exist.

The SSH Relay Agent shifts the SSH connection workload from a Windows monitoring engine to a Unix machine, which generally make SSH connections more efficient. A Windows monitoring engine sends the Rule (a Unix shell script) to the SSH Relay Agent, which then makes the connection to the monitored servers via SSH, executes the Rule, then sends the information back to the monitoring engine.

The Unix Rule Engine retrieves its marching orders from the main Argent server, makes SSH connections to

UNIX Monitoring

the monitored servers, and sends back the results to the main Argent server. This provides additional fault tolerance, as well as removing the connection load from a Windows monitoring engine. Since each Unix Rule Engine initiates the connection to the main Argent server, firewall configuration is also simplified, since you only have to open one port.

UNIX Secure Agent (LINUX Example)

As an alternative to Telnet and SSH, you can install a local Linux monitoring agent on the Linux systems you wish to monitor.

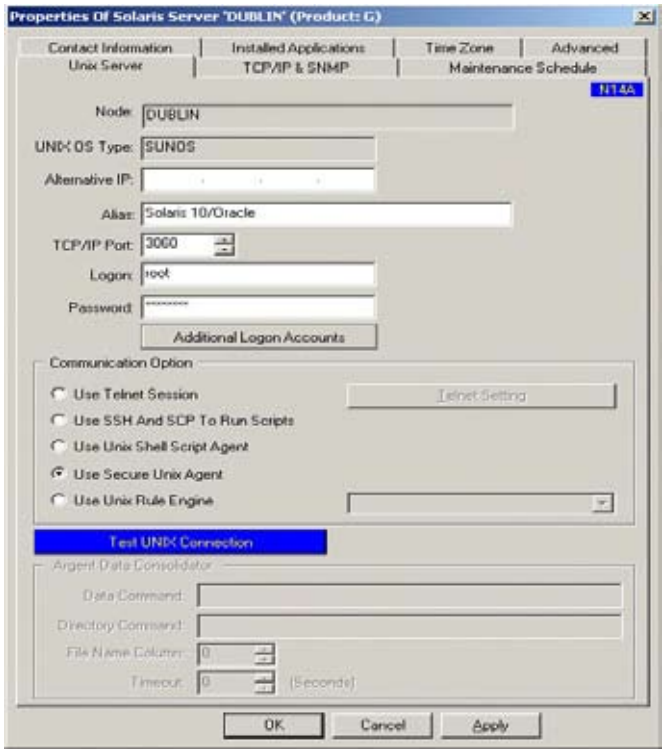
Argent's Secure Linux Agent provides a dedicated communication channel for transferring the Rule (a shell script) to the monitored system and executing it. The local monitoring agent can also, for example, reduce the overhead required by SSH logins.

The Secure Linux Agent is a compiled binary executable. It is available for AIX, HP-UX, Linux, and Solaris.

The Unix Communication Option must be set in License Manager for the Argent Extended Technology product. While the following examples are for the Argent Guardian, the instructions are the same for each product.

To set the communication method, go to Administration section of the Argent Guardian, then select License Manager, and the Licensed Servers tab.

Locate the Unix server in the Server/Device column, then double-click the server name, then double-click the server name, or select it, then right-click and choose Properties.



UNIX Secure Agent (LINUX Example)

Select the Use Secure Unix Agent. By default, the Secure Linux Agent listens on TCP port 3060. If your monitoring agent isn't on the default port, change the TCP/IP Port field to reflect the proper port.

Enter the username in the Logon field, and the password in the Password field and click OK.

Installing an SSH Relay Agent

The SSH Relay Agent enables agentless monitoring of your Unix machines using SSH.

