TOC

About This Guide

This is an archived copy of the v23 documentation provided for Logi Info v23.3 and its service packs.

Notice: Archived Documentation

This documentation is provided as a courtesy reference for a version of our software that is no longer under active development or support. The information contained herein is offered without warranties of any kind, either expressed or implied, including but not limited to warranties of accuracy, completeness, or fitness for a particular purpose.

While this archived material may assist with understanding historical functionality, please be aware that the software described is no longer maintained at this version level and may contain outdated or inaccurate information. Images may not reflect currently supported modules, support sites, or third party products. This software may not be compatible with current versions of previously compatible third party products.

To access and upgrade to current software solutions and receive ongoing support, please contact our customer support team. They can assist you in migrating to the latest appropriate software version that meets your needs. Our support team is available to help ensure a smooth transition to actively maintained alternatives that provide the functionality and reliability you require.

System Requirements - Logi Info

This topic presents the general hardware and software requirements use of Logi Info and the products and data sources it supports.

The following topics discuss requirements and recommendations for installing Logi Info v23.1:

- Sample Architecture
- Sizing for Performance
- Scaling Logi Info Applications
- Server Virtualization
- Container Deployments
- Cloud Deployments

General Requirements

The following general hardware and software components are supported by the latest releases of Logi Info.

Logi add-on modules and related products may not support all data sources and supporting software listed below; refer to their introductory documentation for details.

Browsers	- Chrome, Firefox: all current public versions (Certified) - Microsoft Edge: all current public versions (Certified) - Microsoft Internet Explorer: 11 (Certified through v14.1); see notes below about earlier versions - Safari, Opera: all current public versions (Supported)

Hardware	Development, QA/Test, Production, and Disaster Recovery servers: - Modern, high-speed CPU, with 4 cores, minimum - 8 GB RAM, minimum - 100GB storage device - 2 GB free disk space minimum (with .NET or JDK already installed) See "Sizing for Performance" on page 22 for more information.
Operating Systems	64-bit versions of the following are supported: - Windows Server 2019 (v12.7 + only), 2016 (v12+ only), 2012 R2, 2008, 2003 - Windows 10 (Logi Info v12+ only, all editions except RT) - Windows 7-8 (all editions) - Red Hat (7,8) Ubuntu (latest version), CentOS (7, Stream8), SUSE, and most other flavors of Linux
Web Servers	- Internet Information Server (IIS) 7x (support ending 2023), 8, 8.5, 10 - Apache-Tomcat 8 (without Tomcat FHS), 8.5, 9.0, 10.0 - JBoss EAP- 7 - GlassFish (SJSAS) 2.1, 3.0, 3.1, 4.0, 4.1 - WebLogic 12.2, 14.1 - Websphere 8.5, 9.0 - WildFly 8, 9, 10
Supporting Software	Required for all development work, using Logi Studio: - Microsoft .NET Framework 4.6.2+ Required for development and execution of Logi .NET applications on Windows servers: - Microsoft .NET Framework 4.6.2+ Required for development and execution of Logi Java applications on Windows or Linux servers: - OpenJDK 8, 11, 12 (see note below regarding 11+), 13, 14, 17

Oracle has changed its Java usage policies - see "Java Usage Policy" on page 342 for important information.

Data Sourcesserver

Logi Info can connect to, use data from, and in most cases write back to the following data sources: - DB2 database

- Files: JSON, XML, Excel, CSV
- Google Docs, Google Maps
- HP Vertica
- HPCC
- JDBC-compliant database servers
- Microsoft SQL Server database server
- Microsoft SQL Server Analysis Services (OLAP)
- MongoDB (up to and including v2.6, but *not* later versions)
- MySQL database server (excluding v5.5)
- ODBC-compliant database servers
- OLEDB-compliant database servers
- Oracle database server (up to and including 19c)
- PostgreSQL database server
- Progress OpenEdge database server
- Salesforce.com
- Twitter.com
- Web Services (REST and SOAP) With the addition of supporting products such as Logi DataHub, many other online commercial data sources can be accessed. See each product's System Requirements or introductory documents for more information.

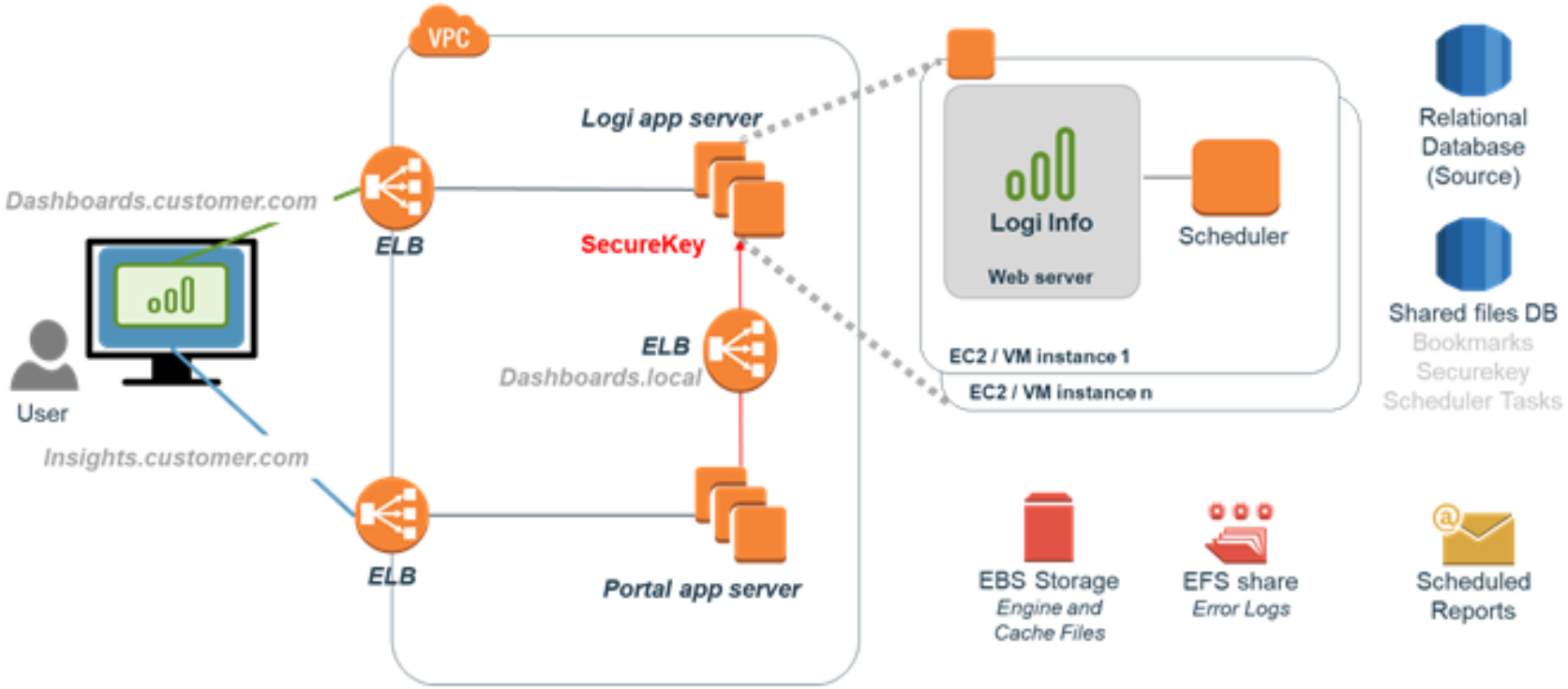
- IIS must be installed *before* installing Logi products. Logi .NET applications on Windows and Logi Studio require the .NET Framework 4.6+. If not already in place, with your consent, appropriate versions of the .NET Framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.
- More information for Linux developers about Logi Java applications is available in "About Logi Apps and Java" on page 332.
- While Internet Explorer versions 7-10 are certified for use with Logi Info v12.x, Microsoft has ended support for them. Microsoft has also ended support IE 5 and 6. You should use Microsoft Edge for viewing applications built with Logi Info v12.x and later.
- Microsoft ended support for .NET Framework 4.5 in January 2016. .NET 4.6 is required for Logi .NET applications 12.6+.
- Use of Java 11+ (supported in Info v12.6 SP2 and later) requires special configuration - see "Java Server Configurations" on page 346 for more information.


Adobe no longer supports their Flash Player, and has blocked Flash content from running in a Flash player since January 12, 2021. Select this link for more information.

Sample Architecture

The architecture below represents a deployment of Logi Info and the Logi Scheduler service to AWS. In this example, the Logi Info application and the parent application in which it's embedded are hosted in a Virtual Private Cloud behind a load balancer. EC2 instances hosting the Logi Info application can be spun up or down according to user demand.

This example also assumes that shared resources (such as Bookmark files and SecureKey references) are managed in a database accessible to each EC2 instance. An alternative approach is to configure a network share in place of the database.



Sizing for Performance

The following are general sizing considerations for a Logi Info web application, followed by high level recommendations for server components based on these considerations. Specific recommendations will vary depending on your implementation.

To accurately determine the appropriate deployment model for your Logi Info applications, take the following into consideration:

- **Maximum Number of Concurrent Users** - As with any system, the application needs to respond to end users in a timely fashion, even during peak load times.
- **Number of Concurrent Data Visualizations** - The number of visualizations that are loaded concurrently into a page or Dashboard will affect performance.
- **Complexity of Data Visualizations** - Some data visualizations present a single metric and dimension, while others present multiple metrics and/or dimensions. More resources are required to render complex data visualizations.
- **Volume of Data Used to Generate Data Visualizations** - Logi Info data visualizations range from simple, summary charts to interactive, self-service analysis tools. The volume of data required to be delivered from the data tier to the application tier to generate these visualizations can range from a few dozen records to tens of thousands of records.
- **Logi Info-Based Data Aggregation or Manipulation** - Whenever possible, data manipulations, such as calculations and aggregations, should be performed within the data server or system because they're optimized to perform these operations. In addition, the amount of data transferred over the network from the data system to the Logi application should be limited to the actual needs of the data visualization. When it is necessary to perform data manipulations in the Logi application, the type of manipulation and the volume of required data will have an impact on the resources available.

Recommendations

Based on load-testing of benchmark data from our Large Enterprise customers, we have established the following recommendations:

Processors

The following are guidelines for a typical scenario.

- **Minimum Configuration Cores**: The minimum number of required CPU cores is determined by the expected number of end users, as follows:

End Users	Baseline # Cores
1 - 100	4
101- 250	8
251 - 500	12
501 - 1000	16
> 1000	Custom Sizing Required

Using the minimum configuration as a starting point, the following factors will determine the processor requirements for your production environment:

- **High Concurrency** - If you anticipate end-user concurrency of greater than 10%, increase the number of cores by 25% to 50% of the Minimum Configuration Cores.
- **Application Complexity** - If you judge that your application is complex, based on the considerations presented earlier (e.g. high number of visualizations in a single page, complex workflows, and complex data processing at the app tier), we recommend that you assess the impact of that complexity on overall performance and sizing. It is not unusual to add an

additional 25% to 100% of the Minimum Configuration Cores to ensure better performance.

- **High Availability** - Load-balanced, high-availability systems will require an increased core count. It is typical for Large Enterprises to deploy an additional 50% to 100% of the Minimum Configuration Cores based on the target service level.

Memory

We recommend a 2-to-1 ratio of gigabytes of memory to CPU core (i.e. two GB of RAM per core).

Storage

Logi Info applications are not memory-constrained because they leverage a hybrid caching (see *The Logi Server Engine*) mechanism that utilizes both disk- and memory-based storage when performing data acquisition, data aggregation, visualization generation, and other data-intensive operations.

Using the mechanism, a Logi application dynamically allocates storage resources depending on the processing stage in order to optimize their use. Therefore, we recommend that the server utilize dedicated, "fast" storage devices, such as SSDs or high RPM disk drives. Total storage usage is highly variable and is dependent on the size of the data volumes being processed and cached by the application. At a minimum, drives with a capacity of 100GB are commonly used.

Environments

In addition to the primary production environment, other environments should be appropriately sized as well:

- **Development Environments** - We recommend a four core minimum for any development environment. For environments with sophisticated development needs, such as a high number of developers, continuous integration development style, and test driven development, it's typical to size a development environment at 25% to 50% of the baseline number of cores,

with a minimum of four cores.

- **Quality Assurance and Load Testing Environments** - Requirements for load testing or automated QA practices typically require a mirror images of the production environment, with the same number of cores in the same machine configuration, in order to accurately replicate production performance characteristics. If load testing is not required, then 25% to 50% of the production core sizing is ideal for more complex environments or automated QA, with a minimum of four cores.

- **Disaster Recovery Environment** - Some large enterprises require a stand-by system for disaster recovery. In this scenario, the Disaster Recovery system is typically a mirror image of the full production environment.

How Many Servers Do You Need?

Your production web server hosts your Logi applications, which run as extensions to the web server. Ideally, this computer should be dedicated to this task alone.

However, other configurations are possible, including those in which the web server also serves other functions (i.e. is not dedicated to Logi applications alone) and/or is also the database server. Some Logi large enterprise customers use multiple "clustered" servers for top performance and high reliability.

Numerous factors external to your Logi application can affect this decision, including the amount of general web server traffic, the number of concurrent database users, the size of the databases, the complexity of the database queries, the frequency of access, and, not least, cost.

You may care to begin with a combined configuration and, as your report usage grows, change to a dedicated configuration. The nature of Logi products allows you to do this easily and often without additional cost (if CPU core count remains constant).

Recent studies concerning server virtualization suggest that database servers are frequently under-utilized. On the other hand, many database vendors recommend that their products be run on a dedicated server. You may wish to check with your database vendor for their recommendations concerning database servers.

Scaling Logi Info Applications

Logi Info can be deployed to clustered environments on premise or in the cloud. This topic outlines how to prepare a Logi Info deployment for scaling with user demand and greater data volumes over time.

Considerations

While Info applications can adapt for vertical or horizontal scaling, Logi recommends a horizontal scaling approach. The following items are crucial for planning a Logi Info deployment for horizontal scaling:

1. **Sticky Session Configuration is Recommended** – Some application features function best in a load-balanced environment in which user sessions are "sticky" to the server on which they originate.
2. **Some Application Resources are Shared** – Saved Dashboard configurations (known as Bookmarks) and other key items, such as reference files generated by Logi's SecureKey SSO method, should be stored on a network share or in a database accessible to all servers in a cluster hosting the Logi Info application.
3. **For Autoscaling, Scheduling is Recommended** - Common patterns include usage surge over weekday mornings when users get into office and nightly scheduled report generations. Multi-threaded requests increase CPU usage, and can result in unnecessary VMs being provisioned when using an auto-scaling algorithm.
4. **Embedding** – If using the Logi Embedded API, the parent app should provide the load balanced URL to client browser.
5. **Scheduling** – Logi Scheduler is a standalone service that can be scaled independently. However, typical recommendation is to install one scheduler per production environment.
6. **Upgrading** – When upgrading Logi info, all load balanced environments will need to be updated.

Configuring for Load Balancing

The following checklist can be used when deploying to a node in a load balanced environment for the first time. This process can be automated as part of a larger Dev/Ops process to increase productivity.

- **Prepare Server(s)** – Install preferred IIS or Java web server container. Size servers for planned load and verify that each node is networked to other environment dependencies including data sources and file shares.
- **Install add-on services like Logi Scheduler** – Where scheduling capabilities are required, the Scheduler service will need to be installed separately.
- **Deploy Logi Info application** – Promote the application and relevant support files from a staging environment.
- **Configure Settings** – Some Logi application settings are environment specific such as license file location, IP ranges for SSO configuration and potentially data source connections. Values for these settings can be dynamically set in an automated process.

For more information, see *Load Balancing Configuration*.

Server Virtualization

Many organizations are using server virtualization to maximize hardware usage and reduce costs.

Server virtualization products allow the assignment of CPU resources to processes. This may take the form of a maximum percentage of combined CPU utilization, or as specific allocation of logical CPUs, to a virtual machine (VM). The server administrator is responsible for making these configuration decisions. Logi Analytics' product licenses treat a VM just like a regular, non-virtualized server.

In order to ensure good performance in any virtualized server environment, administrators must be careful to allocate appropriate resources to VMs.

It's not uncommon to relocate a VM from one hardware platform to another, for example, for hardware maintenance. The Logi license will "move" with the VM, as long as the machine name remains unchanged.

Container Deployments

The Logi Analytics Platform and Logi applications can be bundled into and executed from container environments, such as Docker. Some of our customers have had success deploying their Logi applications in this manner.

Logi Professional Services staff may be able to assist you with such a deployment, for a fee. However, we don't recommend any particular container over another and we don't certify that Logi applications will work using a container.

Docker Instructions

A best practice for Docker is to separate the Tomcat container and Logi Scheduler into their own nodes behind a load balancer (for information about load balancing, see *Load Balancing Configuration*).

1. Copy Logi Info App to Linux host (follow configuration steps in Deployment Checklist below for details)
    - Must have security configured with users, sharing rdSecurekey configured for multiple nodes
    - Must have Data connections set up for production configuration
    - Must have rdError sharing set up for multiple nodes (complete with custom error page)
    - Must have an OEM license
    - Scheduler connections setup to work behind a load-balanced end point
2. Install Docker
3. Pull Tomcat image from Docker Hub

4. Run Tomcat container and enter shell to install Scheduler
    - `Docker run -it Tomcat bash`
    - Install Scheduler
    - `From a second shell commit changes to Tomcat Docker image as Tomcat`
5. Create Dockerfile to buil dimage that includes Logi App and Scheduler content
    - Transfer Info app to container
    - Transfer script that starts both Scheduler and Tomcat
    - Run start-up script(s)
    - Expose the ports required by Tomcat and Scheduler
6. Build a new image (you must do this every time you make a change to the Info app)
    - `Docker build -t Tomcat`
7. Run Tomcat

At this point, you will have a container with your Logi app working on port 8080. The following steps show you how to use Docker-compose to stand up multiple instances of Info and load balance them with Nginx:

8. Write docker-compose.yml file
9. Run Docker-compose
    - `docker-compose up -d`
10. Scale Info up:
    - `docker-compose scale app=2`

Useful Links

The following links are resources for implementing Docker:

- How to maintain Session Persistence (Sticky Session) in Docker Swarm
- Swarm 1.12 Routing data to specific container sticky session
- Docker rm

Nginx:

- Docker Example Nginx Tomcat MySQL
- Bitnami/Tomcat
- Docker-compose scale with sticky sessions

Multiple executables in container script:

- Run multiple services in a container

Bind Mount:

- Use bind mounts

Dockerfile Example

Below is a file that accompanies the directions mentioned in the Docker-compose scale with sticky sessions link above. This example is not meant to be run, it is simply a guide for creating your own:

```
FROM tomcat

MAINTAINER Author Name <david.abraham@logianalytics.com>

ADD /InfoGo /usr/local/tomcat/webapps/InfoGo

ADD logiStart.sh /usr/local/tomcat/logiStart.sh

#CMD ./usr/local/tomcat/logiStart.sh

CMD ["/usr/local/tomcat/logiStart.sh"]

EXPOSE 56982

EXPOSE 8080
```

Docker-Compose Example

Below is an example of a docker-compose file:

```
app:

image: tomcatdiscovery

environment:

-VIRTUAL_HOST=localhost

-VIRTUAL_PORT=8080

-USE_IP_HASH=1

volumes:

-/home/logise/Docker/share:/usr/local/tomcat/webapps/InfoGo/share

nginx:

image: tpcwang/nginx-proxy

ports:-"80:80"

volumes:

-/var/run/docker.sock:/tmp/docker.sock:ro
```

Cloud Deployments

Cloud-based hosting services, such as Amazon Web Services (AWS), can host Logi Platform components and Logi applications, though there can be some limitations depending on service features. Our customers have had success deploying their Logi applications to cloud-based services.

We recommend hosting Info Platforms on general purpose VMs hosted by cloud providers. For example:

- AWS - M5.xlarge EC2 instances - https://aws.amazon.com/ec2/
- GCP - N2 Compute Engines - https://cloud.google.com/compute/
- AZure - Dv3 Virtual Machines - https://azure.microsoft.com/en-us/services/virtual-machines/

For a representation of a deployment of Logi Info to AWS, see the example in Sample Architecture.

Recommended minimum environment to host Logi Info applications:

- VPC with multiple Availability Zones (required for load balancing)
- Two EC2 M5.xlarge (4-core, 16Gb RAM, 50 Gb EBS Internal storage)
- EFS Storage (5+ Gb) for storing Error logs.
- RDS (Microsoft SQL, MySQL, PostgreSQL, Oracle) with 10 GB+ storage to share information between server instance and to persist user state, including
  - security handshake information
  - user bookmarks
  - user activity logs (using Event Logging, which developer defines in Info application)
- Application Load Balancer (ELB) with Sticky session

If you'd like to try deploying your Logi Info application to AWS by yourself, this blog post from dbSeer may be useful. It includes a link at the bottom to step-by-step instructions.

We do not provide support related to this third-party blog post.

Otherwise, Logi Professional Services staff may be able to assist you with such a deployment. However, we don't recommend any particular service over another.

Choosing EC2 Over Other Services

Logi Info platform profile:

1. Logi Info based apps are based on a classic, monolithic architecture
2. Application combines multitude of features beyond HTML generation, e.g. PDF exporting, which rely on many additional services and libraries
3. Application depends on very high file I/O
4. Application depends on web server capabilities
5. Sticky Load Balancing strategy is recommended due to high file I/O use-cases

Virtual machines are a recommended approach since they support flexibility required for above profile.

AWS provides EC2 as a flexible VM solution. AWS provides command line interface to create and manage EC2 based infrastructure, which allows for integratinginto existing scripting solutions, e.g. Ansible, Chef, etc.

AWS Lambda

AWS Lambda is a serverless environment. There are restrictions on how application can be managed or deployed. Also, it may impact Logi Info platform's capabilities, e.g. PDF generation.

AWS Container Service

Docker containers are a great alternative to manually defining server environment. More information on packaging Logi Info platform in containers is covered in a separate document.

Benefits of using containers include:

- Developers can control the application environment; DevOps have limited interaction
- Containers can be deployed to other cloud providers with limited reconfiguration

Drawbacks with containers, compared to EC2 VMs:

- For non-OEM licenses, provisioning licenses based on host name of instances can be challenging
- Containers can run only single services. Scheduler service (optional) will need to be deployed separately and networked with Info platform. This will require additional configuration.

Deployment and Auto-Scaling

AWS Beanstalk

Beanstalk makes it easier to provision EC2 instances and deploy code. Some concerns about lack of control over resources provisioned and monitoring state.

AWS Cloudformation Template

Users access Logi Info application through customer's existing web application. Logi Info application is embedded into existing parent application. The application is dependent on customer's existing infrastructure architecture. We do not have templates to support different scenarios. With above recommendations, customers can update any existing templates to provision appropriate infrastructure to host Logi Info application.

Auto-Scaling Strategy

Majority of our customers use scheduled scaling strategy to support known usage patterns. Common patterns include usage surge over weekday mornings when users get into office, nightly scheduled report generations.

Logi Info application has built-in capability to multi-thread individual user requests for Dashboards and other analytical capabilities. This optimizes resource usage and speeds up delivery. Multi-threaded requests spike CPU usage. Most Auto-scaling solutions monitor CPU usage to provision and scale up additional VMs. Auto-scaling algorithms can clash with the usage and unnecessarily provision VMs. Hence, the scheduled scaling approach.

Product Licensing

All Logi products are *licensed* to customers using legal documents executed as part of the sales contract. Most products require the issuance of a license *key* (stored in our licensing system) and the presence of a license *file* on the system where the product and/or application is installed.

The following topics discuss the licensing scheme for Logi Info and its add-on modules, as well as the most common license-related situations you're likely to encounter:

About Logi Product Licensing

Logi Analytics licenses are **server-based,** rather than individual-user- or concurrent-access-based, so an unlimited number of end-users can access Logi reports through a single web server. Our Logi Info licensing scheme allows you to install our product on one development machine and one production server.