Install packages are available for download for the following UNIX platforms:

SunOS 2.8 (Solaris 8) or later on SPARC hardware

SunOS 5.10 (Solaris 10) or later on Intel hardware

HP-UX 11.x or later

AIX 4.3 or later

LINUX 2.4.7 or later

To install the SSH Relay Agent, do the following:

1) Copy the installation tarball to the UNIX host

2) Create /etc/argent by typing:

mkdir /etc/argent

3) Move the tarball into /etc/argent by typing:

mv ARGENT_GUARDIAN_SSH_RELAY_ver_yymm.TAR /etc/argent

4) Change to the /etc/argent directory by typing:

cd /etc/argent

5) Extract the tar file by typing:

tar xfv ARGENT_GUARDIAN_SSH_RELAY_ver_yymm.TAR

6) Create an empty tag_relay.log file by typing:

touch tag_relay.log

Installing an SSH Relay Agent

7) Ensure the user running the SSH Relay Agent has Read and Execute permissions on tag_relay and run_ssh

Type the following:

chmod 755 tag_relay

chmod 755 run_ssh

(The user running the SSH Relay Agent must be the owner of these files, or at least be a member of the specified group.)

*8) Ensure the user running the SSH Relay Agent has read and write permissions on log files. Type: chmod 666 *.log*

9) Verify the location of scp and ssh on the system. By default, these are /usr/bin/scp and /usr/bin/ssh, respectively.

Any scp and ssh executables can be used. If not using /usr/bin/scp and /usr/bin/ssh, edit run_ssh to change the absolute path to these two executables.

10) Insert the following line into /etc/services:

tag_relay 3062/tcp # The Argent SSH Relay Agent

Installing an SSH Relay Agent

Note: The SSH Relay Agent may be configured to listen on any TCP port. If desired, change the port number in the above line in /etc/services

11) Insert the following line into /etc/inetd.conf, replacing [USER] with the user account that will run the SSH Relay Agent:

tag_relay stream tcp nowait [USER] /etc/argent/tag_relay tag_relay

12) Restart inetd

Note: To secure the SSH Relay Agent, use TCP wrappers.

13) Add the following line to /etc/hosts.allow:

tag_relay: [allowed IP addresses or hostnames]

SSH Key Exchange

In order for the Argent SSH Relay Agent to communicate with the monitored Unix systems using SCP and SSH, a proper SSH key exchange must first take place.

This includes saving the key fingerprint of the monitored system, and transferring the public key from the SSH Relay Agent host to the monitored system.

By default, SSH tries to authenticate with the private/public key pair first. If this is unsuccessful, SSH then requests the password be entered manually.

These steps will allow the SSH Relay Agent to communicate with the monitored systems via SSH without a password. This allows the password for the account used to change periodically without having to update Argent.

1) *Logon to the SSH Relay Agent host as the account that will run the agent. Alternatively, logon to the host and type:*

   su - [USER]

2) *Create the authentication keys by using ssh-keygen. RSA, DSA, or SSH1 keys can be generated with this utility.*

   To generate a DSA key, for example, type:

   ssh-keygen -t dsa

This will create id_dsa and id_dsa.pub. (The latter file is the public DSA key.)

3) *Copy the public key from the SSH Relay Agent host to each host to be monitored, appending it to $HOME/.ssh/authorized_keys for the account that runs the Argent agent.*

If $HOME/.ssh does not exist on a monitored host, use ssh-keygen to create the host keys first. Use SCP to transfer the public key file.

SSH Key Exchange

4) If $HOME/.ssh/authorized_keys already exists, transfer the public key file to the monitored server as a different name, then append the contents of that file to the existing authorized_keys file.

5) If prompted to save the host key of the monitored server, answer "yes". This will permanently save the key.

6) Test the SSH connection from the SSH Relay Agent by typing:

ssh [servername] hostname.

The Argent Unix Daemon

Previously the Argent Guardian UNIX Agent was a simple command line program that runs under INET or XINET. The program could read/write to standard I/O while making use of INET/XINET to talk to TCP sockets.

Customers with enhanced security may not allow INET/XINET on UNIX machines.

One workaround is to use the UNIX Rule Engine. The UNIX Rule Engine pulls the rule scripts and task schedule from the Argent Guardian main engine. The Argent Guardian main engine cannot directly contact UNIX Rule Engine. This makes relator testing difficult to implement.

The Argent Guardian UNIX Agent Daemon is used to address the issue.

The Argent Guardian UNIX Agent Daemon does following:

1. Daemon starts from system init.
2. Daemon runs as a TCP server listening to a configurable port (default 3060)
3. Driven by the Argent Guardian monitoring engine.
4. All communication protocol is compatible with existing Argent Guardian UNIX Agent.

iSeries (AS400) Monitoring Engines

To install the Argent Guardian Agent for iSeries, you need to sign on as the system security officer.