Logi Info requires a license for both our development tool, **Logi Studio**, and for our **Server Engine**, which is part of each Logi application you create. Your development system will likely require both but production web servers generally do not need Studio installed on them. So, for Logi Info we offer both a "Studio and Development Server" combination license and a standalone Server license, and this two-license approach allows flexibility in different situations.

Below is a list of the Logi Info versions and their license file requirements:

Product	License Requirement
Logi Info v23.1	Requires Logi Info v14 license file
Logi Info v14.0-14.1	Requires Logi Info v14 license file
Logi Info v12.0-12.8	Requires Logi Info v12 license file

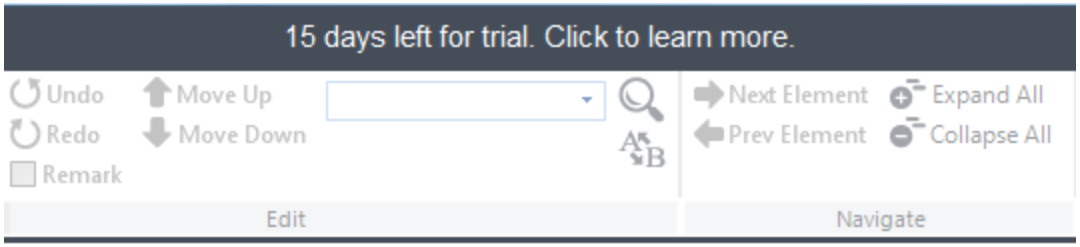
Major Version Compatibility

v23.1 You no longer need a new license when upgrading an existing installation of one of our products from one major version to another major version, for example v14 to v23.1. However, if you are upgrading from v12 to v23.1, *you will need a v14 license key and file* for the new product/application. If you have an active maintenance contract with Logi, we've already created licenses for your license point of contact, which you can find on Devnet by selecting **Support** > **License Manager**. Contact Customer Service if you want to upgrade and need a new license.

If you upgrade to a newer version of Logi Info you do not need a new license for the add-on products, such as Discovery Module and DataHub. For more information about Discovery Module and DataHub licensing, see the corresponding section in "License Keys and License Files" on page 42.

Built-in Trial License

Logi products come with a built-in **15-day trial license**. You don't need to do anything but install the product and you can begin using it immediately. For Logi Info, aclearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the Studio license counter-down display will take you to a web page that offers information about *purchasing* a product license.

After the Logi Info 15-day trial license period expires, unless you have a regular license file in place Studio will no longer start up and any Logi reports you've developed will *no longer run*. An expired trial license will cause other Logi products to *block access* to certain features and/or prevent you from *saving* your work.

License Keys and License Files

When you purchase a Logi product, here's what happens in the licensing process:

1. We create a **license key** for each product you buy, specific to you or your organization, in our licensing database.
2. You log into DevNet and visit the **License Management** page to create a **license file**. You do this by *assigning* one of your license keys to a specific computer, by Computer Name. DevNet then generates a license file.
3. You *download* this file and use it to license Logi Studio and each Logi Info application you develop, or other Logi products.
4. Repeat Steps 2 and 3 for any additional license files required.

A typical license file name is `lgx120201.lic` and licenses are product-specific. License files are the same for both .NET and Java deployments.

When you download your license file, save it to:

Product/Platform	File Location	Name of License
Logi Info Server	In the application folder of any existing Logi application that you've created with a trial license, or any application that's being upgraded to v14.0 or greater.	`lgx140102.lic`
Logi Info Studio & Dev Server	`C:\Program Files\LogiXML IES DEV\LogiStudio` and in the application folder of any existing Logi application that you've created with a trial license, or any application that's being upgraded to v14.0 or v14.1.	`lgx140201.lic`

Moving a License to Another Machine

If you need to move an application to another machine, or to re-install your Logi product on a different machine, you can "un-assign" the license you've been using from one machine, assign it to another one, and generate a new license file. **OEM** licenses operate in a similar fashion, but can be assigned to multiple machines.

Logi Info Centralized License File

If you have a large number of Logi Info apps on the same development machine or web server, you may care to use a single, centralized license file rather than managing multiple license file copies.

In Logi Info v23.1, the **General** element in the _Settings definition includes a **License File Location** attribute where the location of a centralized license file can be specified. This value is a fully-qualified web server file system path to the folder where the file is stored. For example, `C:\myProjects\License`.


You must ensure that all web applications that will use this license run under an account that has **file access permissions** to access files in the folder you specify as the centralized license file location. With this centralized license file approach, if you use the exact same file path to your license file on both your development machine and your production server, you can deploy Logi applications by copying them, without having to replace the license file or adjust your _Settings attributes each time.

Logi Info OEM License File

Customers with Logi Info OEM licenses can embed the entire license key *inside* their Logi application. In the _Settings definition, the **General** element has an **OEM Distribution License** attribute which allows Logi applications to run on a server that does not otherwise have a license key installed. This is especially useful for XCOPY-type deployments: the Logi installation program does not have to be run on the web server. To use this attribute, double-click the attribute name to open its Zoom window and then

copy the entire XML contents of your OEM license file into it. This attribute *onlyworks* with OEM licenses; other types of licenses copied into here will be ignored.

Discovery 3.0+ and DataHub 3.0+ Licenses

The very first time one of these products runs, the contents of any license file found (usually the trial license) are automatically copied into the Platform Database (PDB) and used directly from there. After that, any license file in that folder will be *ignored*. When the trial license expires or if you need to use a different license, you'll need to perform the steps detailed in "Other Products: Missing or Expired License" on page 56 to "replace" the license in the PDB with another license.

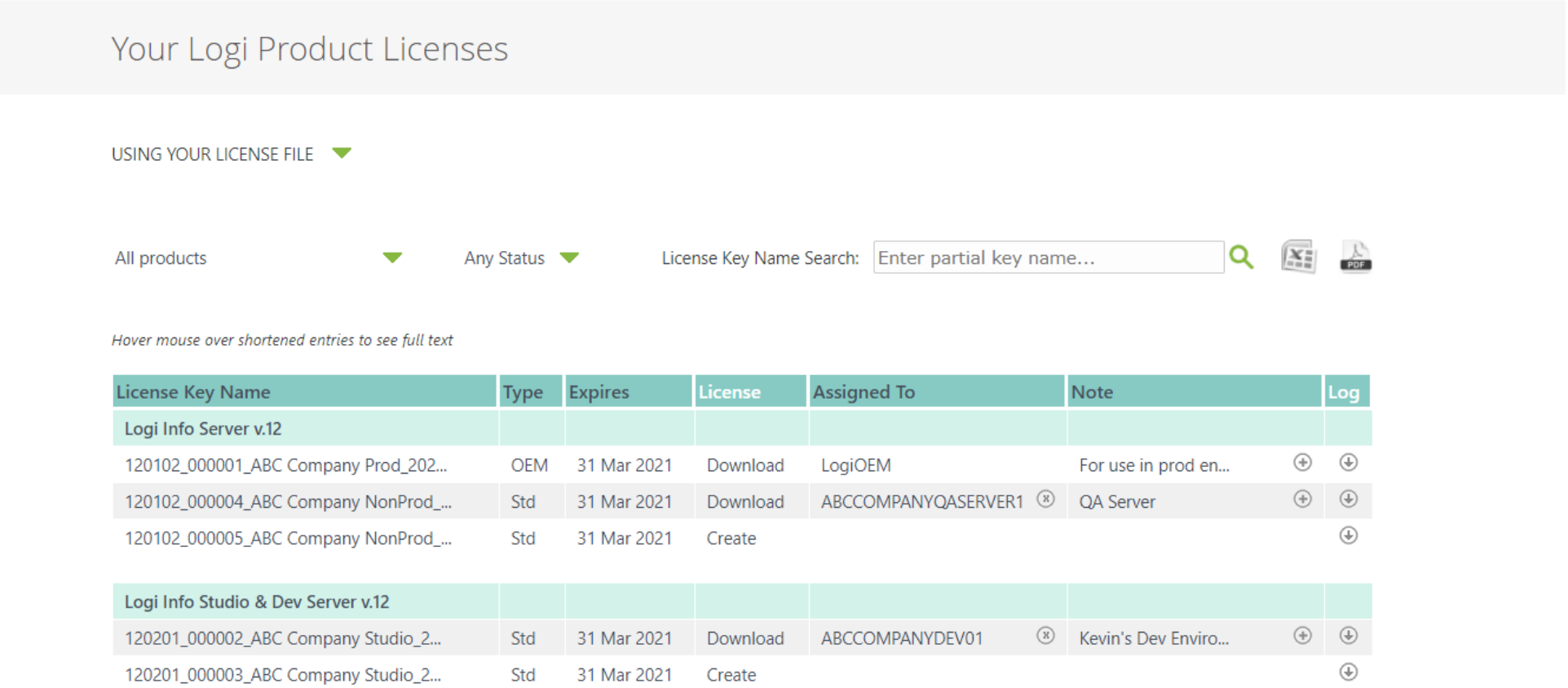
Product	License Requirement	License File Location
Discovery Module pre-v3.0	Requires Logi Info license file	Logi application folder
Discovery Module v3.0+	Requires Discovery v3.0 license file	Installation folder, default is `C:\LogiAnalytics\Discovery`
DataHub pre-v3.0	Requires Logi Info license file	Logi application folder
DataHub v3.0+	Requires DataHub 3.0 license file	Installation folder, default is `C:\LogiAnalytics\Discovery`
Logi Info, v12.5+ includes Discovery v3.x/Logi Services	Requires Logi Info v12 and Discovery v3.0 license files, or Logi Info v12 license file, or Logi Info v12 and DataHub v3.0 license	Logi application folder

Product	License Requirement	License File Location
	files	

 If you deploy your PDB to another computer, for example by copying it from a development system to a production system, you will need to replace the development system license in the PDB with the production system license using this technique.

License Management

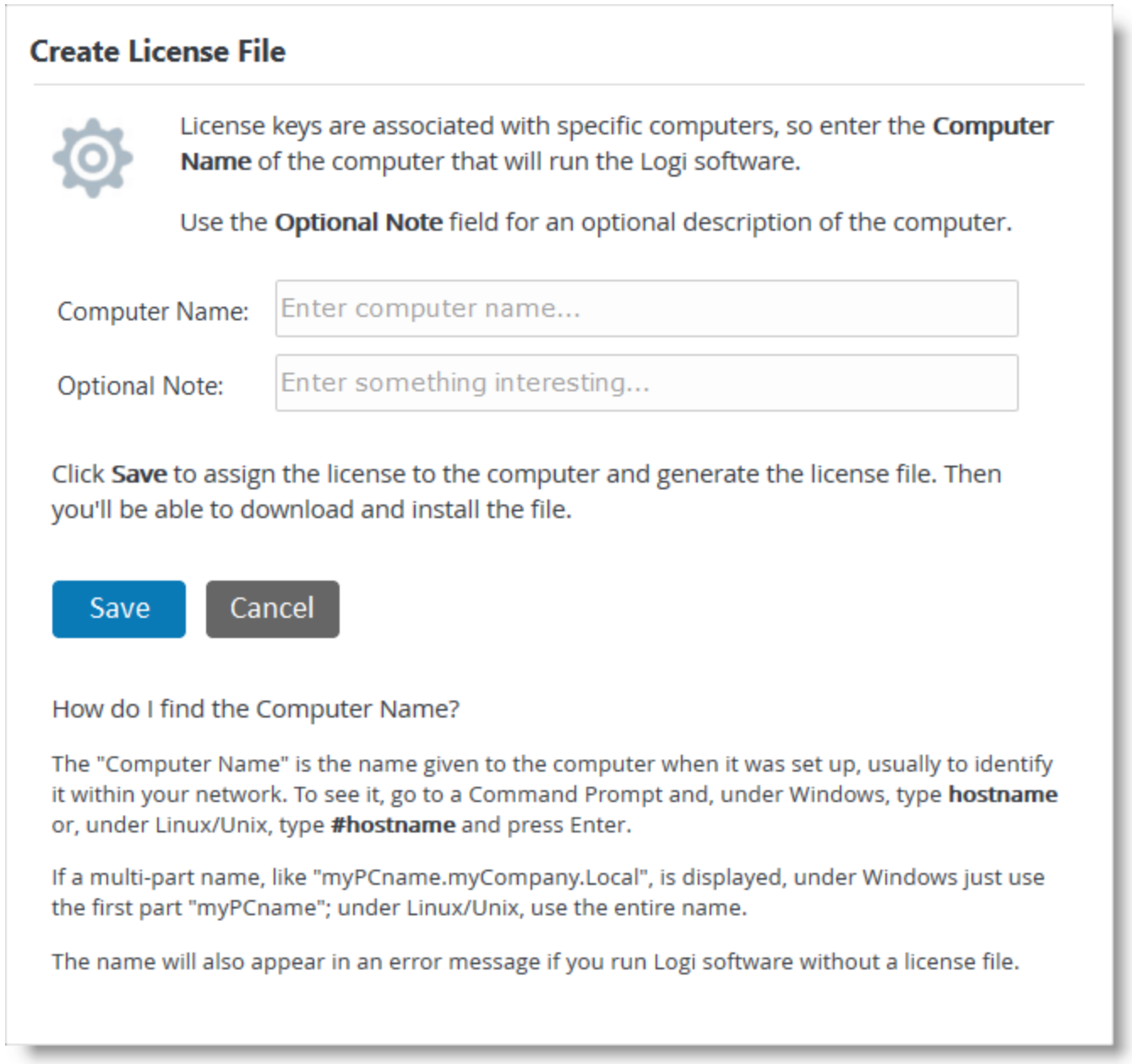
DevNet includes a **License Management** page where you can manage your product licenses and download license files. In order to manage your licenses, you must be a DevNet member and login to the DevNet site. Once you are logged in, go to **Support → License Manager**.



The details of the License Management page are shown above and explained below:

1. **Products, Status, Search** - These controls let you filter the table of license keys, by product type and status, and search based on License Key name.
2. **Export** - Use these icons to export your license key list to Excel or PDF.
3. **License Key Name** - The license key we created for you when you purchased the product appears here.
4. **Type & Expires** - The license type and expiration date appear here.
5. **License** - License actions: click the "Create" link in this column to assign your license to a specific machine. Once done, the machine name will appear in the next column and the link will change to "Download". Click the link again to download and save your license file. You may download the file as many times as necessary but it's only valid on the assigned machine.
6. **Assigned To** - The name of the machine a license has been assigned to. The icon allows you to "un-assign" the license - when this is done, the License link will revert back to "Create", and you may assign the license to another machine.
7. **Note** - This column displays optional text you enter to help you more easily identify the computer. Click the icon to add or edit this text.
8. **Log** - Click the icon to display the history of assignments for this license.

Here are the details of the process of creating and downloading a license file:


Create License File
License keys are associated with specific computers, so enter the Computer Name of the computer that will run the Logi software.
Use the Optional Note field for an optional description of the computer.
Computer Name:
Enter computer name...
Optional Note:
Enter something interesting...
Click Save to assign the license to the computer and generate the license file. Then you'll be able to download and install the file.
Save
Cancel
How do I find the Computer Name?
The "Computer Name" is the name given to the computer when it was set up, usually to identify it within your network. To see it, go to a Command Prompt and, under Windows, type hostname or, under Linux/Unix, type #hostname and press Enter.
If a multi-part name, like "myPCname.myCompany.Local", is displayed, under Windows just use the first part "myPCname"; under Linux/Unix, use the entire name.
The name will also appear in an error message if you run Logi software without a license file.

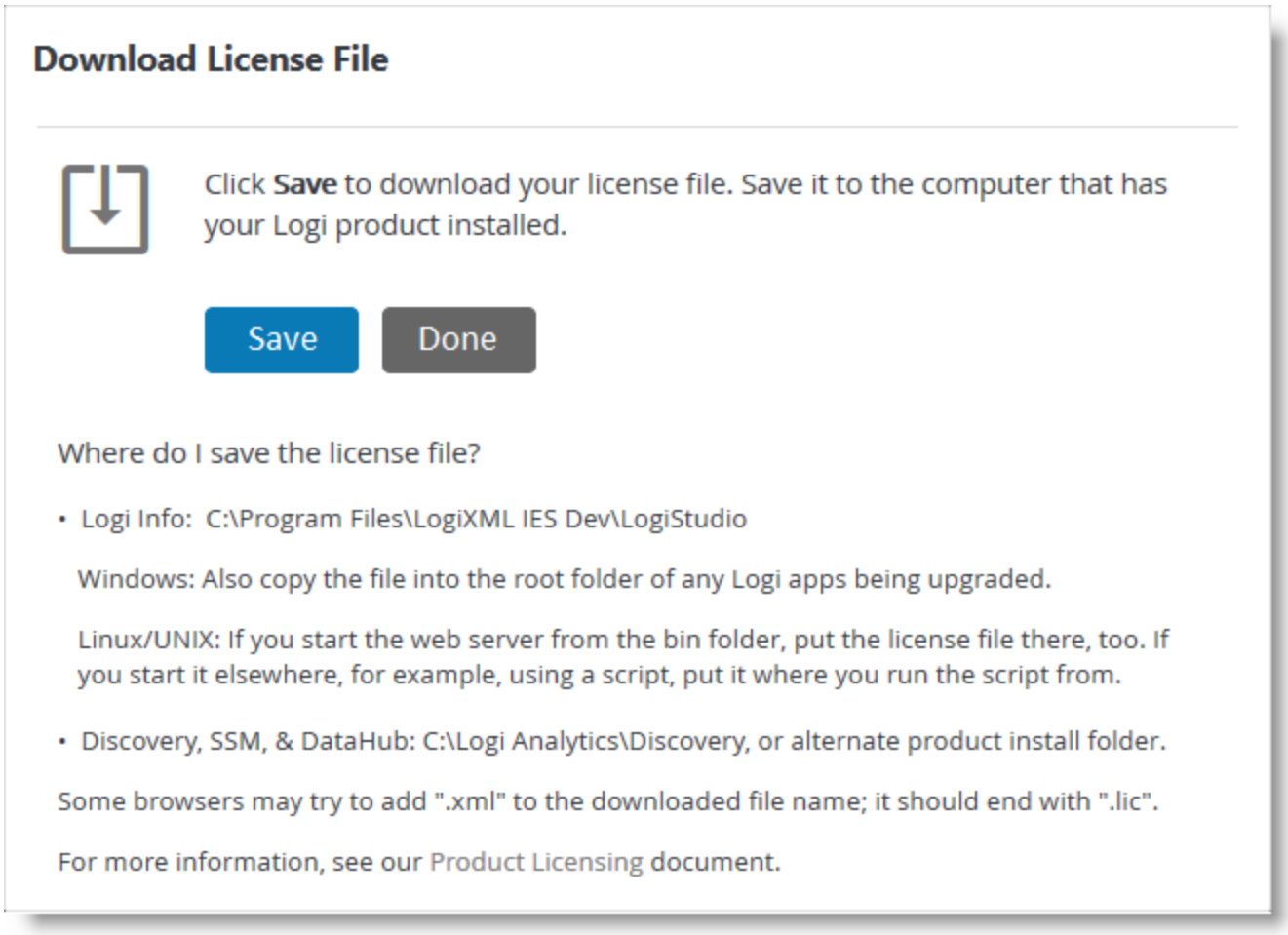
If the link in the Licenses column of the table shown earlier says "Create", clicking it will display the **Create License File** pop-up panel, shown above. You use it to assign the license to a specific machine, by entering the computer name. The Optional Note input is for your use if you want to further identify the machine and its text appears in the table of licenses shown earlier.

 The machine name "LogiOEM" (or any case variation thereof) is not a valid machine name.

If the computer on which this license is to be used runs a **Windows** OS, then the computer name value is what's known in Windows parlance as the "machine name". You can open a Command Prompt window on the machine and enter "hostname", then press Enter to have the name displayed. If a multi-part name is displayed, such as "myPC.myCompany.local", use just the first part, "myPC".

If the computer on which this license is to be used runs a **Linux or other UNIX-derivative** OS, then the "computer name" value is the computer's "hostname". You can go to a command line and enter "#hostname", then press Enter to have the name displayed. Use the *full* hostname value, exactly as it's displayed, for your computer name value entry.

Click **OK** to generate the license file for the designated computer.



Once the license file has been generated, the link in the Licenses column will change to "Download". Click this link to display the Download License File pop-up panel, shown above. Click **Download** to download and save the license file.

Some browsers may add an `.xml` file extension to the downloaded file, which is *wrong*. A typical license filename looks like `lgx120201.lic` so don't let your browser add anything to it. If the computer on which this license is to be used runs a **Windows** OS, then save the license file to

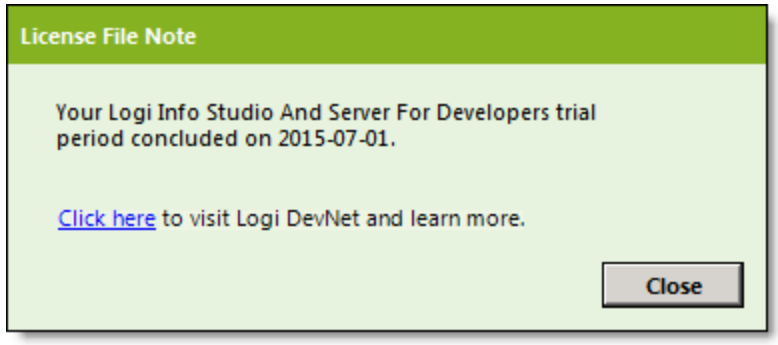
```
C:\Program Files\LogiXML IES Dev\LogiStudio
```

This will provide licensing for development work using Logi Studio, as well as for new Logi applications that are developed on the machine and run there using any web server. If you're upgrading any older Logi applications, you'll also need to copy the license file into their application folder.

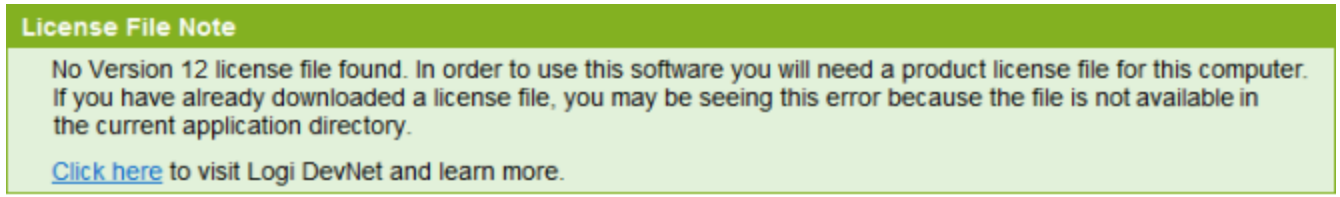
If the computer on which this license is to be used runs a **Linux or other UNIX-derivative** OS, then the license file location will depend on how you start the web server. If you start it from the `bin` folder, then save the license file there. If you start it elsewhere, for example, by running a script, then save it to the folder that you run the script from.

Logi Info: Missing or Expired License

As mentioned earlier, Logi Studio and your Logi Info applications require license files.



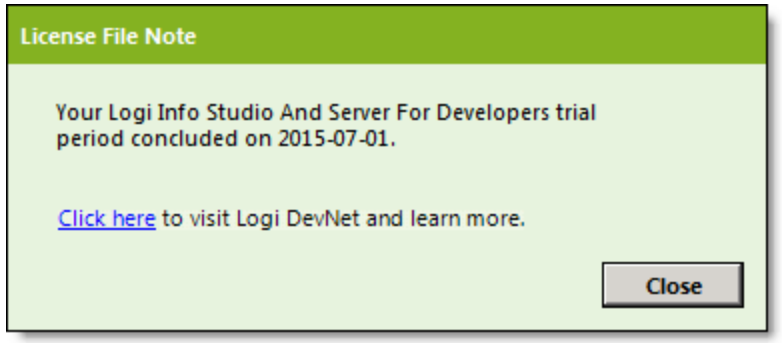
If no license file can be found when you start Logi Studio, a trial license will be automatically created but it may be expired (it's based on the date when the product was installed). If so, you'll see a message similar to the one above and be unable to use Studio.