This is the QSECOFR user profile. The installation must be performed under this user profile.

You will also need to be familiar with manipulating iSeries save file objects and File Transfer Protocol (FTP).

Before beginning the installation procedure, ensure no Argent Guardian Agent iSeries processes are active.

If an Argent iSeries agent is already installed, stop the processes by following these steps:

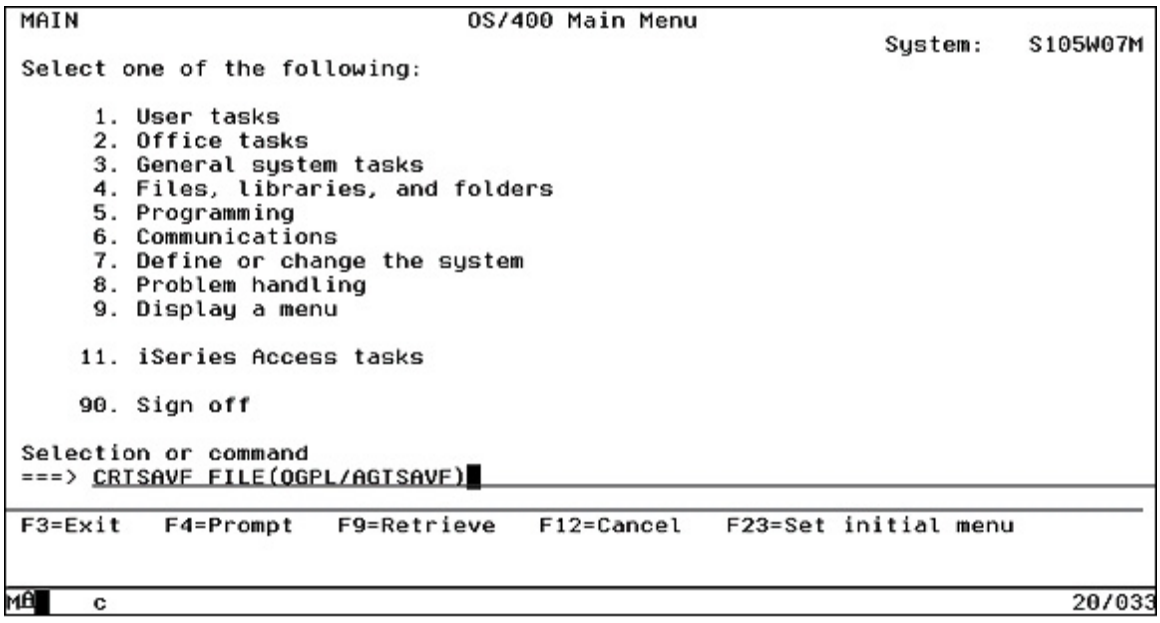
1. ADDLIBLE ARGENT
2. ENDARGAGT
*3. ENDSBS ARGENT *IMMED*

The first thing to do is to create a temporary save file in a suitable library.

QGPL is a library often used for general-purpose tasks like this.

You can create a save file using the CRTSAVF command as shown below

CRTSAVF FILE(QGPL/AGTSAVF)



If this results in a message indicating the file already exists, enter

DLTF FILE(QGPL/AGTSAVF)

to remove the previous file.