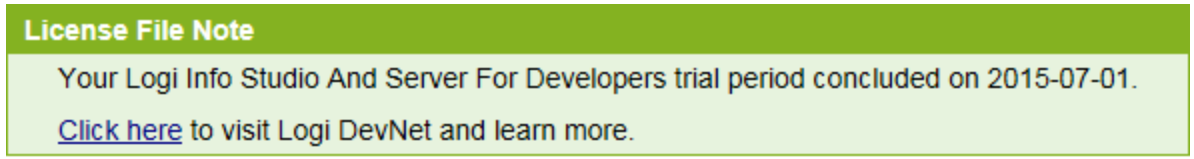
If no license file can be found when you try to browse a Logi application, you'll see a message similar to the one above in your browser, instead of the application page. Providing copies of a valid license file in the correct locations will allow Logi Studio and the application to run.

Logi Info Trial License Expires

When your trial license expires, and there is no regular license, you'll see the following messages:



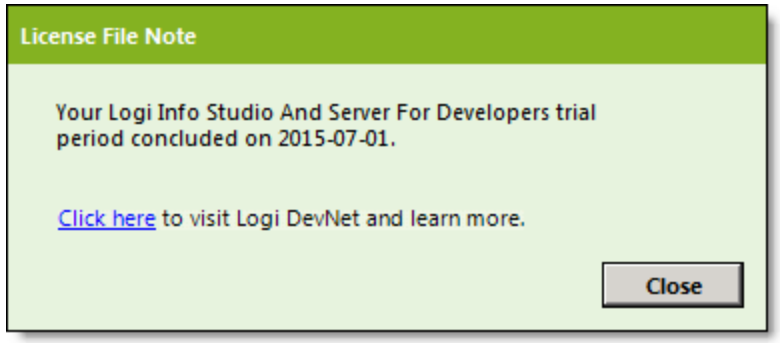
If the trial license has expired, when you start Logi Studio, you'll see a message similar to the one above and be unable to continue to use Studio.



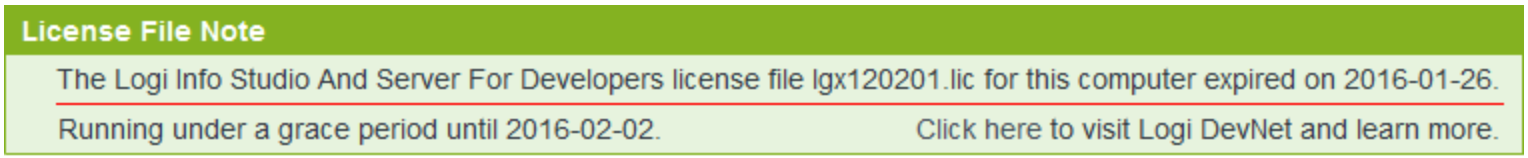
If the trial license has expired and you try to browse a Logi application, you'll see a message similar to the one above in your browser, instead of the application page. Providing copies of a valid license file in the correct locations will allow Logi Studio and the application to run.

Logi Info Regular License Expires

When your regular license expires, you'll see the following messages:



If the regular license has expired, when you start Logi Studio, you'll see a message similar to the one above and be unable to continue to use Studio.



Product Details

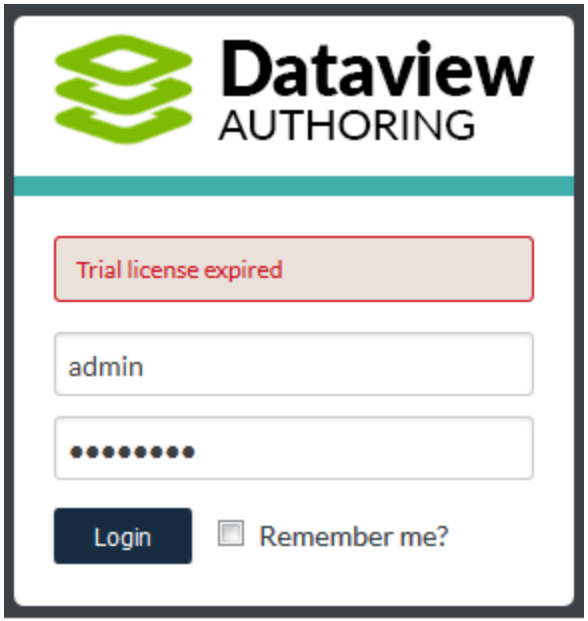
Product ID	Product Name	Quantity/Unit	Unit Price	Units In Stock
1	Chai	10 boxes x 20 bags	$18.00	39
2	Chang	24 - 12 oz bottles	$19.00	17
3	Aniseed Syrup	12 - 550 ml bottles	$10.00	13
4	Chef Antons Cajun Seasoning	48 - 6 oz jars	$22.00	53

If the regular license has expired and you try to browse a Logi application, your application page will be displayed with a special warning banner added at the top, as shown above. A 7-day grace period is provided from the date when the license expired.

Once the grace period ends, reports will no longer run and only the message shown above will be displayed in the browser. Providing copies of a valid license file in the correct locations will allow Logi Studio to be used and the application to run without the warning banner.

Other Products: Missing or Expired License

Other licensed Logi products, including the Discovery Module and DataHub, react differently when their licenses expire. A product or feature that requires a login, such as Dataview Authoring, will display an error message when a login is attempted:



Some products will allow you to view data but you cannot save visualizations or changes if the license has expired. Other products will cause the Logi Info applications they work with to throw an error, with a descriptive error message in the usual format.

Replacing Expired License Files

If your license for Discovery prior to Version 3.2 or DataHub 3.0+ expires, you'll need to replace it. The very first time one of these products runs, the trial license file contents are automatically copied into the Platform Database (PDB) and used directly from there. Afterwards, the actual trial license file and any other license files subsequently placed in the folder, will be *ignored*. You'll need to perform the following to "replace" your trial license with an extended-trial or regular license.

WINDOWS: The license file record can be deleted from the PDB, and another one inserted to replace it, using DeleteLicenseDB.txt and InsertLicenseDB.txt, as follows:

1. Place the new license file in the same folder as the old license file, typically `C:\LogiAnalytics\Discovery`.
2. Download and save the two files above to your desktop.
3. Edit each file to include the correct Discovery Admin password, license file name, and license file path (if not the default).

   When editing DeleteLicenseDB.txt, you specify only the license file name, without its extension. For example, to delete the standard trial license, you specify "license-id=logiTrial03" (no quotes).
4. Rename the downloaded file extensions to ".bat".
5. Using *Run As Administrator*, execute the DeleteLicenseDB.bat file and allow it to fully complete.
6. Using *Run As Administrator*, execute the InsertLicenseDB.bat file and allow it to fully complete.

Information about this process is logged to: `C:\LogiAnalytics\Discovery\platform\logs\licenseImport.log`

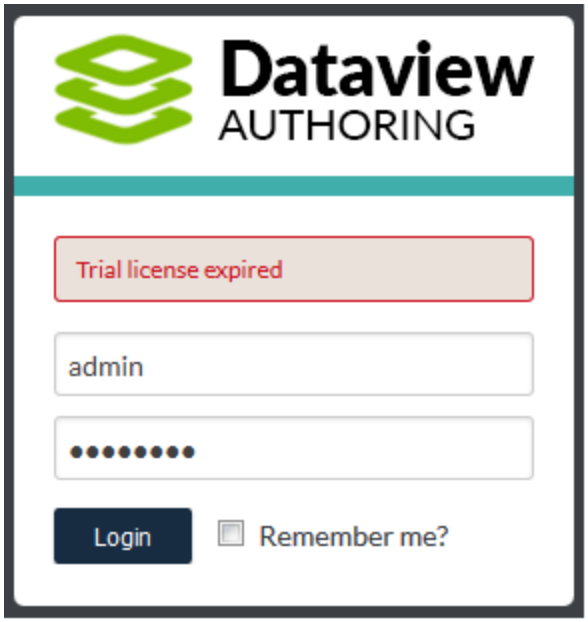
LINUX/UNIX:

These steps assume Discovery was installed in `/opt/Discovery` - adjust as necessary for your installation location.

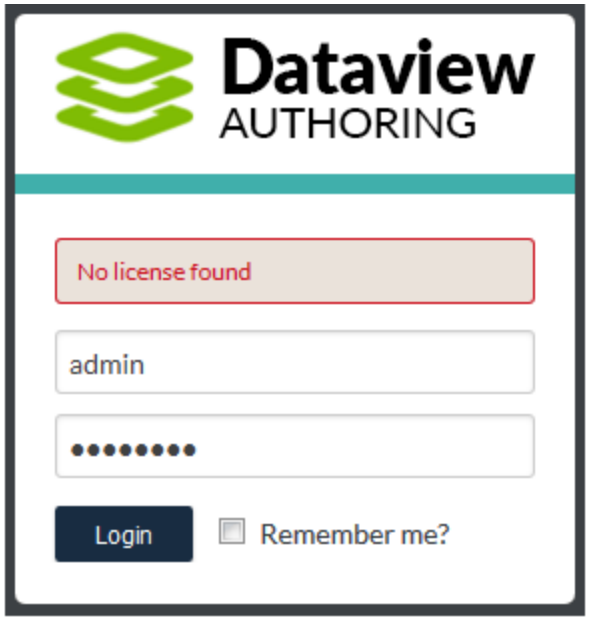
1. Place the new license file in the same folder as the old license file, typically `/opt/Discovery`.
2. Download and save the two files above to your desktop.

3. Manually stop the Logi Application Server and Logi Data Service daemons.
4. Manually run `/opt/Discovery/platform/bin/licenseTool.sh`, editing and using the arguments found in the DeleteLicenseDB file.
5. Manually run `licenseTool.sh` again, editing and using the arguments found in the InsertLicenseDB file.
6. Manually start the Logi Data Service daemon.
7. Wait 20 seconds, manually start the Logi Application Server daemon

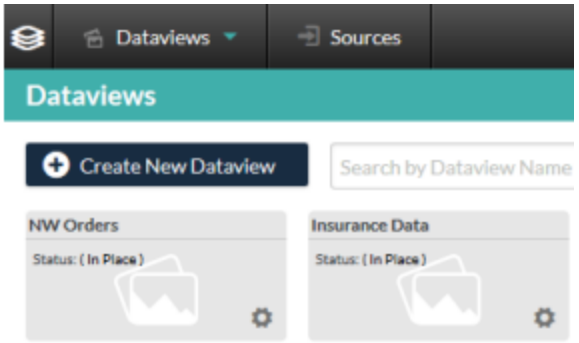
Information about this process is logged to: `/opt/Discovery/platform/logs/licenseImport.log`



Login attempt: License expired...



...after License deleted...



... after License inserted

Server Virtualization

Many organizations are using server virtualization to maximize hardware usage and reduce costs.

Server virtualization products allow the assignment of CPU resources to processes. This may take the form of a maximum percentage of combined CPU utilization, or as specific allocation of logical CPUs, to a virtual machine (VM). The server administrator is responsible for making these configuration decisions.

Logi v23.1 product licenses treat a VM just like a regular server and run just fine in this environment.

In order to ensure good performance in any virtualized server environment, administrators must be careful to allocate appropriate resources to VMs.

It's not uncommon to relocate a VM from one hardware platform to another, for example, for hardware maintenance. The Logi license will "move" with the VM, as long as its machine name doesn't change.

Install Logi Info on Windows Server 2019

Welcome to Logi Analytics reporting products. The following topics guide you through installing Logi Info on a single computer under the Windows Server 2019 operating system, for use with the IIS web server:

Install Logi Info on Windows Server 2019 - Installation Scenarios

There are two major parts to the Logi product in the installation file you downloaded:

- Logi Studio, our development environment, is a Windows application that's typically installed on a Windows development machine and interacts with the local IIS web server for Logi application development and testing.
- Logi Server Engine is a set of files that's part of each Logi application and which provides an extension to the IIS web server at runtime. When you build a Logi application, Studio adds the Engine files to the application. The engine also includes a Windows utility application, Logi Server Manager, that allows you to perform basic configuration of Logi applications without using Studio.

A very typical installation scenario is to install Studio and the Server Engine on a development machine, and then install only the Server Engine (including Server Manager) on the production web server. This "development and production" dual installation is allowed by our licensing scheme.

Additional copies of Studio may be licensed and installed on additional development machines.

There are a number of ways to *deploy* your Logi applications to production, including using Studio's built-in Application Deployment tool.

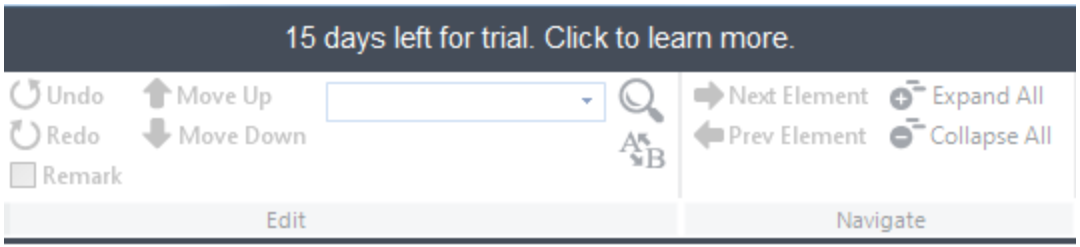
Other installation scenarios, involving shared network drives and team development, can be used. This topic covers installation of Studio and the Server Engine on a development machine, and also explains what's needed to install the Server Engine on a single production server.

64-bit Only

Windows Server 2019 is a **64-bit** operating system. There are different Logi product distribution files for 32-bit and 64-bit versions, so be sure you've downloaded the 64-bit version.

Install Logi Info on Windows Server 2019 - Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You need do nothing but install the product and you can begin using it. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about purchasing a Logi Info license.

After the trial period expires, Studio and any Logi reports you may have developed will *no longer run* without a real license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server.

Our licensing scheme allows you to deploy our product on one development machine and on one production server. Additional separate licenses for Studio, for additional developers, are also available.

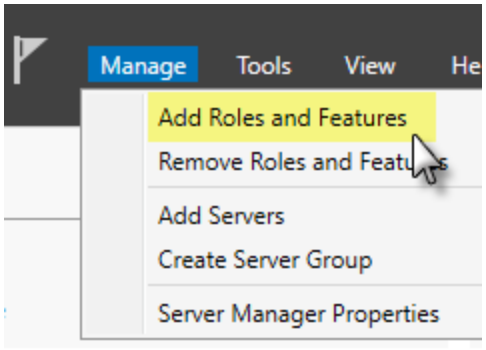
Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Install Logi Info on Windows Server 2019 - Preparing to Install

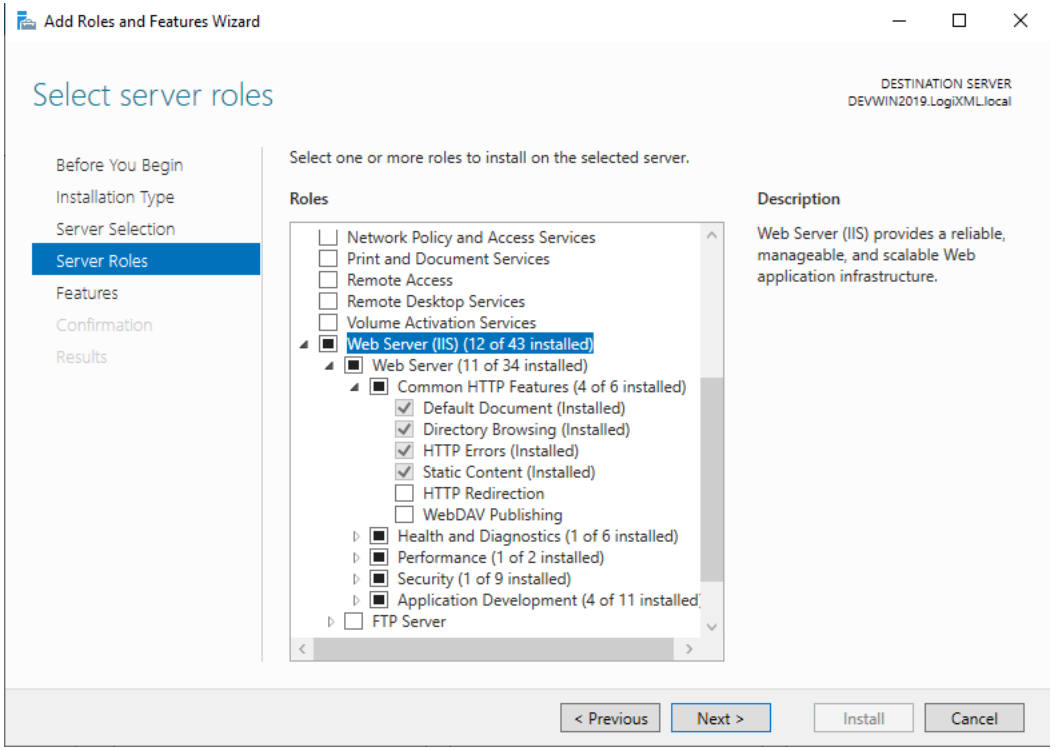
Administrator privileges on the target computer are *required* to complete the installation of Logi products. Logi Info for the Windows environment requires the .NET Framework 4.x. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Roles and Features

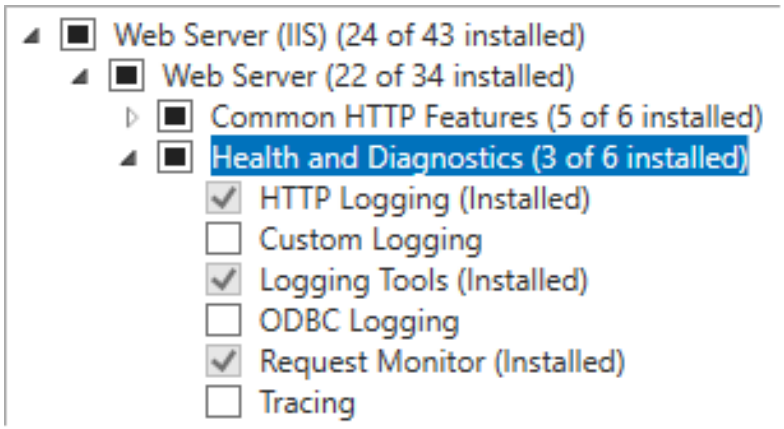
Before Logi product installation, you must ensure that the appropriate IIS roles and features have been enabled.

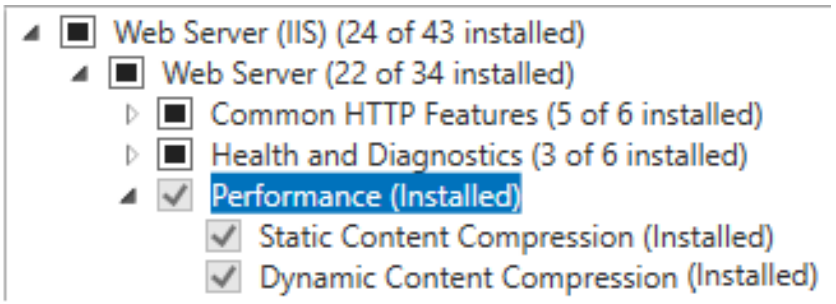


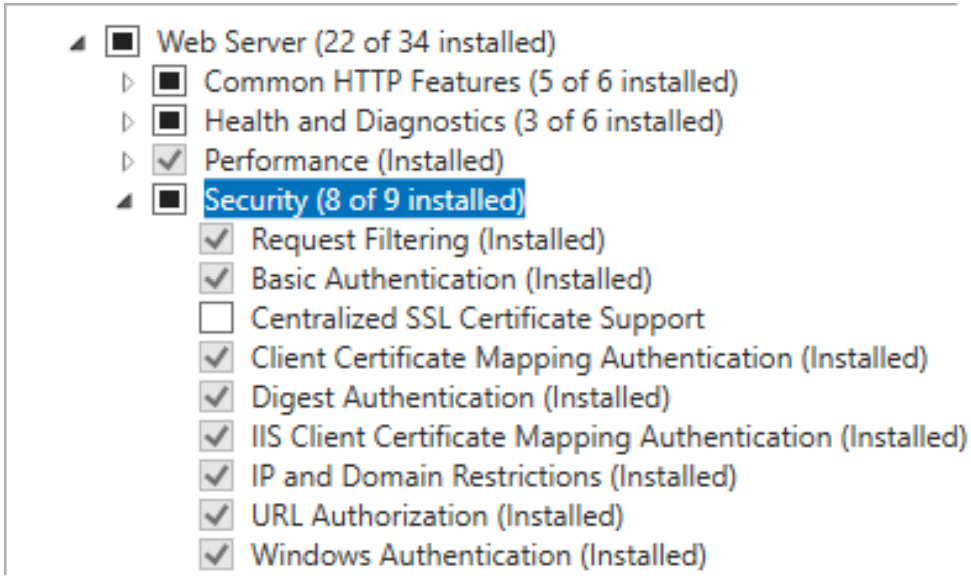
1. Use the Server Management tool to examine Web Server (IIS) role:

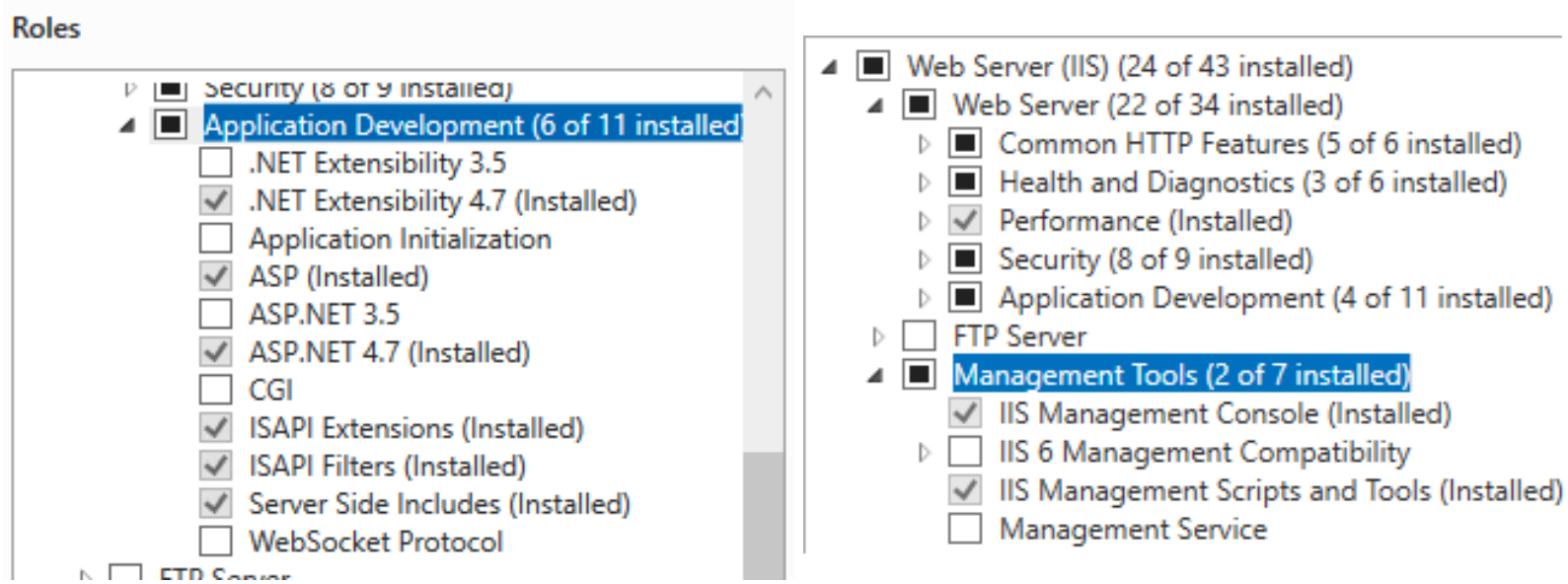


2. Check each sub-category to ensure correct configuration. Ensure that the features shown below are installed and enabled correctly:

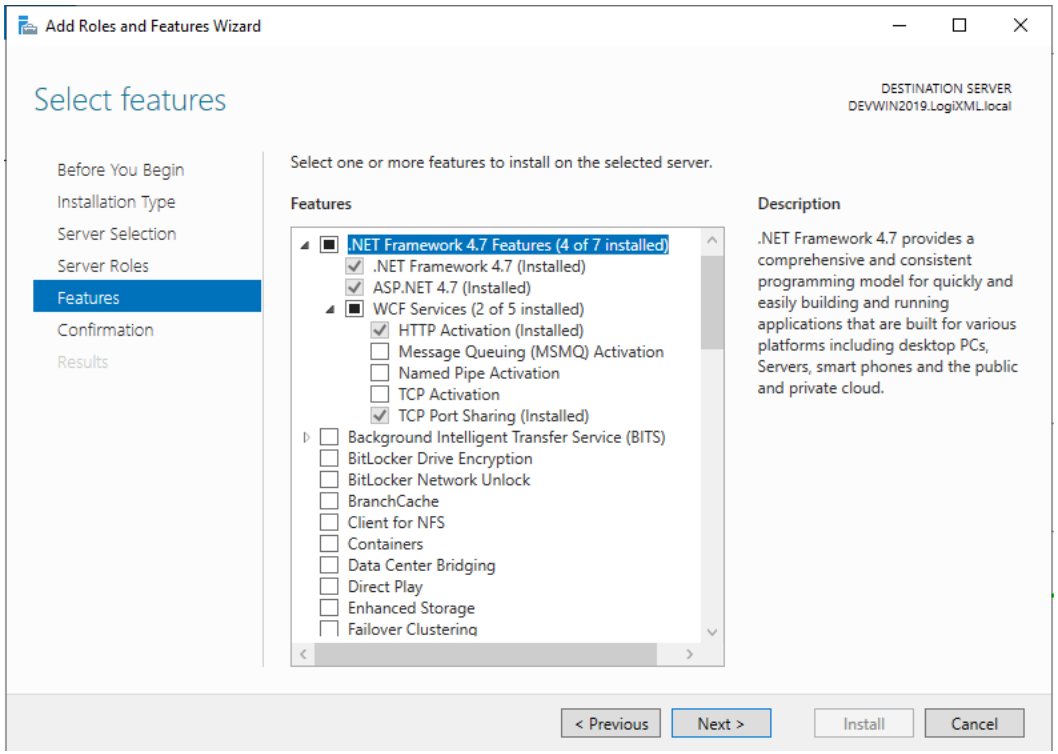

Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
HTTP Logging (Installed)
Custom Logging
Logging Tools (Installed)
ODBC Logging
Request Monitor (Installed)
Tracing


Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
Performance (Installed)
Static Content Compression (Installed)
Dynamic Content Compression (Installed)


Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
Performance (Installed)
Security (8 of 9 installed)
Request Filtering (Installed)
Basic Authentication (Installed)
Centralized SSL Certificate Support
Client Certificate Mapping Authentication (Installed)
Digest Authentication (Installed)
IIS Client Certificate Mapping Authentication (Installed)
IP and Domain Restrictions (Installed)
URL Authorization (Installed)
Windows Authentication (Installed)



3. Select **Next**.
4. Then examine the .NET Framework Features and ensure that the features shown below are installed and enabled:



5. Select **Next**.
6. Then, select **Install**.

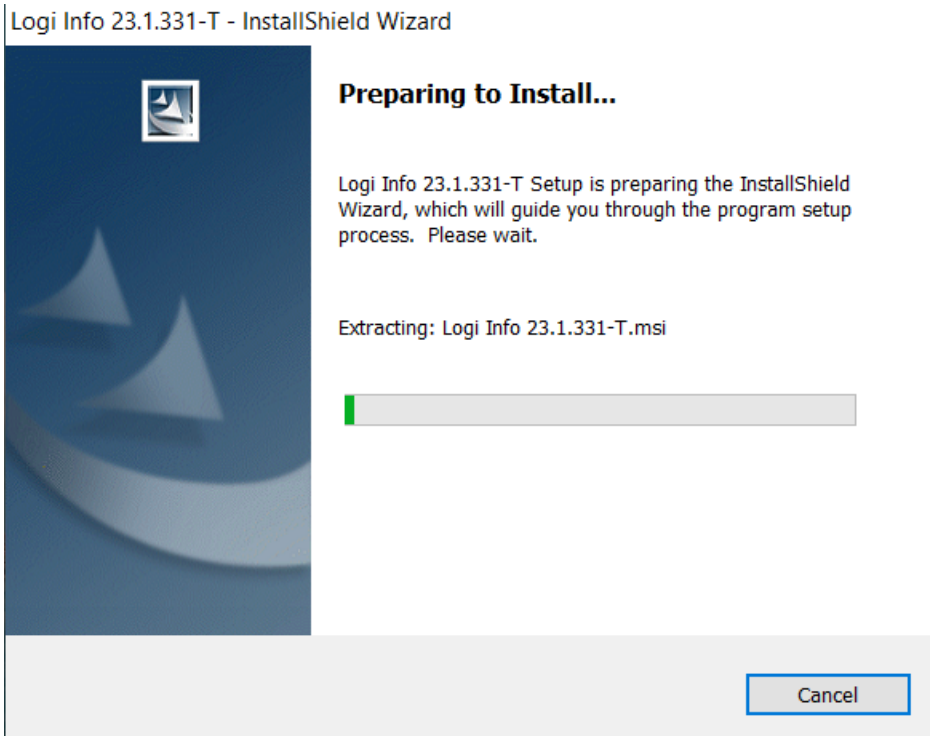
The IIS "Default Web Site"

When it's installed, IIS creates a "Default Web Site" and the New Application wizard in Logi Studio expects to create all new Logi applications as virtual directories of that web site. If you have renamed, replaced, or disabled the "Default Web Site", or have installed another web server that handles HTTP requests on Port 80, the wizard will fail during its application registration phase. Under these circumstances, you can continue to use Logi Studio to develop Logi applications but you will need to manually register them. This process is described in Windows IIS Configuration.

Install Logi Info on Windows Server 2019 - Installing the Software

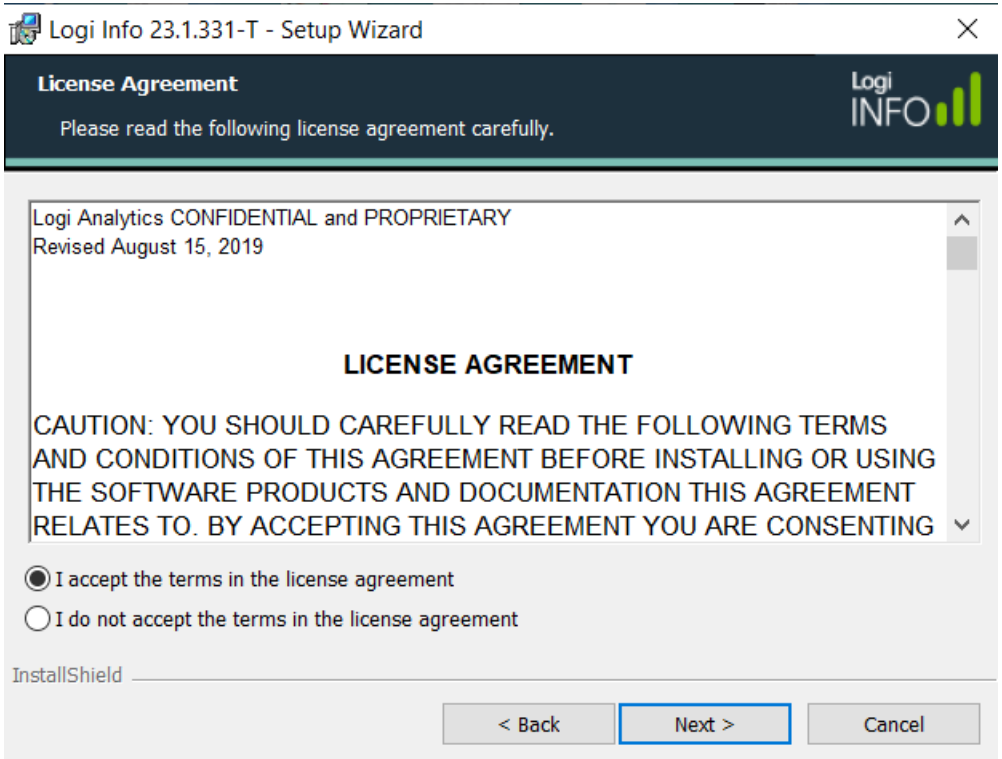
This topic describes the installation steps and the dialog boxes you'll see when using the interactive tool built into your Logi product installation file. If you need a "silent" (non-interactive) installation, see "Command Line Install" on page 327.

1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.



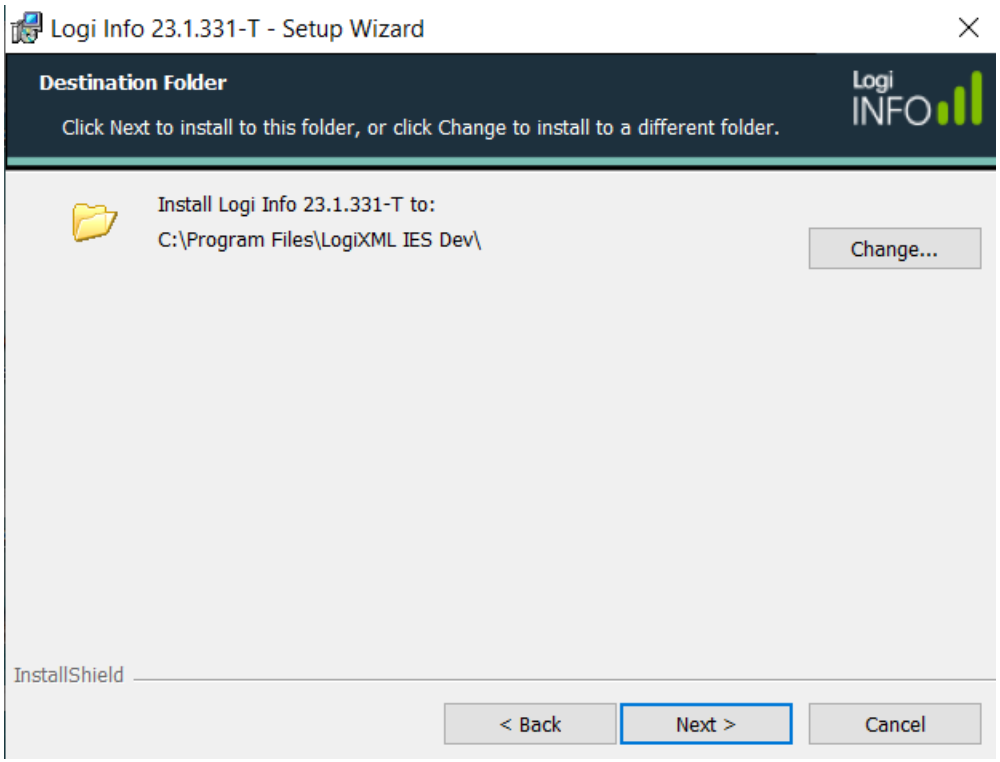
2. When the **Welcome Screen** appears, select **Next:**

3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and select **Next** to continue:

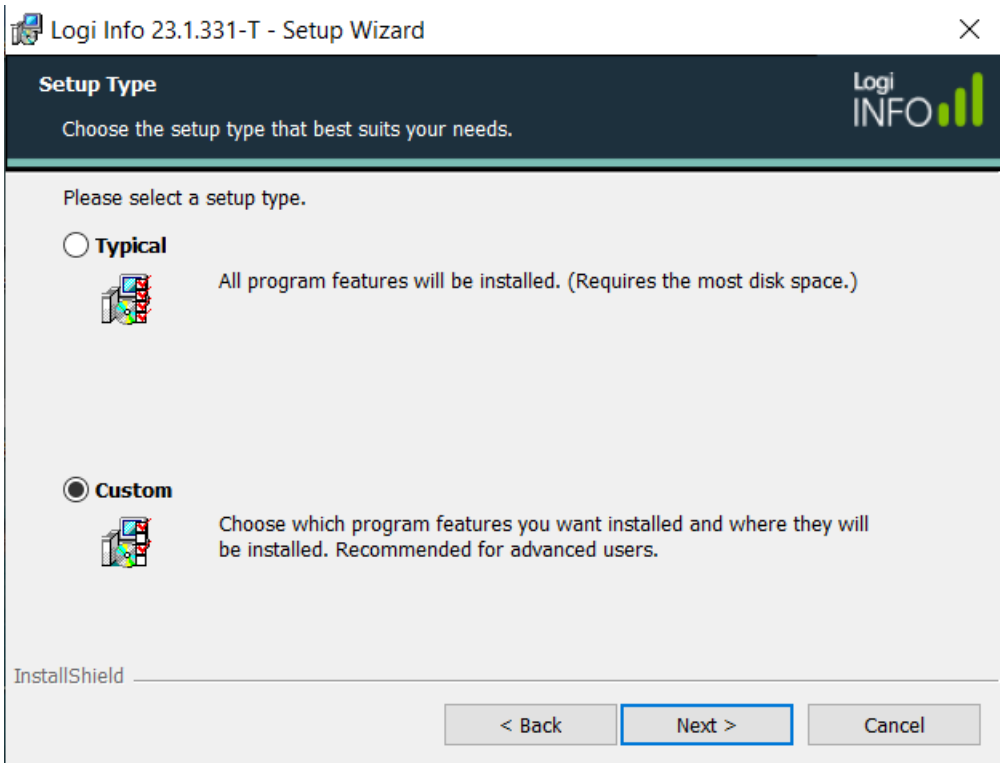


As usual, you can select **Back** at any time before the physical installation begins to go back to the previous screen.

4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



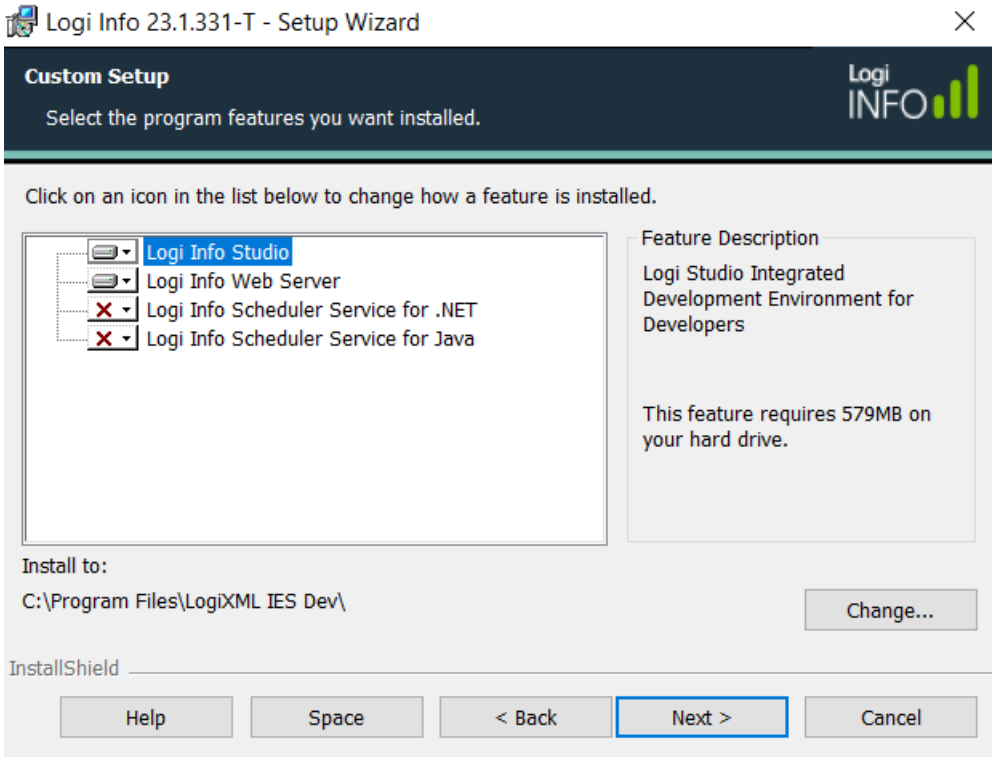
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

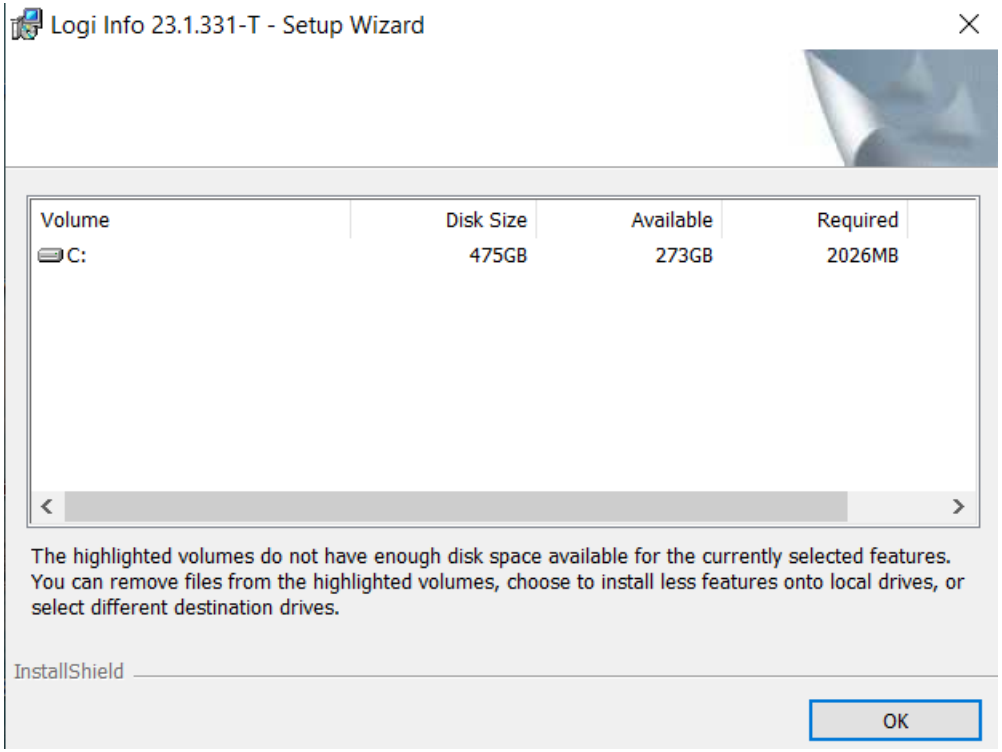
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

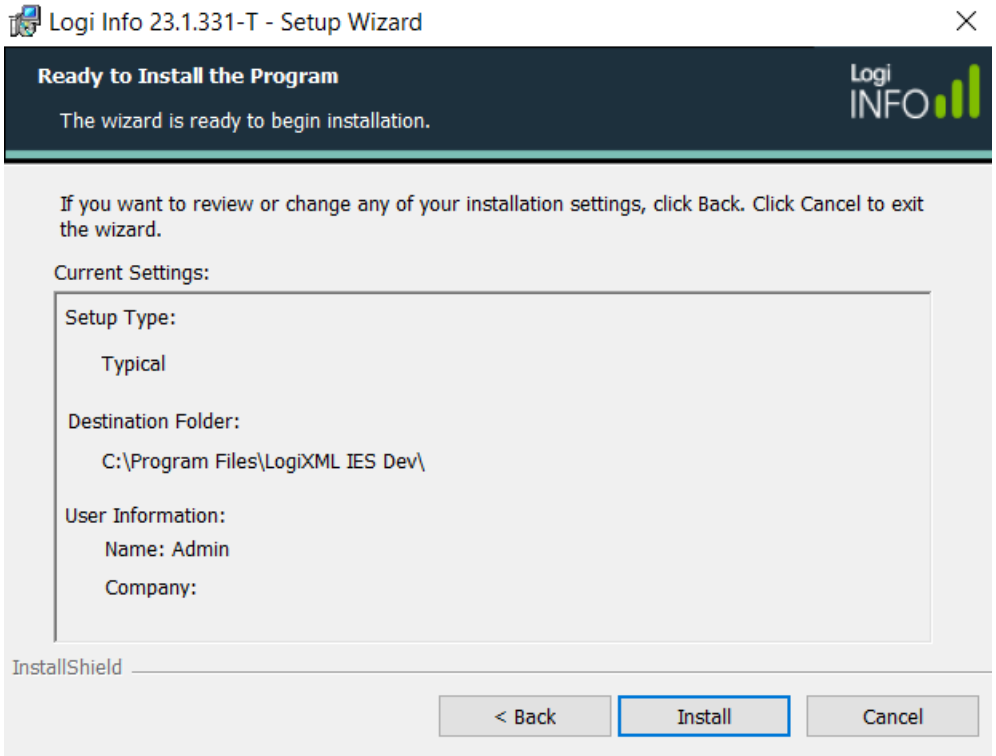
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



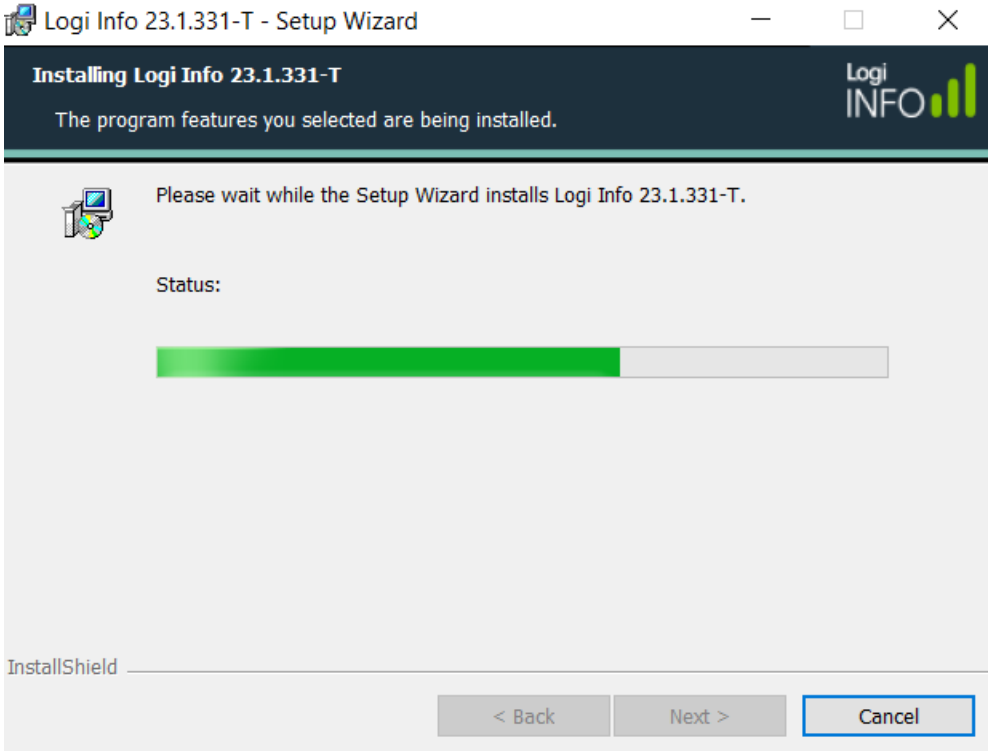
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



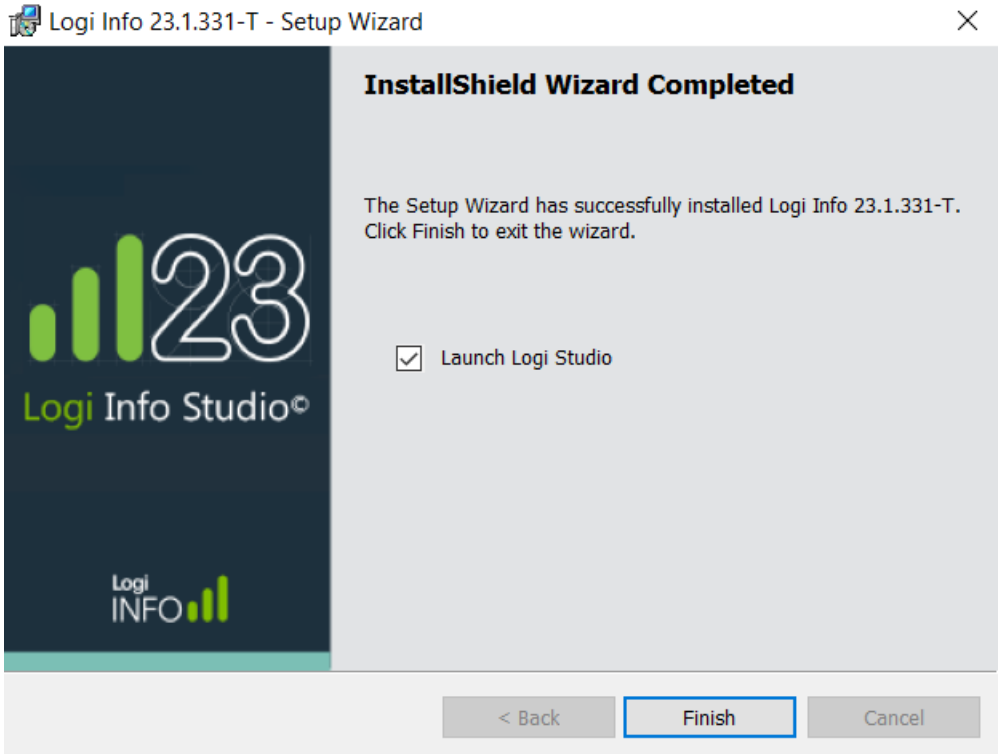
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).