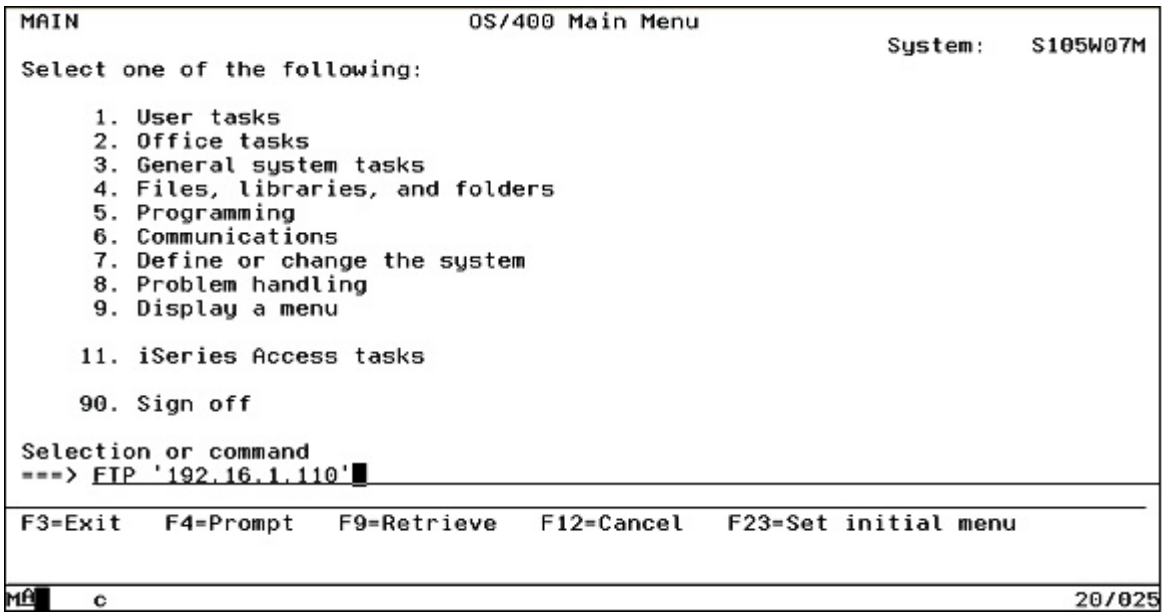
Depending on where in the network the Argent Guardian Agent for iSeries distribution kit is located, you may need to adapt the general instructions that follow.

You can use FTP to retrieve the distribution kit from another iSeries, an Argent distribution media server, or a location on your network where an administrator has placed the distribution kit.

You can also use a Windows FTP client to send the distribution kit to the iSeries. This example helps you retrieve it to the iSeries from a network server in your enterprise.

You can use the FTP command from any iSeries command line to invoke FTP.

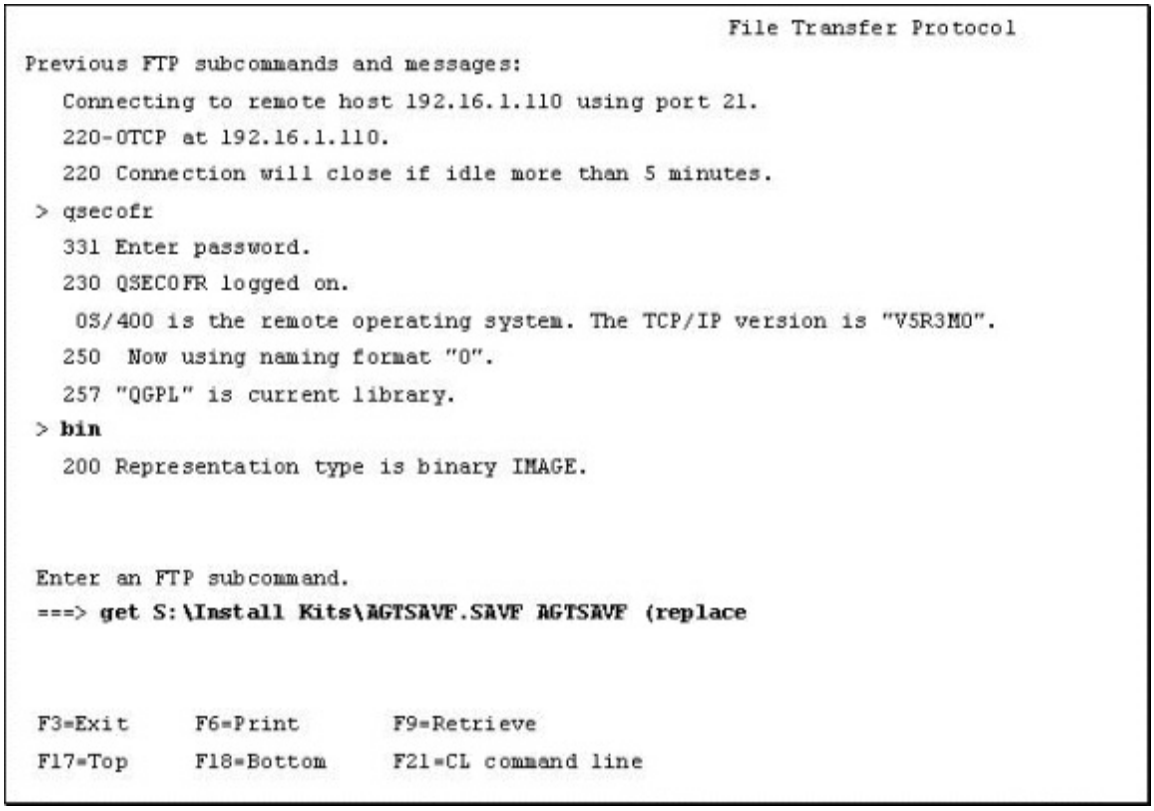
Example:



When FTP starts, it provides a set of prompts that help you complete the process of retrieving a file. Overall, the most important things to remember are:

1. The bin FTP sub-command must be used to preserve the save file characteristics of the distribution kit,
2. You must receive the distribution kit into the save file you created in the previous step, and
3. You must specify the (replace parameter on the FTP get sub-command.

The general flow of your FTP dialogue will look similar to the following screen:



After the distribution kit has been successfully downloaded into the save file, you need to create a library named ARGENT.

As of this version of the product, ARGENT is required as the name of the installed program product library.

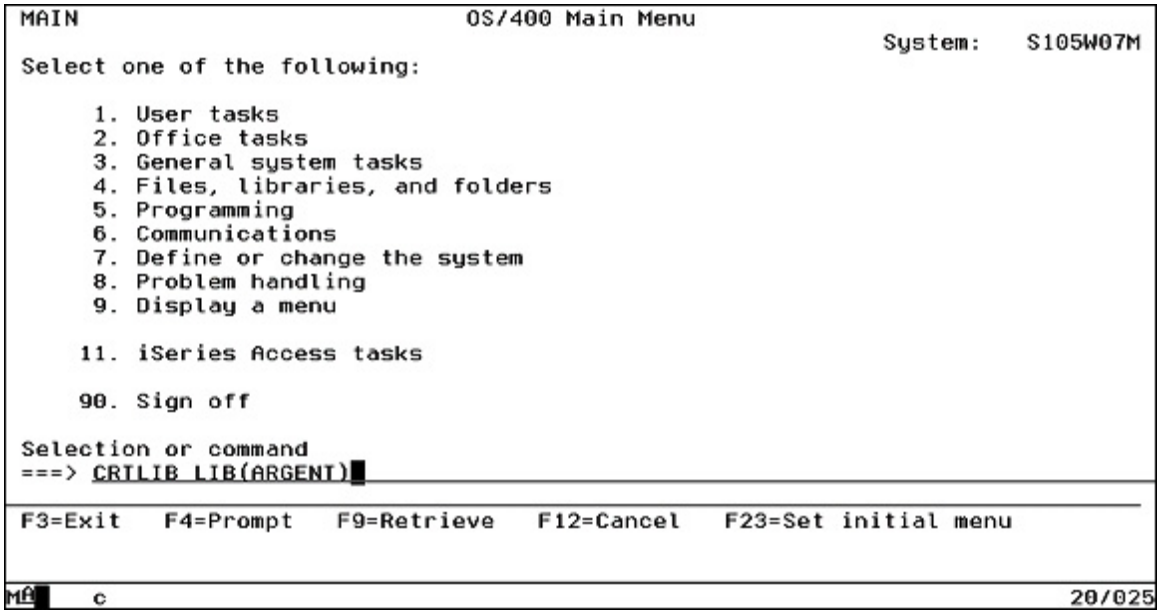
If the ARGENT library already exists from a previous installation, use the

CLRLIB LIB(ARGENT)

to prepare it for a new installation. The CRTLIB command is not needed.