9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

You can also launch Studio using the Start menu.

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in Using Logi Studio.

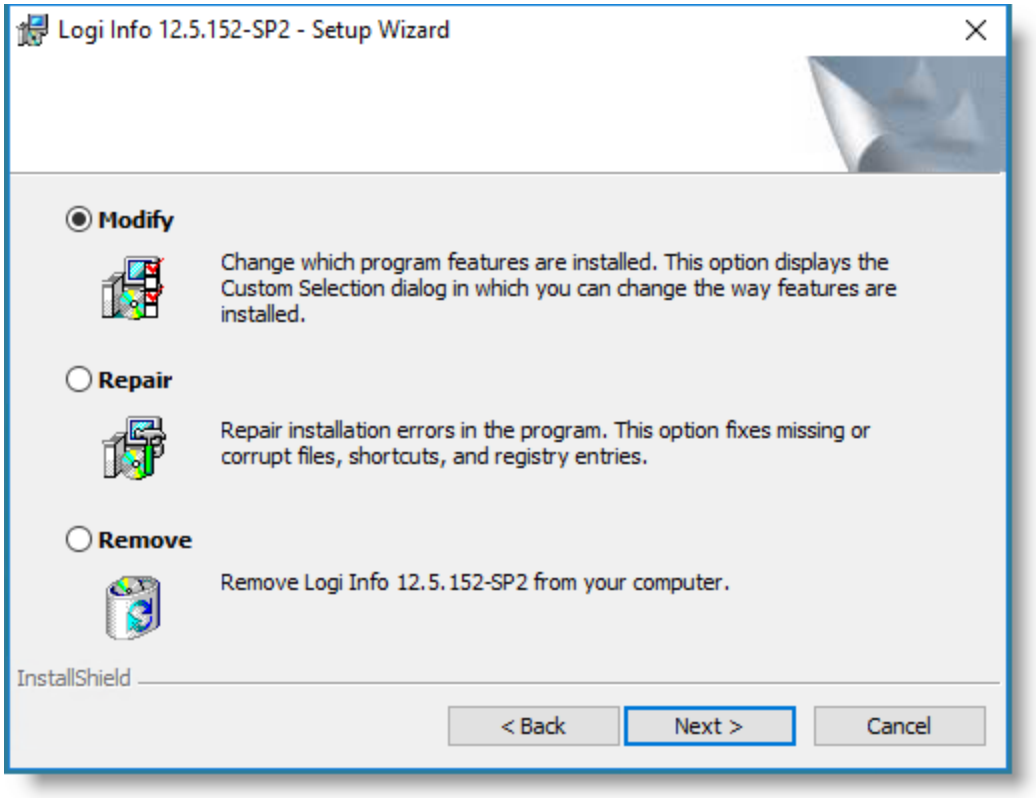
Installation is complete and you may begin to use Studio and/or the Server Engine immediately.

Install Logi Info on Windows Server 2019 - Installing Add-On Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. describes them and includes links to their installation topics.

Install Logi Info on Windows Server 2019 - Modifying or Repairing an Installation

Suppose you installed Logi Info but didn't initially install the Scheduler, and now you find you want to schedule reports. Or you suspect a .DLL file is missing or is corrupted and you want to fix it. These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation .exe file, using "Run As Administrator".



Do not use Control Panel → Add-Remove Programs to do this; it will request an .msi file, which is not retained after the original installation.

Install Logi Info on Windows Server 2016

Welcome to Logi Analytics reporting products.

The following topics guide you through installing Logi Info on a single computer under the Windows Server 2016 operating system, for use with the IIS web server:

Install Logi Info on Windows Server 2016 - Installation Scenarios

There are two major parts to the Logi product in the installation file you downloaded:

Logi Studio, our development environment, is a Windows application that's typically installed on a Windows development machine and interacts with the local IIS web server for Logi application development and testing.

The **LogiServer Engine** is a set of files that's part of each Logi application and which provides an extension to the IIS web server at runtime. When you build a Logi application, Studio adds the Engine files to the application. The engine also includes a Windows utility application, **Logi Server Manager**, that allows you to perform basic configuration of Logi applications without using Studio.

A very typical installation scenario is to install Studio and the Server Engine on a development machine, and then install only the Server Engine (including Server Manager) on the production web server. This "development and production" dual installation is allowed by our licensing scheme.

Additional copies of Studio may be licensed and installed on additional development machines.

There are a number of ways to *deploy* your Logi applications to production, including using Studio's built-in **Application Deployment** tool.

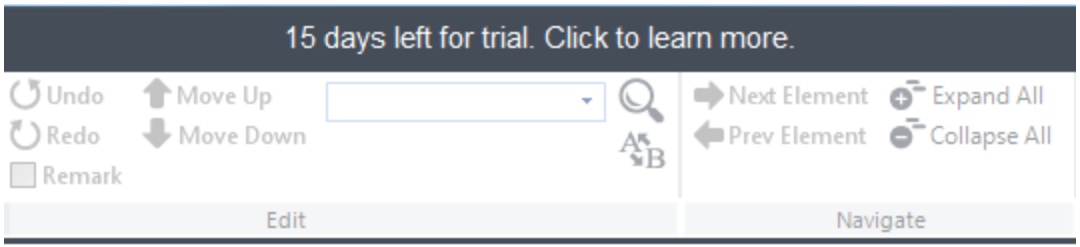
Other installation scenarios, involving shared network drives and team development, can be used. This topic covers installation of Studio and the Server Engine on a development machine, and also explains what's needed to install the Server Engine on a single production server.

64-bit Only

Windows Server 2016 is a **64-bit** operating system. There are different Logi product distribution files for 32-bit and 64-bit versions, so be sure you've downloaded the 64-bit version.

Install Logi Info on Windows Server 2016 - Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You need do nothing but install the product and you can begin using it. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about **purchasing** a Logi Info license.

After the trial period expires, Studio and any Logi reports you may have developed will *no longer run* without a real license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server.

Our licensing scheme allows you to deploy our product on one development machine and on one production server. Additional separate licenses for Studio, for additional developers, are also available.

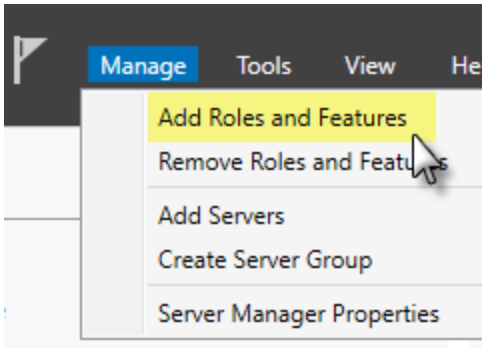
Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Install Logi Info on Windows Server 2016 - Preparing to Install

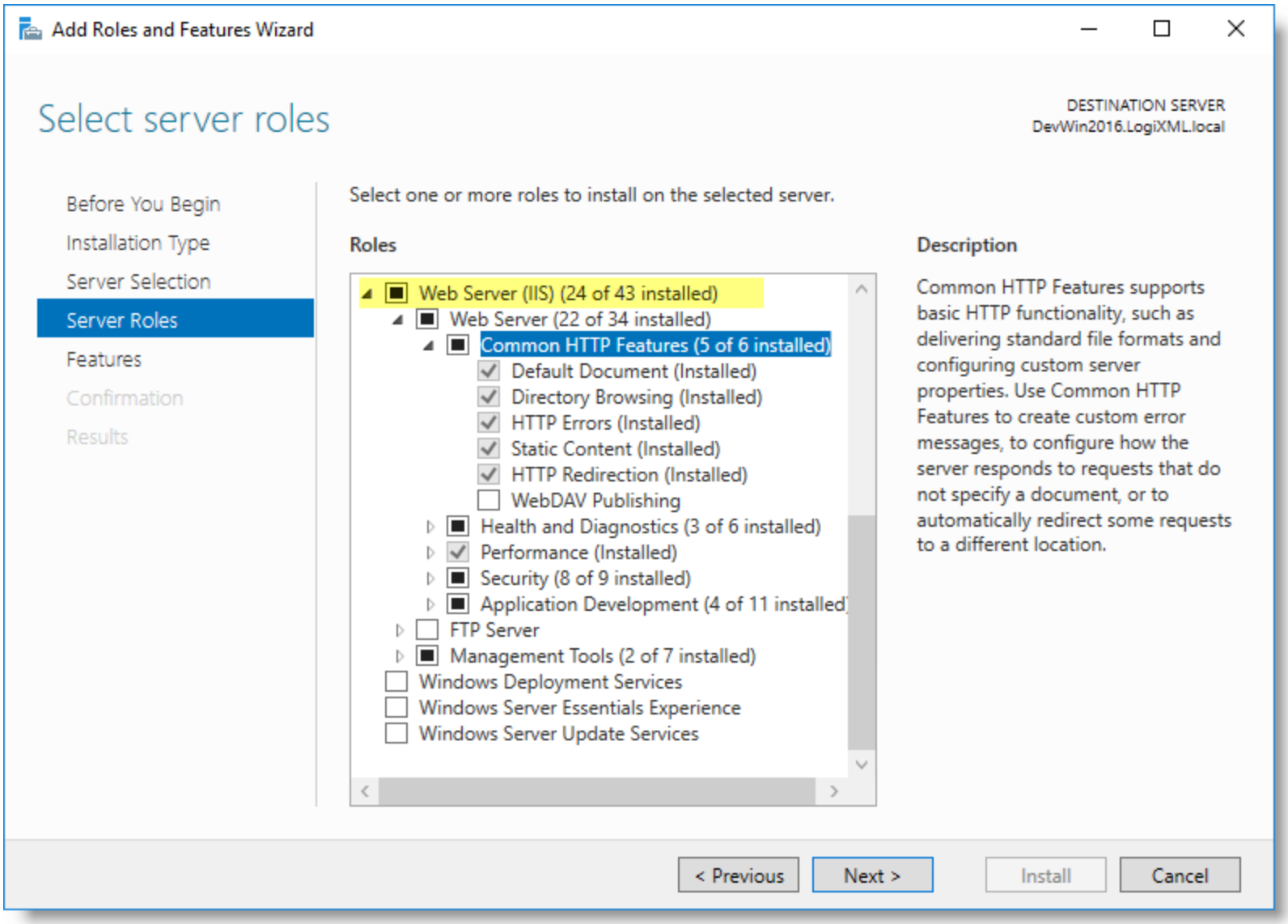
Administrator privileges on the target computer are *required* to complete the installation of Logi products. Logi Info for the Windows environment requires the .NET Framework 4.x. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Roles and Features

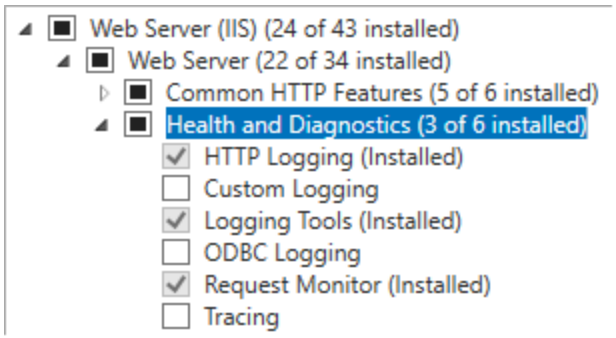
Before Logi product installation, you must ensure that the appropriate IIS roles and features have been enabled.

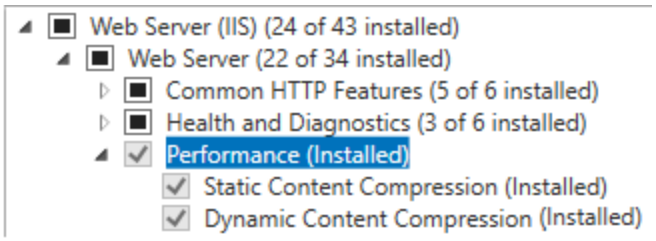


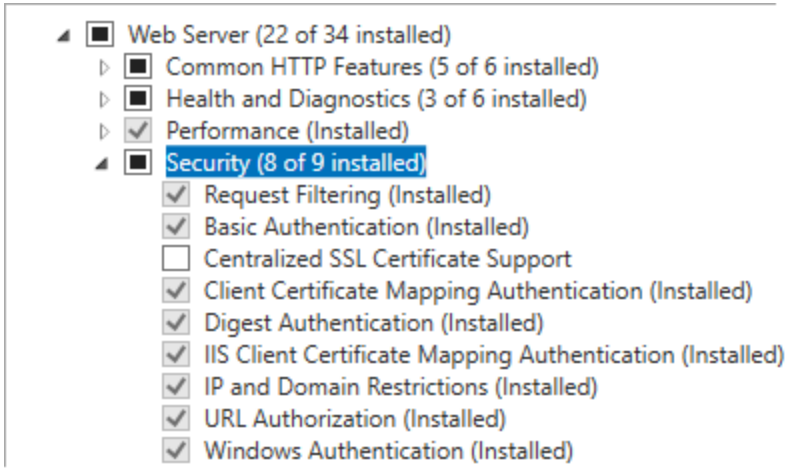
Use the Server Management tool to examine Web Server (IIS) role:

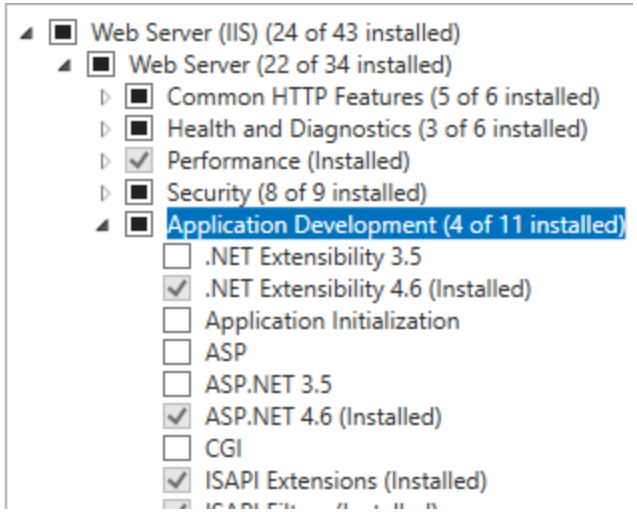


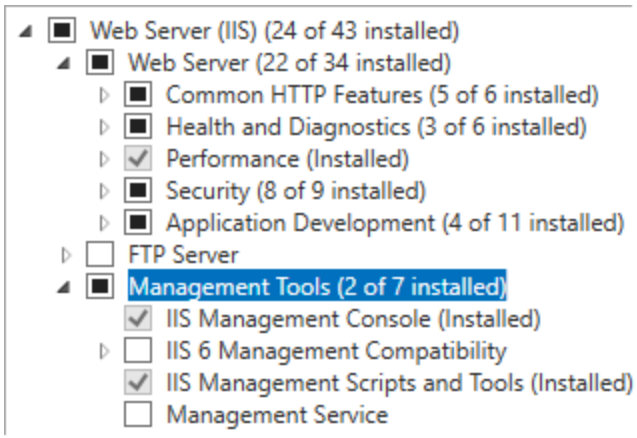
Check each sub-category to ensure correct configuration.


Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
HTTP Logging (Installed)
Custom Logging
Logging Tools (Installed)
ODBC Logging
Request Monitor (Installed)
Tracing

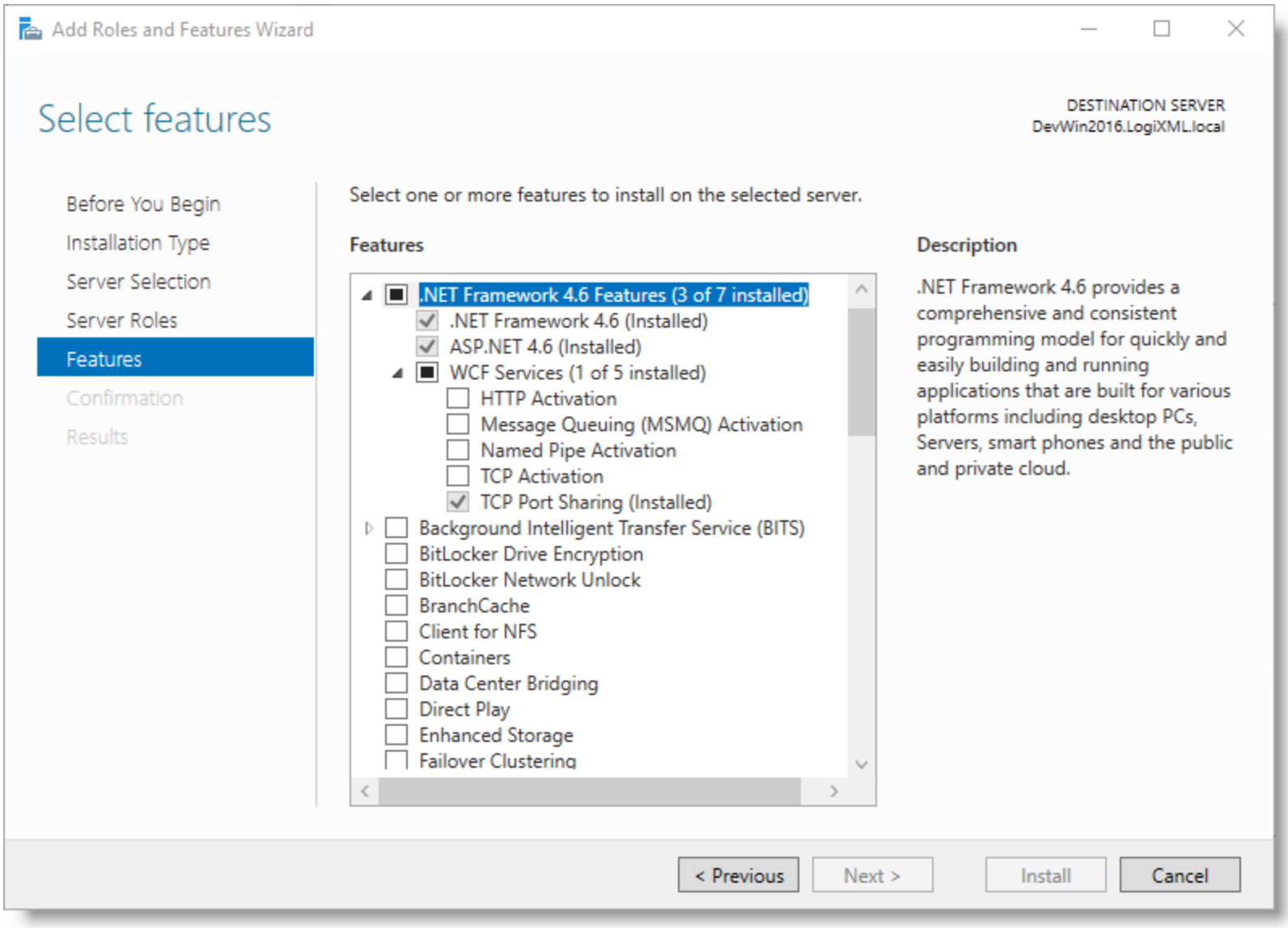

Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
Performance (Installed)
Static Content Compression (Installed)
Dynamic Content Compression (Installed)


Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
Performance (Installed)
Security (8 of 9 installed)
Request Filtering (Installed)
Basic Authentication (Installed)
Centralized SSL Certificate Support
Client Certificate Mapping Authentication (Installed)
Digest Authentication (Installed)
IIS Client Certificate Mapping Authentication (Installed)
IP and Domain Restrictions (Installed)
URL Authorization (Installed)
Windows Authentication (Installed)


Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (3 of 6 installed)
Performance (Installed)
Security (8 of 9 installed)
Application Development (4 of 11 installed)
.NET Extensibility 3.5
.NET Extensibility 4.6 (Installed)
Application Initialization
ASP
ASP.NET 3.5
ASP.NET 4.6 (Installed)
CGI
ISAPI Extensions (Installed)



Ensure that the features shown in the images above are installed and enabled correctly.



Then examine the .NET Framework Features and ensure that the features shown above are installed and enabled.

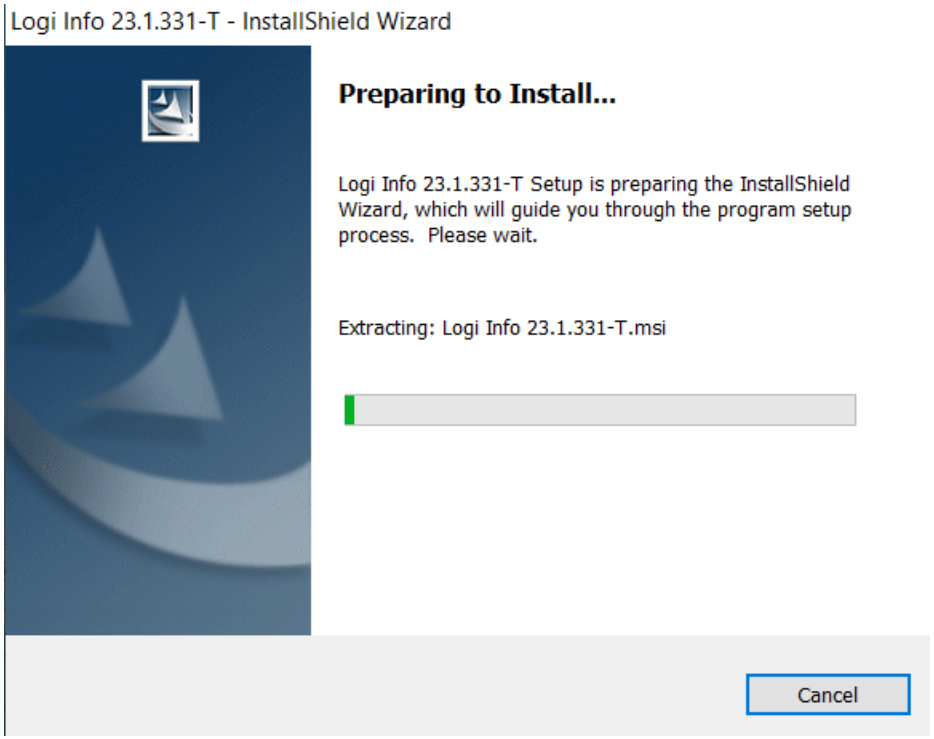
The IIS "Default Web Site"

When it's installed, IIS creates a "Default Web Site" and the New Application wizard in Logi Studio expects to create all new Logi applications as virtual directories of that web site. If you have renamed, replaced, or disabled the "Default Web Site", or have installed another web server that handles HTTP requests on Port 80, the wizard will fail during its application registration phase. Under these circumstances, you can continue to use Logi Studio to develop Logi applications but you will need to manually register them. This process is described in *Windows IISConfiguration*.