If you are installing the Argent Guardian Agent for iSeries for the first time, create the library using

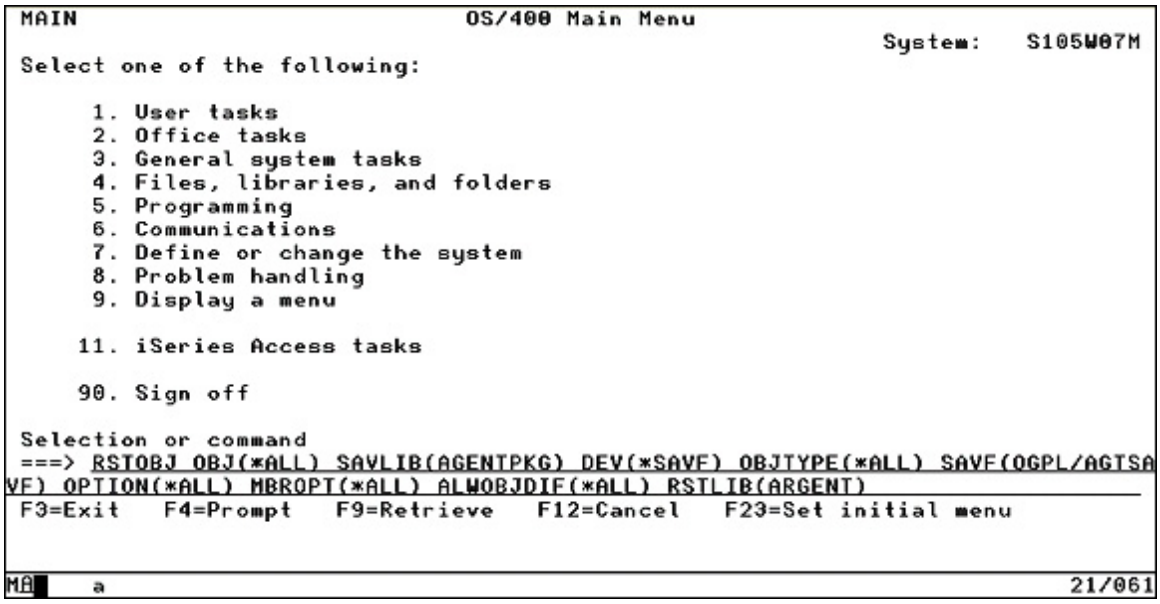
CRTLIB LIB(ARGENT)



After the ARGENT library has been created, you need to restore the contents of the distribution kit into the library you just created.

You can restore the contents of the distribution kit using

*RSTOBJ OBJ(*ALL) SAVLIB(AGENTPKG) DEV(*SAVF) OBJTYPE(*ALL) SAVF(QGPL/AGTSAVF) OPTION(*ALL) MBROPT(*ALL) ALWOBJDIF(*ALL) RSTLIB(ARGENT)*



The system will respond with an informational message about the results of the RSTOBJ command.

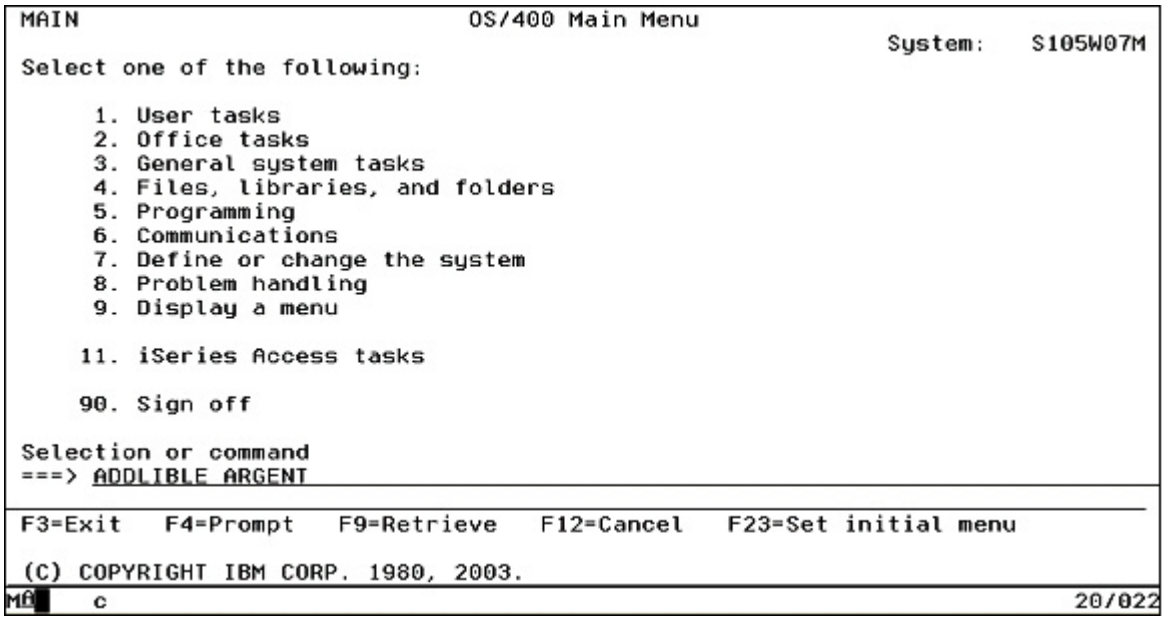
After the RSTOBJ command completes, you should see a message near the bottom of your display terminal session that reads as follows:

```
34 objects restored from AGENTPKG to ARGENT.
```