Install Logi Info on Windows Server 2016 - Installing the Software

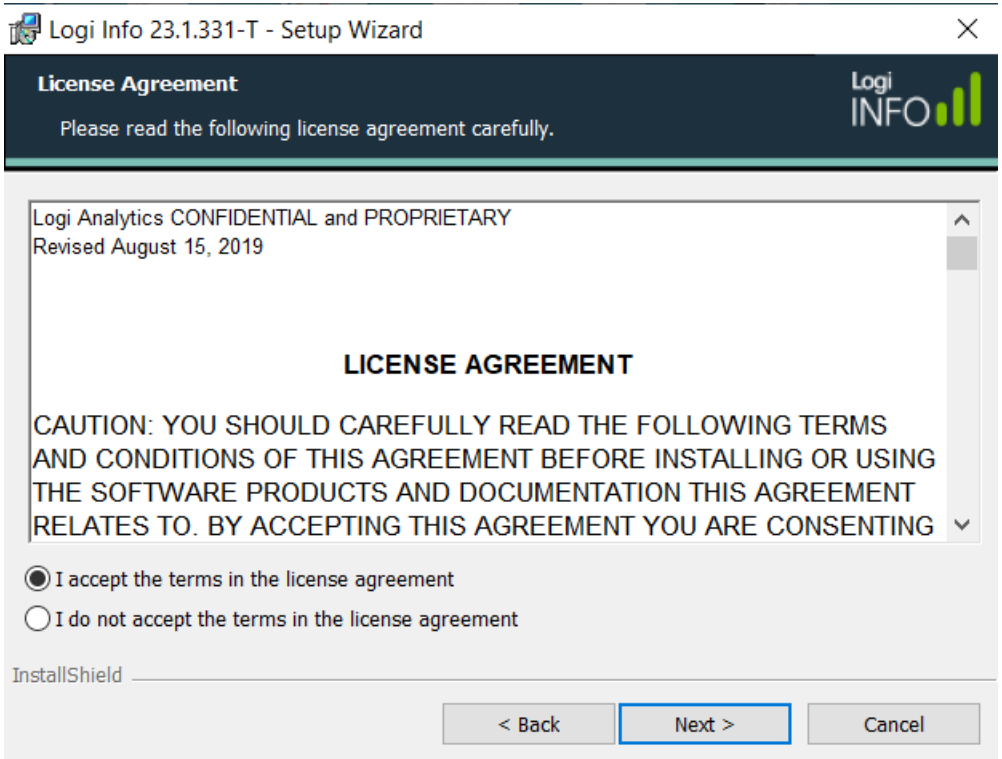
This topic describes the installation steps and the dialog boxes you'll see when using the interactive tool built into your Logi product installation file. If you need a "silent" (non-interactive) installation, see "Command Line Install" on page 327.

1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.

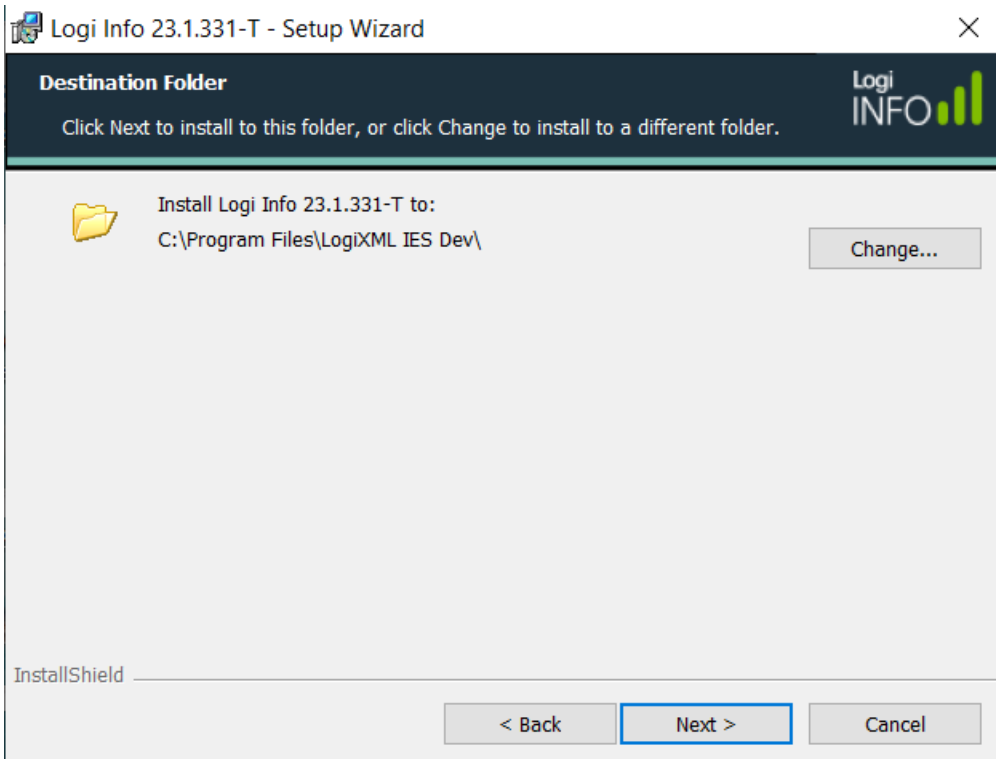


2. When the **Welcome Screen** appears, select **Next:**

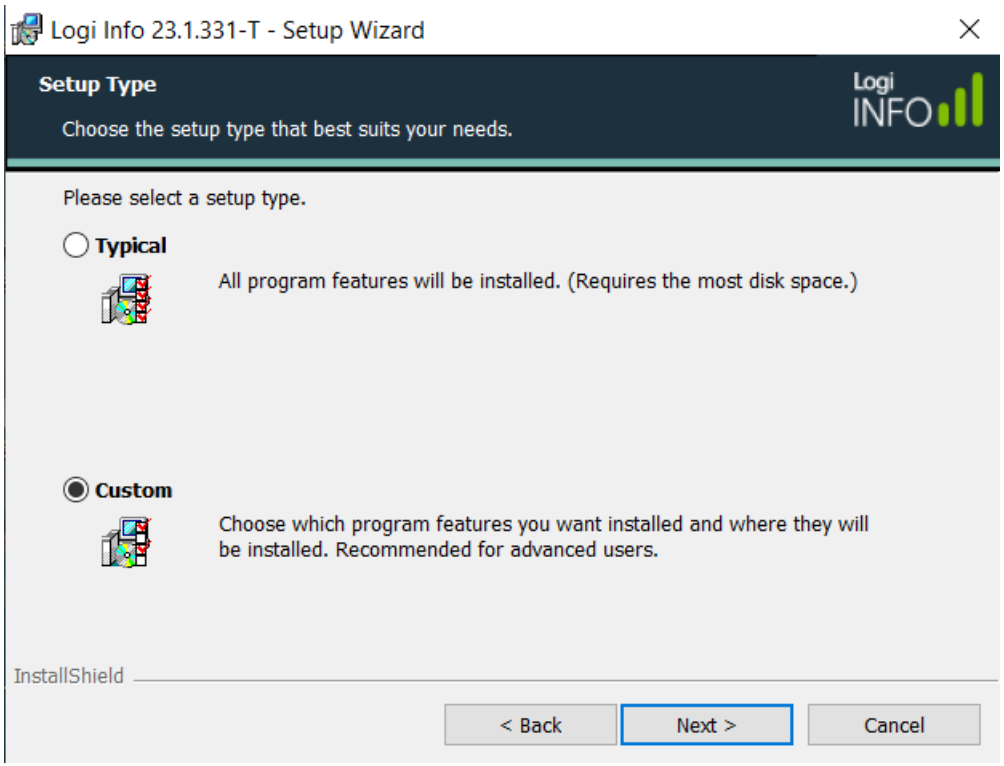
3. **License Agreement**: Select the "**I accept the terms...**" radio button after reading the license agreement and select **Next** to continue:



4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



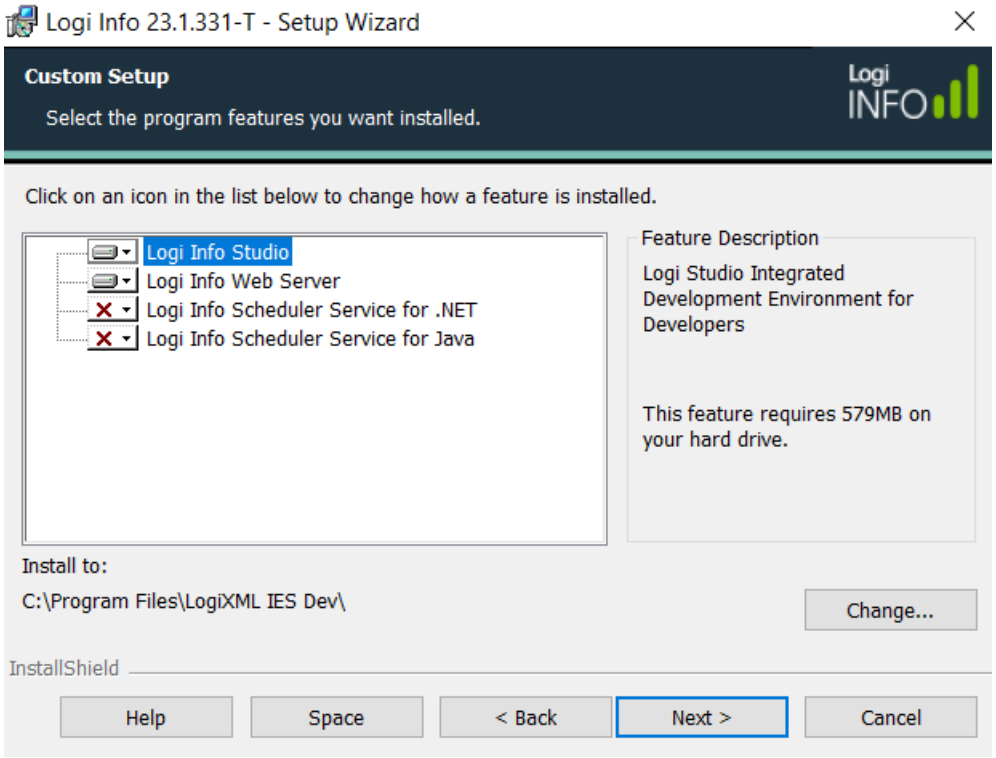
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

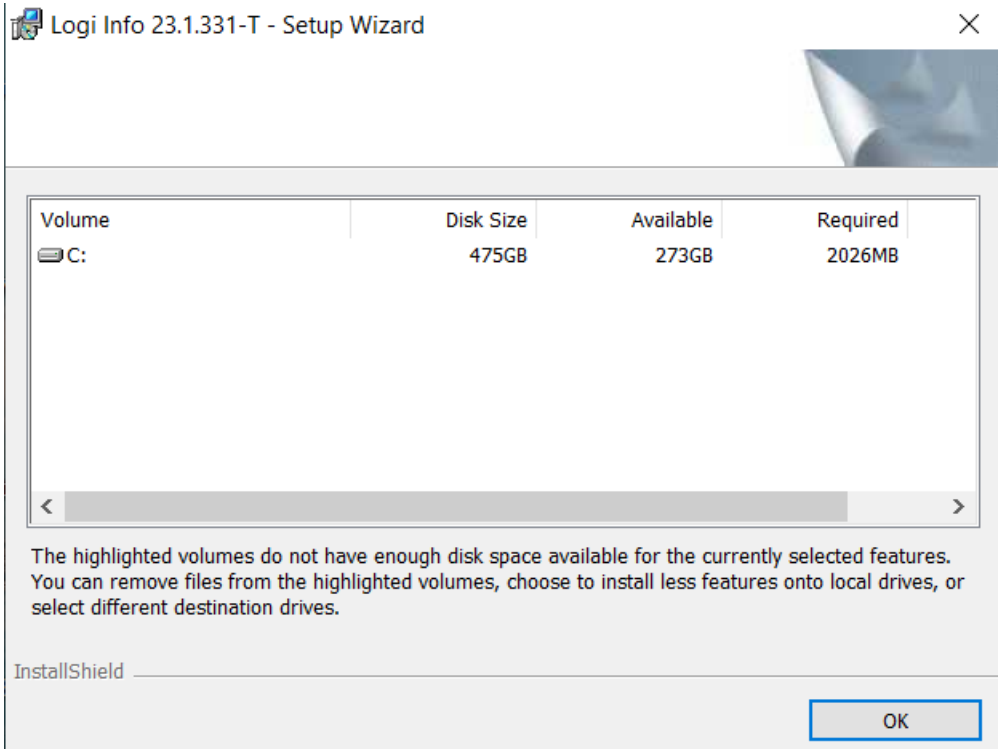
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

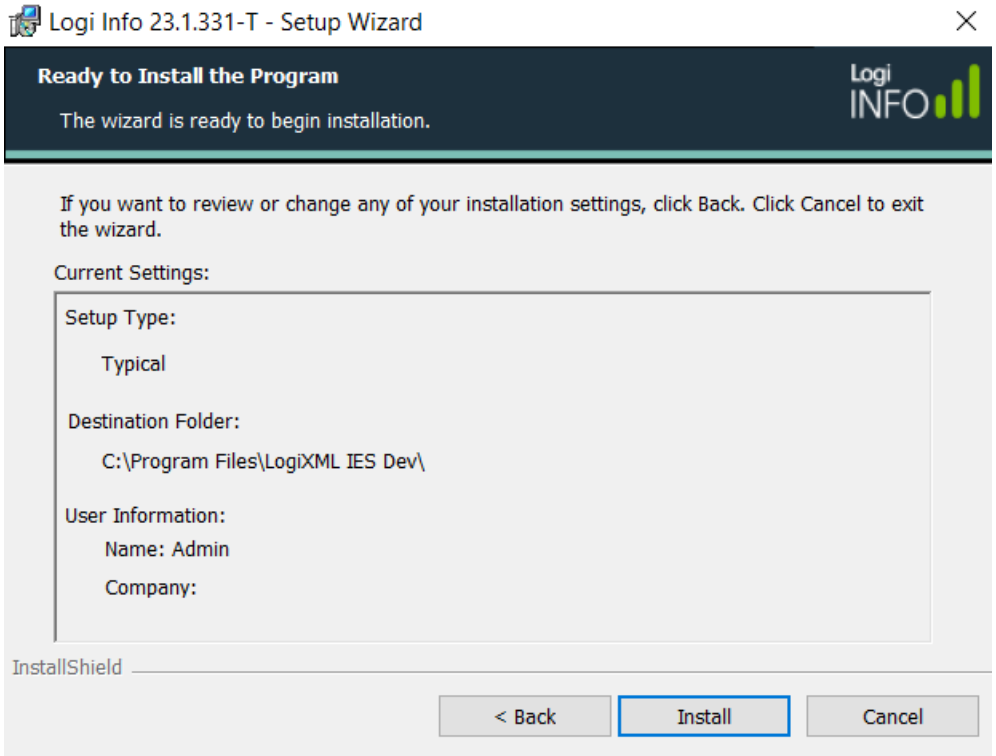
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



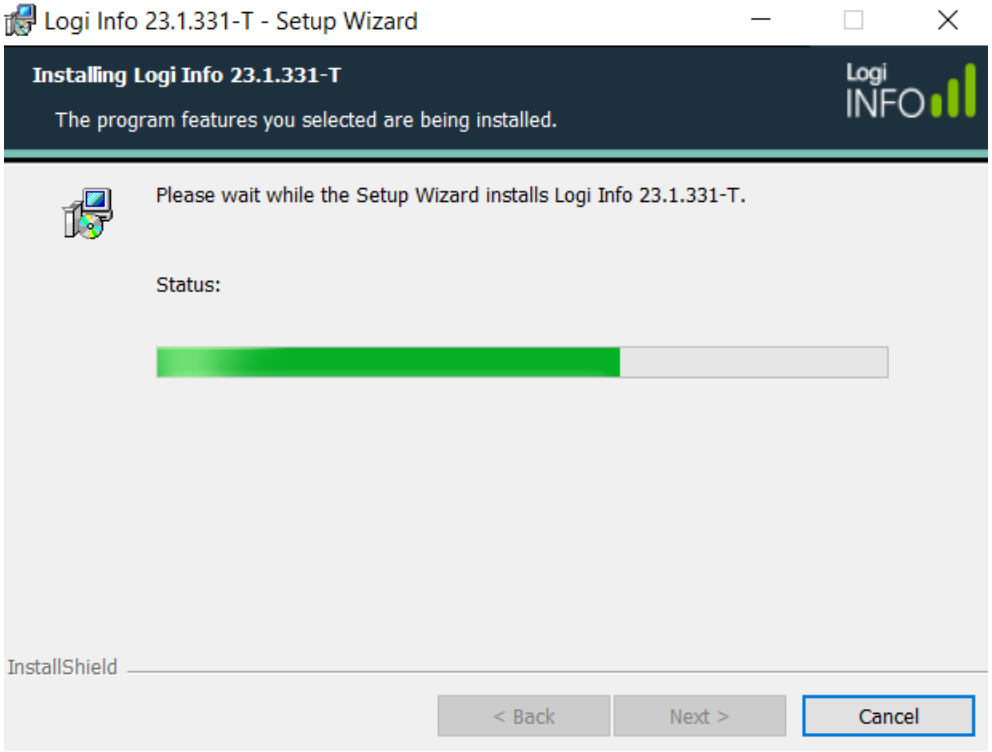
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



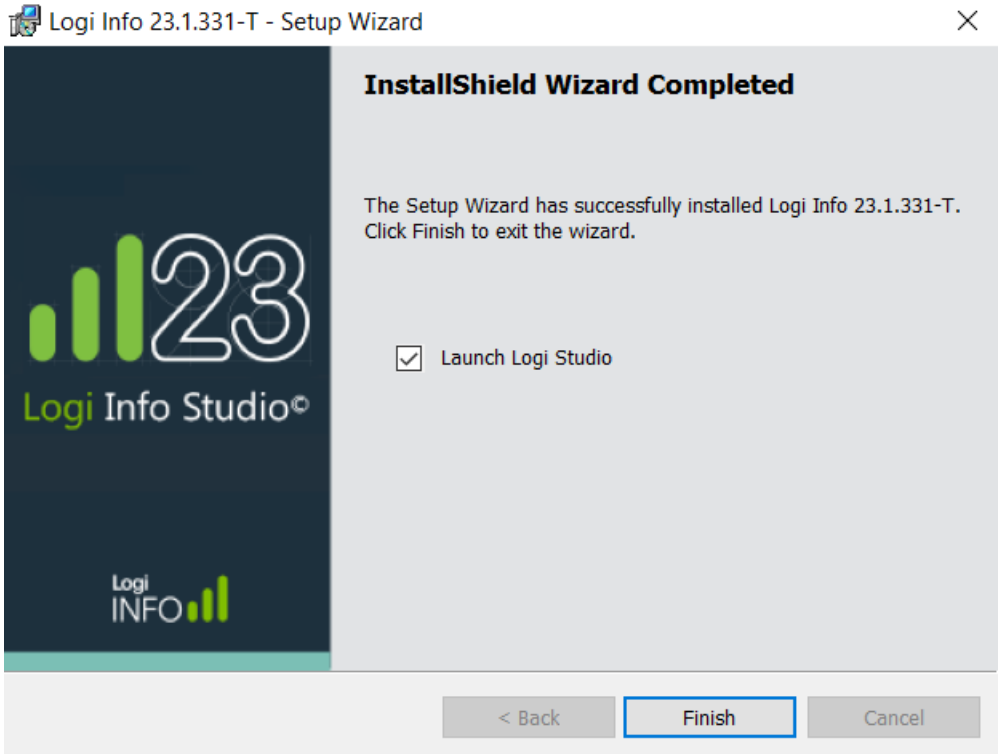
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).



9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

You can also launch Studio using the Start menu.

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in Using Logi Studio.

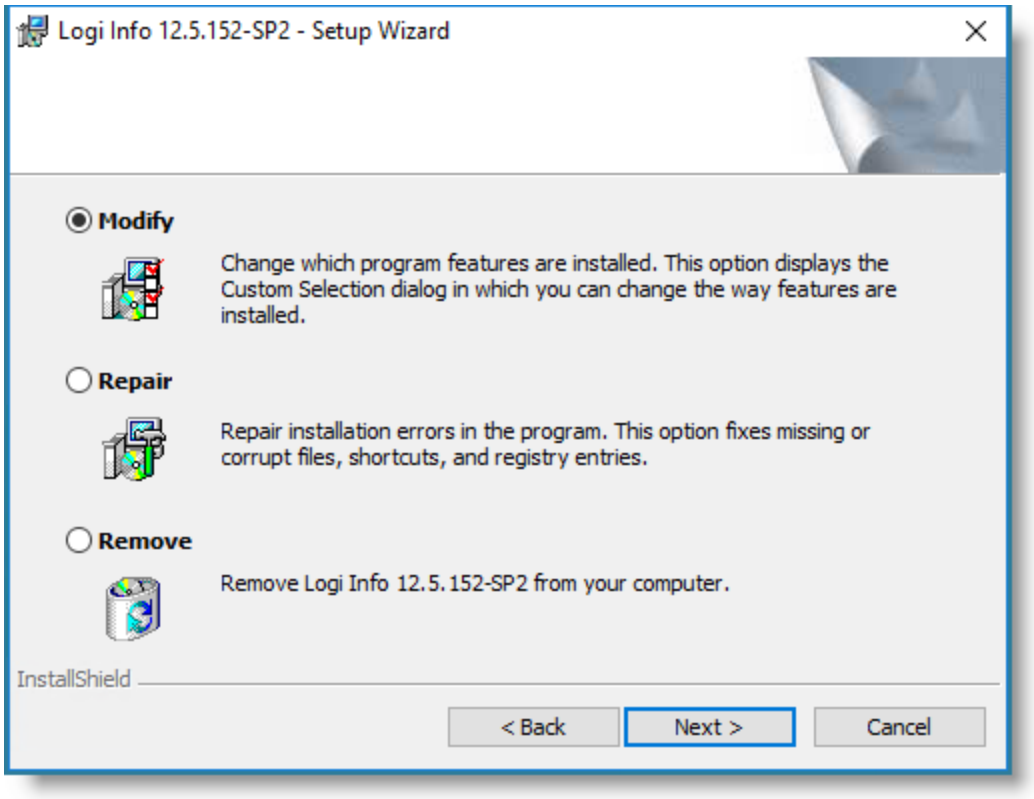
Installation is complete and you may begin to use Studio and/or the Server Engine immediately.

Install Logi Info on Windows Server 2016 - Installing Add-on Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. *Add-on Modules* describes them and includes links to their installation topics.

Install Logi Info on Windows Server 2016 - Modifying or Repairing an Installation

Suppose you installed Logi Info but didn't initially install the Scheduler, and now you find you want to schedule reports. Or you suspect a .DLL file is missing or is corrupted and you want to fix it.



These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation .exe file, using "Run As Administrator". *Do not* use Control Panel → Add-Remove Programs to do this; it will request an .msi file, which is not retained after the original installation.

Install Logi Info on Windows Server 2012 r2

Welcome to Logi Analytics reporting products.

The following topics guide you through installing Logi Info on a single computer under the Windows Server 2012 r2 operating system, for use with the IIS web server:

Install Logi Info on Windows Server 2012 r2 - Installation Scenarios

There are two major parts to the Logi product in the installation file you downloaded:

Logi Studio, our development environment, is a Windows application that's typically installed on a Windows development machine and interacts with the local IIS web server for Logi application development and testing.

The **LogiServer Engine** is a set of files that's part of each Logi application and which provides an extension to the IIS web server at runtime. When you build a Logi application, Studio adds the Engine files to the application. The engine also includes a Windows utility application, **Logi Server Manager**, that allows you to perform basic configuration of Logi applications without using Studio.

A very typical installation scenario is to install Studio and the Server Engine on a development machine, and then install only the Server Engine (including Server Manager) on the production web server. This "development and production" dual installation is allowed by our licensing scheme.

Additional copies of Studio may be licensed and installed on additional development machines.

There are a number of ways to *deploy* your Logi applications to production, including using Studio's built-in **Application Deployment** tool.