Once the objects have been successfully restored into library ARGENT, ensure the ARGENT library is on the library list for your display terminal session.

You can add ARGENT to the library list using the ADDLIBLE command as shown below

ADDLIBLE ARGENT



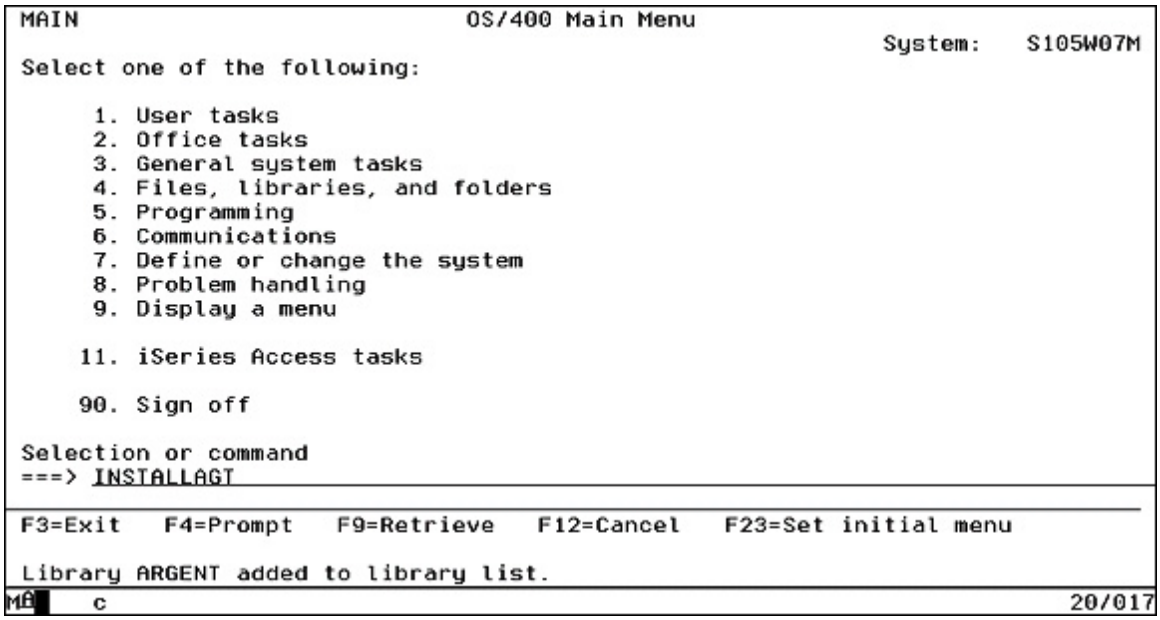
The system will respond with an informational message confirming that ARGENT has been added to the library list. The message should look similar to the following screen:

```
Library ARGENT added to library list.
```

Once ARGENT library is on the session library list, you will need to complete the installation of the Argent Guardian Agent for iSeries product using

INSTALLAGT

iSeries (AS400) Monitoring Engines



The INSTALLAGT command runs silently for a short duration, typically 20 or 30 seconds.

After the INSTALLAGT command completes successfully, the Argent Guardian Agent for iSeries has been fully installed on the system.

TCP/IP Ports Used by Argent Products

For a current listing of TCP/IP ports used by Argent products, please go to http://help.Argent.com