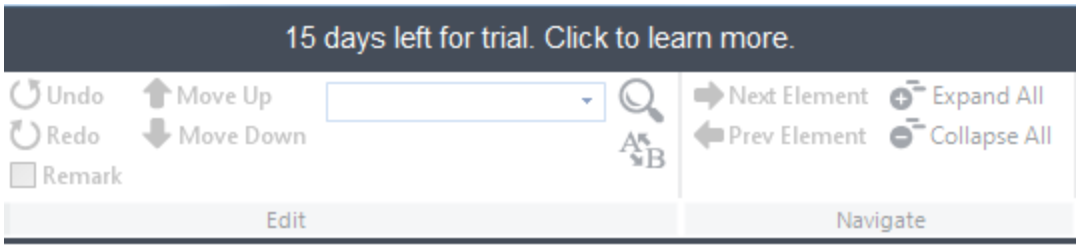
Other installation scenarios, involving shared network drives and team development, can be used. This topic covers installation of Studio and the Server Engine on a development machine, and also explains what's needed to install the Server Engine on a single production server.

64-bit Only

Windows Server 2012 R2 is a **64-bit** operating system. There are different Logi product distribution files for 32-bit and 64-bit versions, so be sure you've downloaded the 64-bit version.

Install Logi Info on Windows Server 2012 r2 - Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You need do nothing but install the product and you can begin using it. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about **purchasing** a Logi Info license.

After the trial period expires, Studio and any Logi reports you may have developed will *no longer run* without a real license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server.

Our licensing scheme allows you to deploy our product on one development machine and on one production server. Additional separate licenses for Studio, for additional developers, are also available.

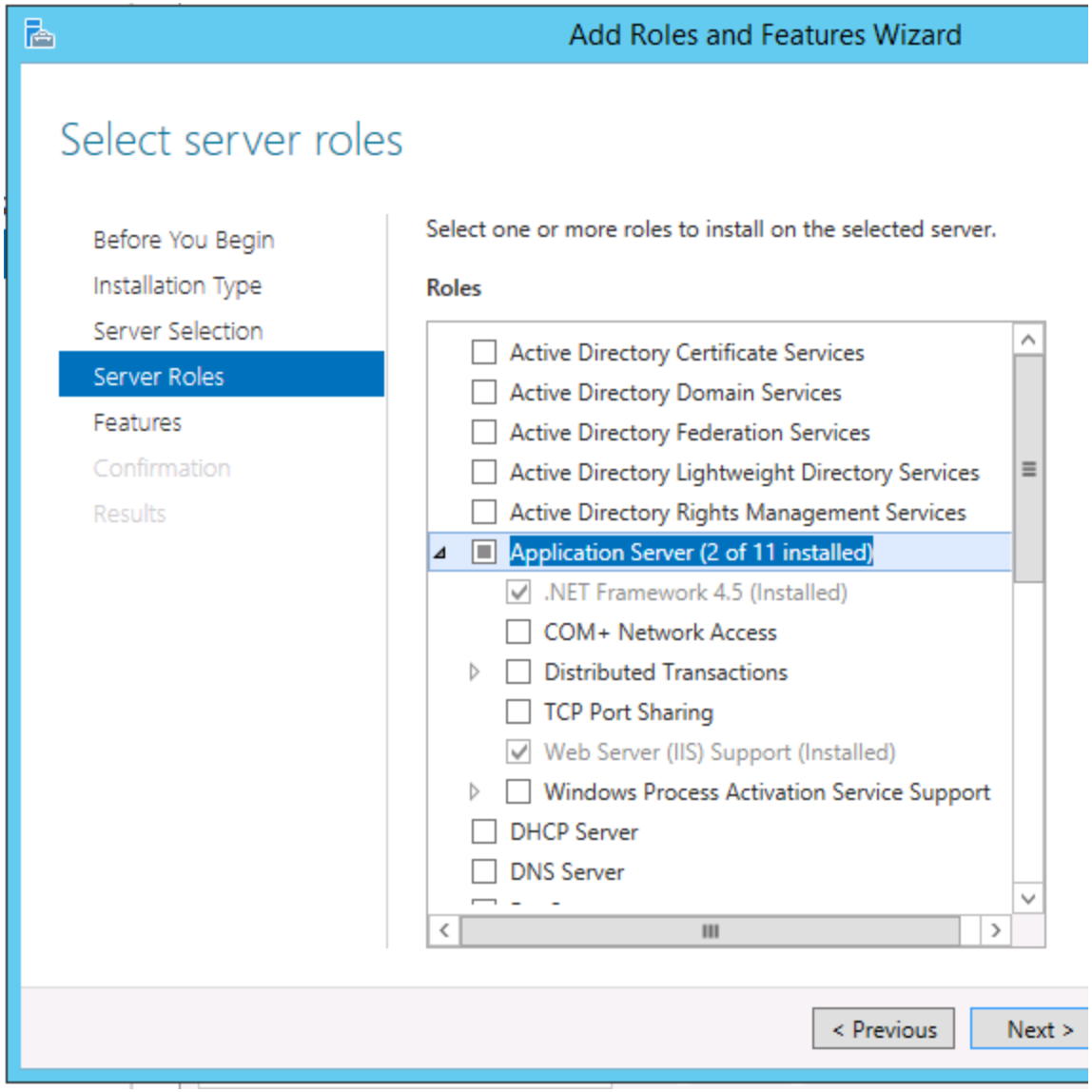
Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Install Logi Info on Windows Server 2012 r2 - Preparing to Install

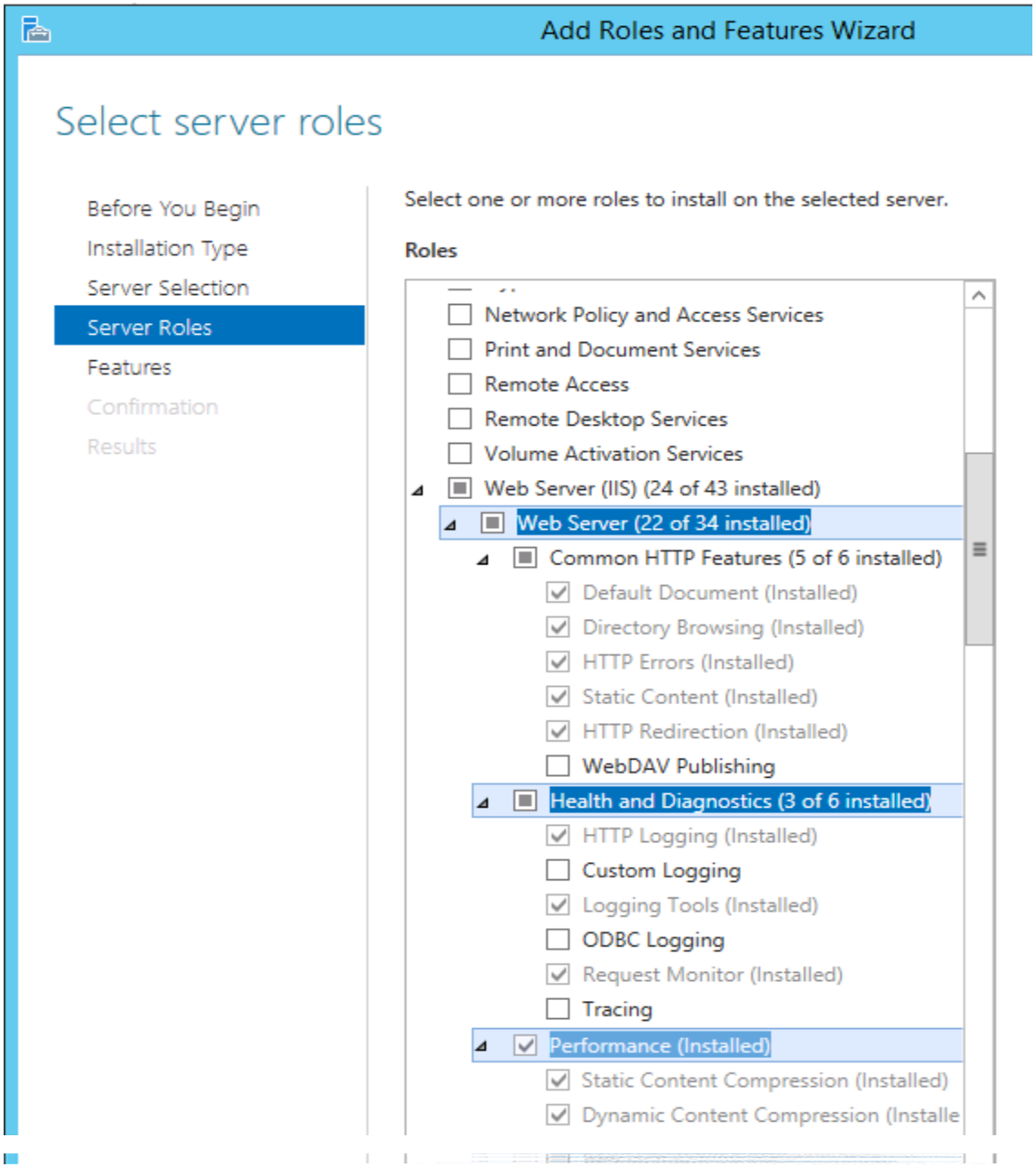
Administrator privileges on the target computer are **required** to complete the installation of Logi reporting tools. Logi Info for the Windows environment requires the .NET Framework 4.x. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Roles and Features

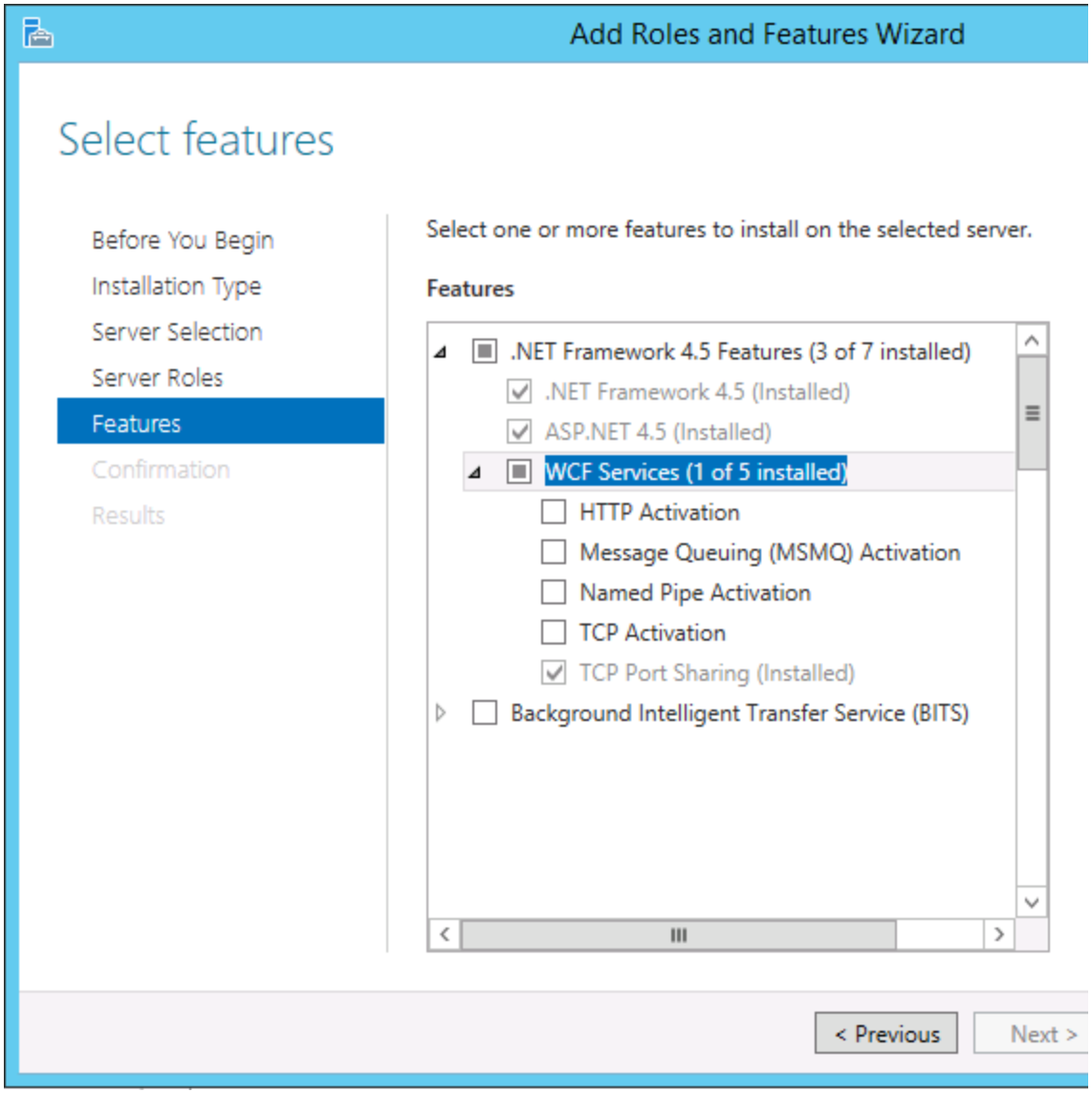
Before Logi product installation, you must ensure that the appropriate IIS roles and features have been enabled.



Use the Server Management tool to examine the server roles and ensure that, under the Application Server role,the .NET Framework and Web Server (IIS) Support roles are installed and enabled, as shown above.


Add Roles and Features Wizard
Select server roles
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Network Policy and Access Services
Print and Document Services
Remote Access
Remote Desktop Services
Volume Activation Services
Web Server (IIS) (24 of 43 installed)
Web Server (22 of 34 installed)
Common HTTP Features (5 of 6 installed)
Default Document (Installed)
Directory Browsing (Installed)
HTTP Errors (Installed)
Static Content (Installed)
HTTP Redirection (Installed)
WebDAV Publishing
Health and Diagnostics (3 of 6 installed)
HTTP Logging (Installed)
Custom Logging
Logging Tools (Installed)
ODBC Logging
Request Monitor (Installed)
Tracing
Performance (Installed)
Static Content Compression (Installed)
Dynamic Content Compression (Installe

Examine the Web Server (IIS) Role and ensure that the features shown in the enhanced image above are installed and enabled.



Examine the .NET Framework Featuresand ensure that the features shown above are installed and enabled.

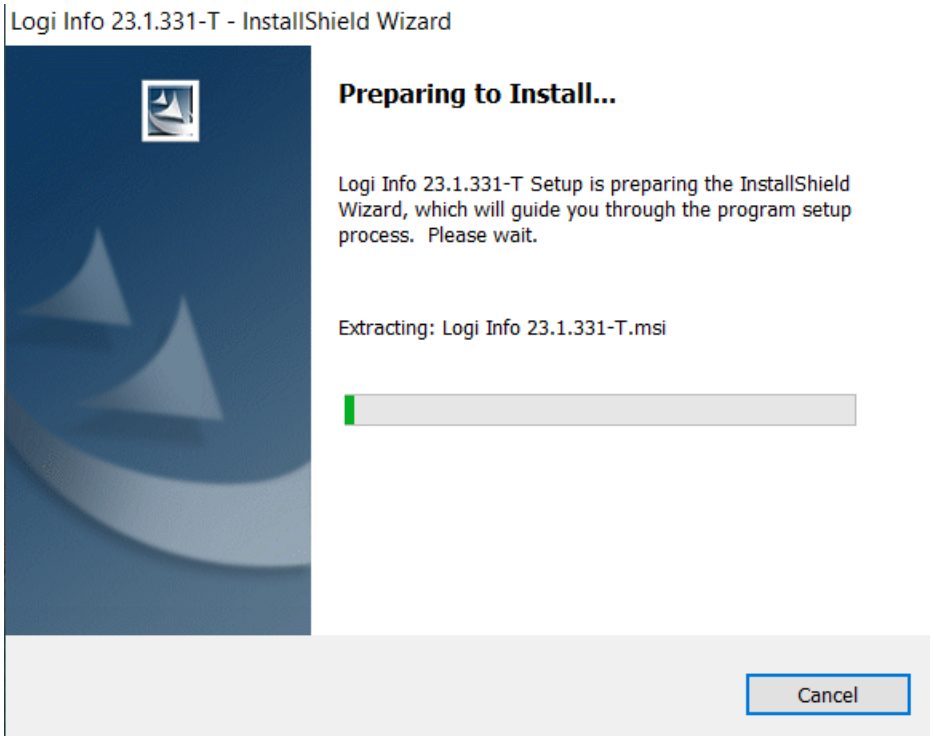
The IIS "Default Web Site"

When it's installed, IIS creates a "Default Web Site" and the New Application wizard in Logi Studio expects to create all new Logi applications as virtual directories of that web site. If you have renamed, replaced, or disabled the "Default Web Site", or have installed another web server that handles HTTP requests on Port 80, the wizard will fail during its application registration phase. Under these circumstances, you can continue to use Logi Studio to develop Logi applications but you will need to manually register them. This process is described in *Windows IISConfiguration*.

Install Logi Info on Windows Server 2012 r2 - Installing the Software

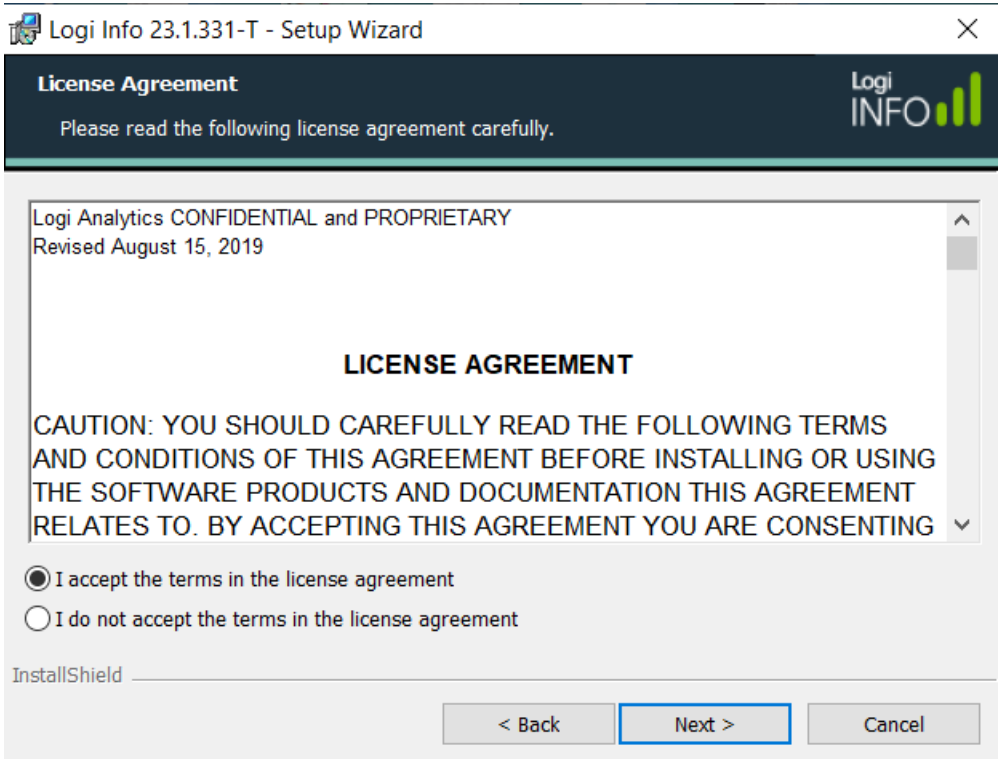
This topic describes the installation steps and the dialog boxes you'll see when using the interactive tool built into your Logi product installation file. If you need a "silent" (non-interactive) installation, see "Command Line Install" on page 327.

1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.

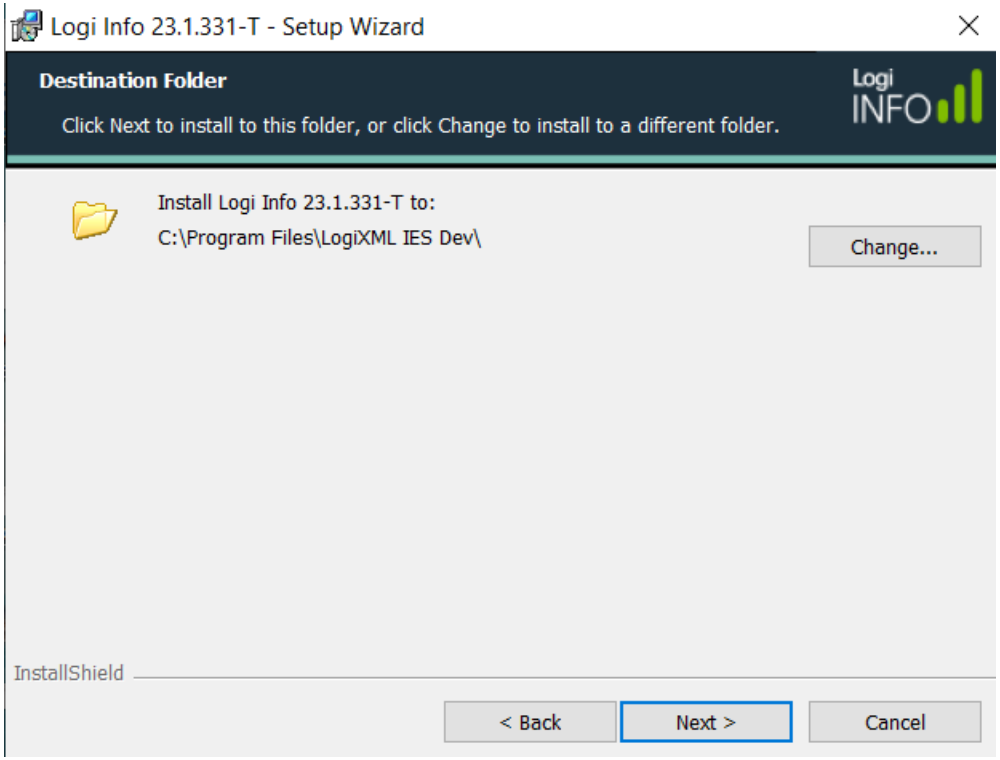


2. When the **Welcome Screen** appears, select **Next:**

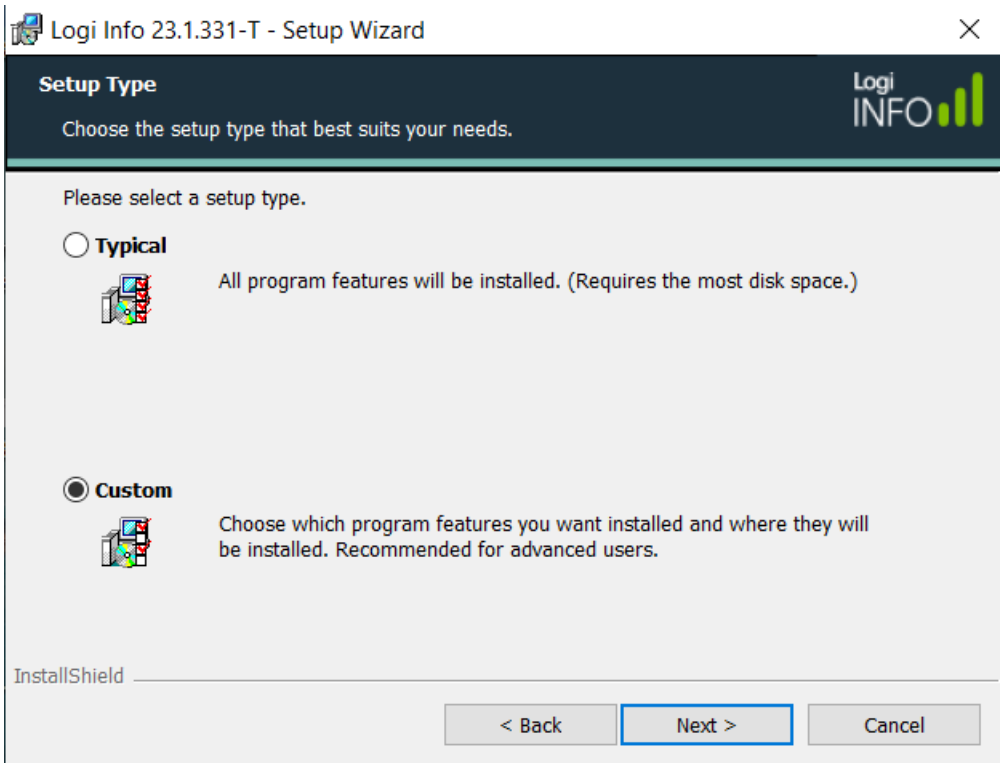
3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and select **Next** to continue:



4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



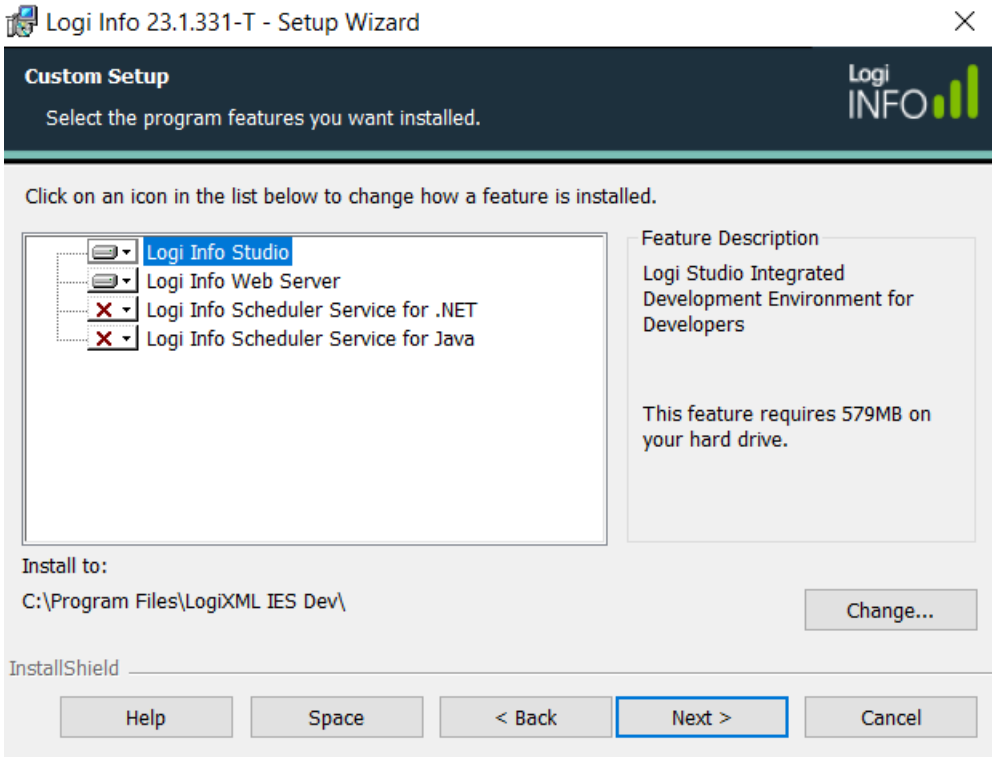
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

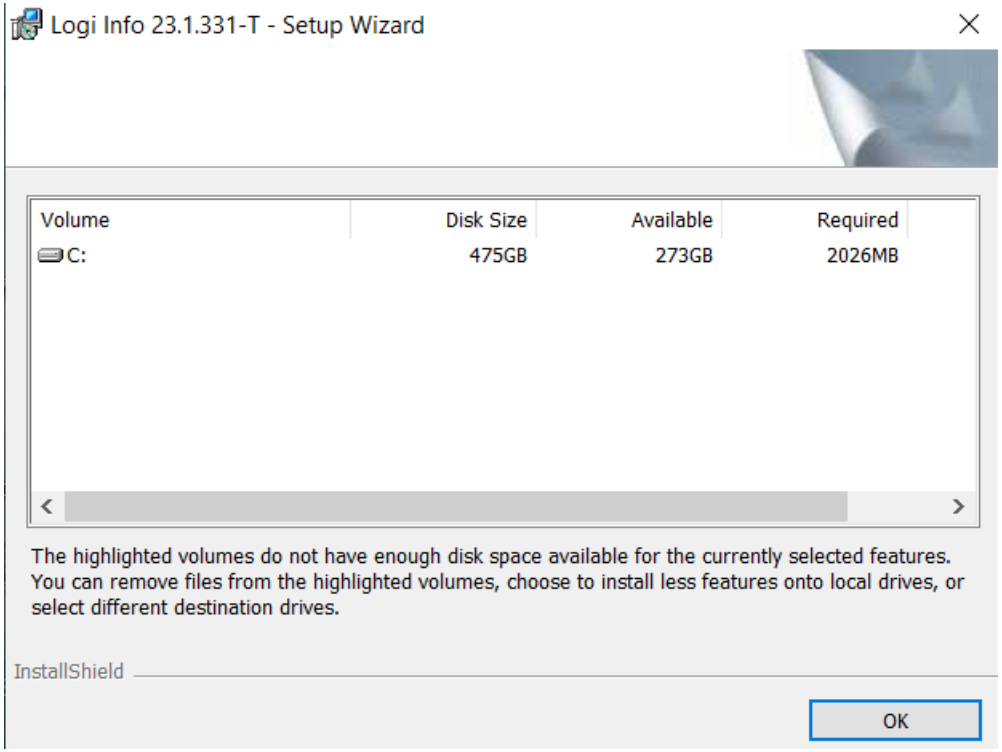
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

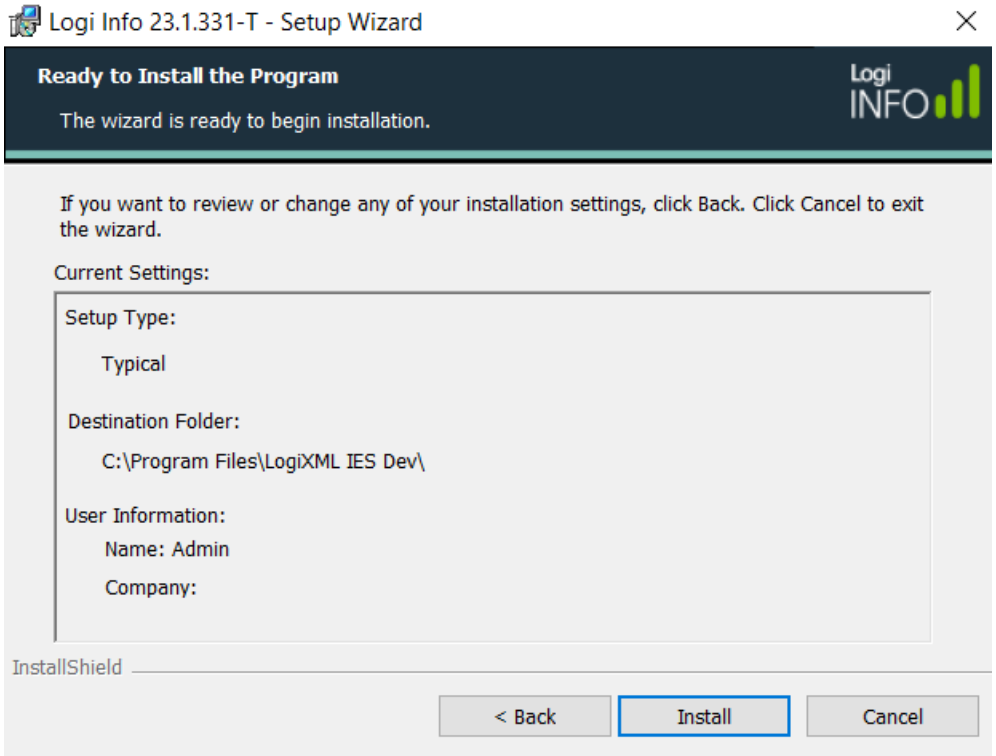
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



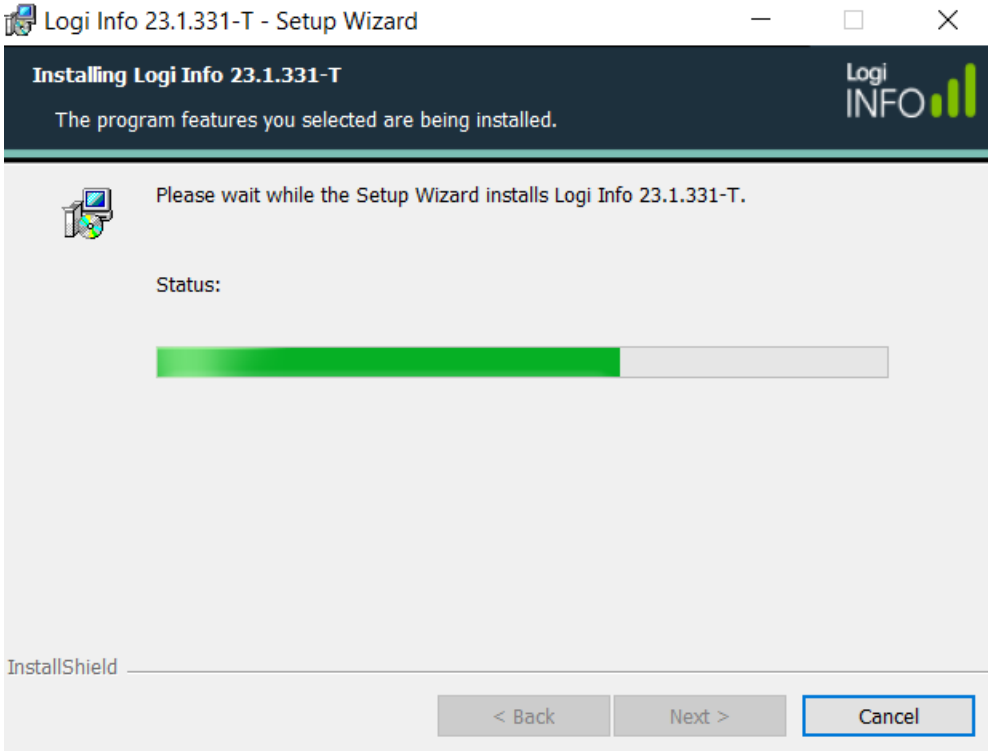
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



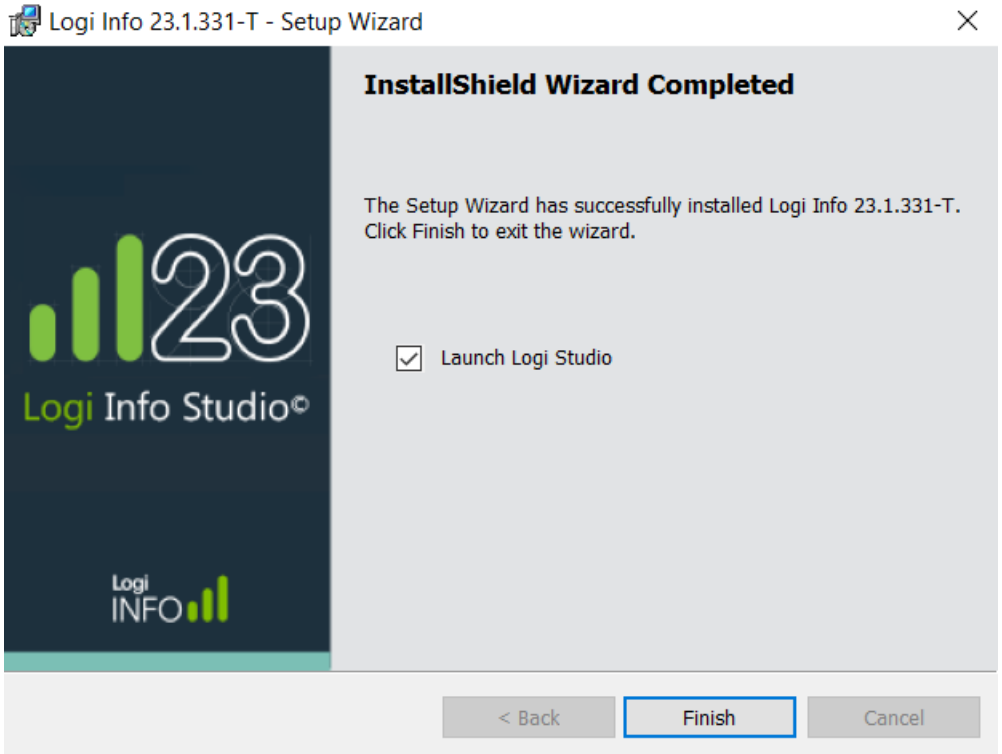
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).



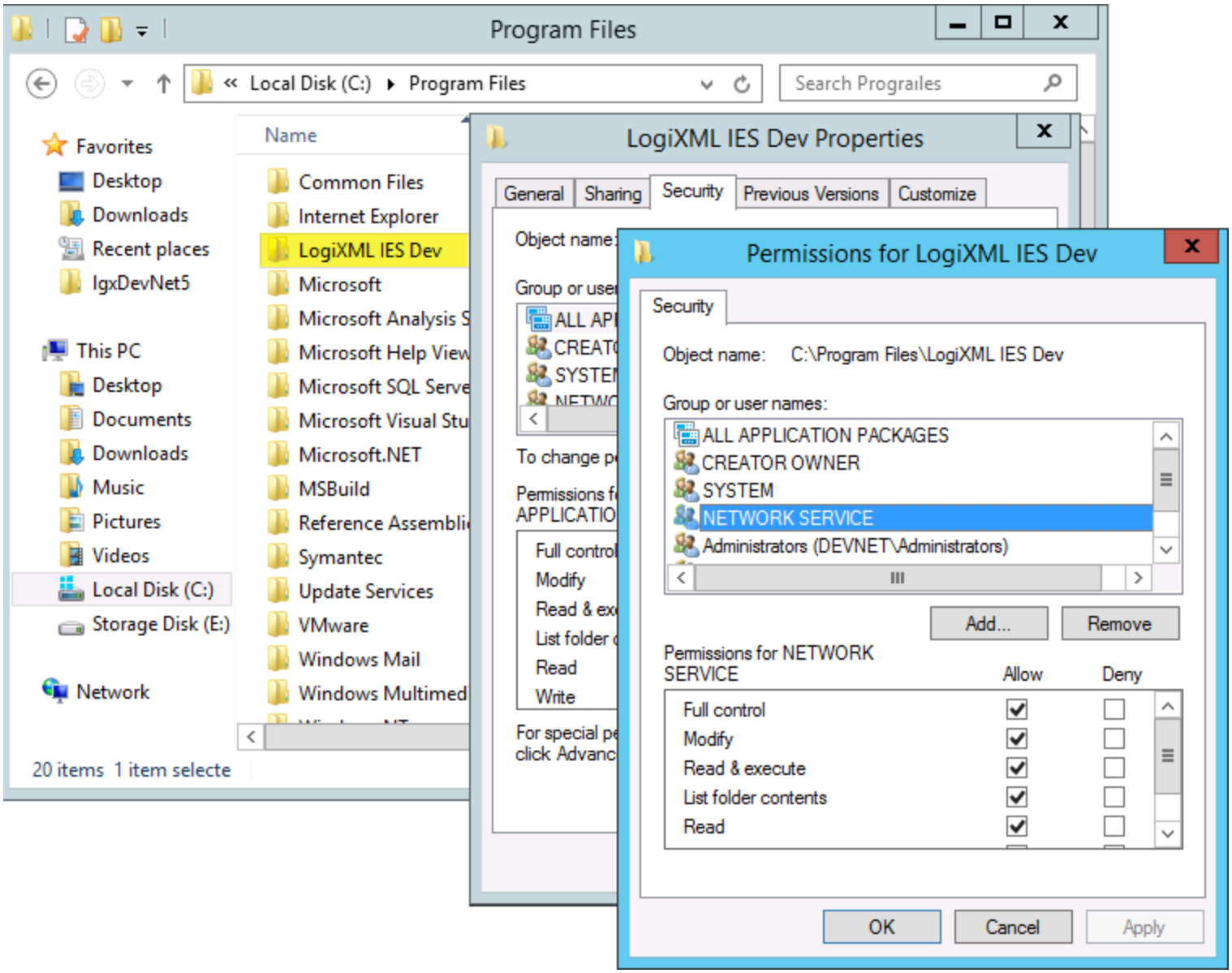
9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

You can also launch Studio using the Start menu.

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in Using Logi Studio.

Install Logi Info on Windows Server 2012 r2 - Setting File Access Permissions

You will need to set file access permissions to allow Logi Studio and IIS to work together smoothly:



Set the permissions, as shown above, using these steps:

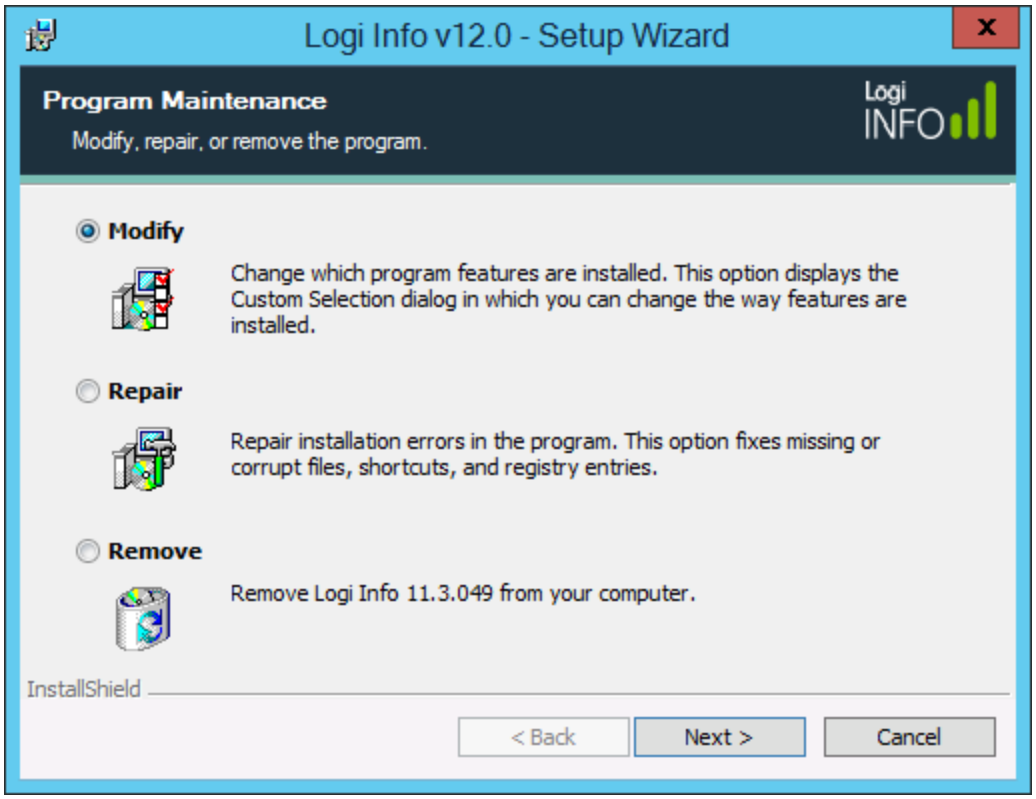
1. Use the File Explorer to navigate to `C:\Program Files\LogiXML IES Dev` and right-click the folder. Select *Properties*.
2. Select the Security tab and click Edit, the Add, and add the account NETWORK SERVICE.
3. Set the Permissions for NETWORK SERVICE to Full Control.
4. Click OK as necessary to exit.

Install Logi Info on Windows Server 2012 r2 - Installing Add-on Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. *Add-on Modules* describes them and includes links to their installation topics.

Install Logi Info on Windows Server 2012 r2 - Modifying or Repairing an Installation

Suppose you installed Logi Info but didn't initially install the Scheduler, and now you find you want to schedule reports. Or you suspect a .DLL file is missing or is corrupted and you want to fix it.



These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation .exe file, using "Run As Administrator". *Do not* use Control Panel→Add-Remove Programs to do this; it will request an .msi file, which is not retained after the original installation.

Install Logi Info on Windows 10

Welcome to Logi Analytics reporting products.

The following topics guide you through installing Logi Info v12 on a single computer under the Windows 10, for use with the IIS web server:

Install Logi Info on Windows 10 - Installation Scenarios

There are two major parts to the Logi product in the installation file you downloaded:

Logi Studio, our development environment, is a Windows application that's typically installed on a Windows development machine and interacts with the local IIS web server for Logi application development and testing.

The **LogiServer Engine** is a set of files that's part of each Logi application and which provides an extension to the IIS web server at runtime. When you build a Logi application, Studio adds the Engine files to the application. The engine also includes a Windows utility application, **Logi Server Manager**, that allows you to perform basic configuration of Logi applications without using Studio.

A very typical installation scenario is to install Studio and the Server Engine on a development machine, and then install only the Server Engine (including Server Manager) on the production web server. This "development and production" dual installation is allowed by our licensing scheme.

Additional copies of Studio may be licensed and installed on additional development machines.

There are a number of ways to *deploy* your Logi applications to production, including using Studio's built-in **Application Deployment** tool.

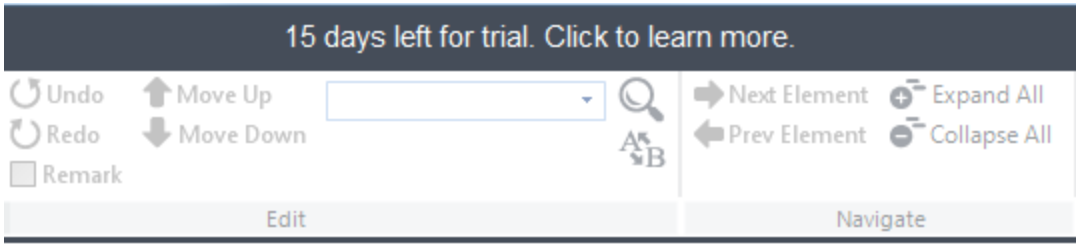
Other installation scenarios, involving shared network drives and team development, can be used. This topic covers installation of Studio and the Server Engine on a development machine, and also explains what's needed to install the Server Engine on a single production server.

32- or 64-bit?

Logi Info versions earlier than v12.2 are available for **32-bit** and **64-bit** systems. Starting with v12.2 only 64-bit versions are available. There are different Logi product distribution files for 32-bit and 64-bit versions, so be sure you've downloaded the right one for your system.

Install Logi Info on Windows 10 - Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You don't need to do anything but install the product and you can begin using it immediately. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about **purchasing** a Logi Info license.

After the trial period expires, Studio won't be usable and any Logi reports you may have developed will *no longer run* without a regular license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server. Our licensing scheme allows you to deploy our product on one development machine and on one production server. As mentioned earlier, additional separate licenses for Studio, for additional developers, are also available.

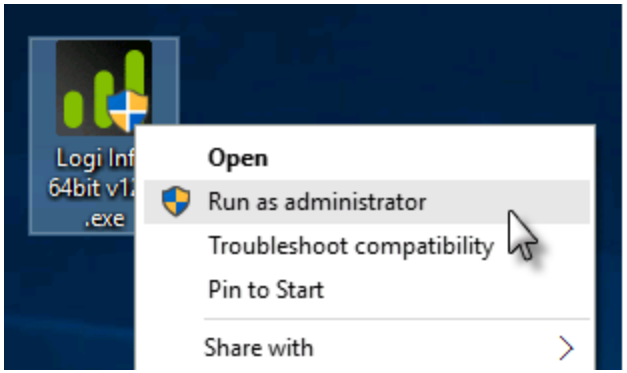
Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Install Logi Info on Windows 10 - Preparing to Install

Logi products work with *all* editions of Windows 10. This topic discusses features that impact the installation and/or operation of Logi Analytics products.

Use the Administrator Account

Windows includes a number of security enhancements that impact software installation and it is critical that installation and configuration occur while using the software with the built-in "Administrator" account.



Even if your personal account has been added to the local Administrators Group, it may not have sufficient privileges, so don't rely on it.

As shown at left, the correct practice when running the Logi installation program or using the Command Line to make configuration adjustments is to start the tool by *right-clicking* its icon and selecting "**Run as administrator**" from the menu to start the program. The ensures that appropriate permissions are provided for the installed components.

Don't see a "Run as administrator" option? If the system is in a network domain, your network admin may have created security policies that don't allow you to see this option, in which case you need to consult your IT staff for assistance.

.NET Framework

Logi products require the .NET Framework components. When Windows is installed, multiple versions of the framework are typically included. Before doing anything else, ensure that you have .NET installed by using File Explorer to browse to:

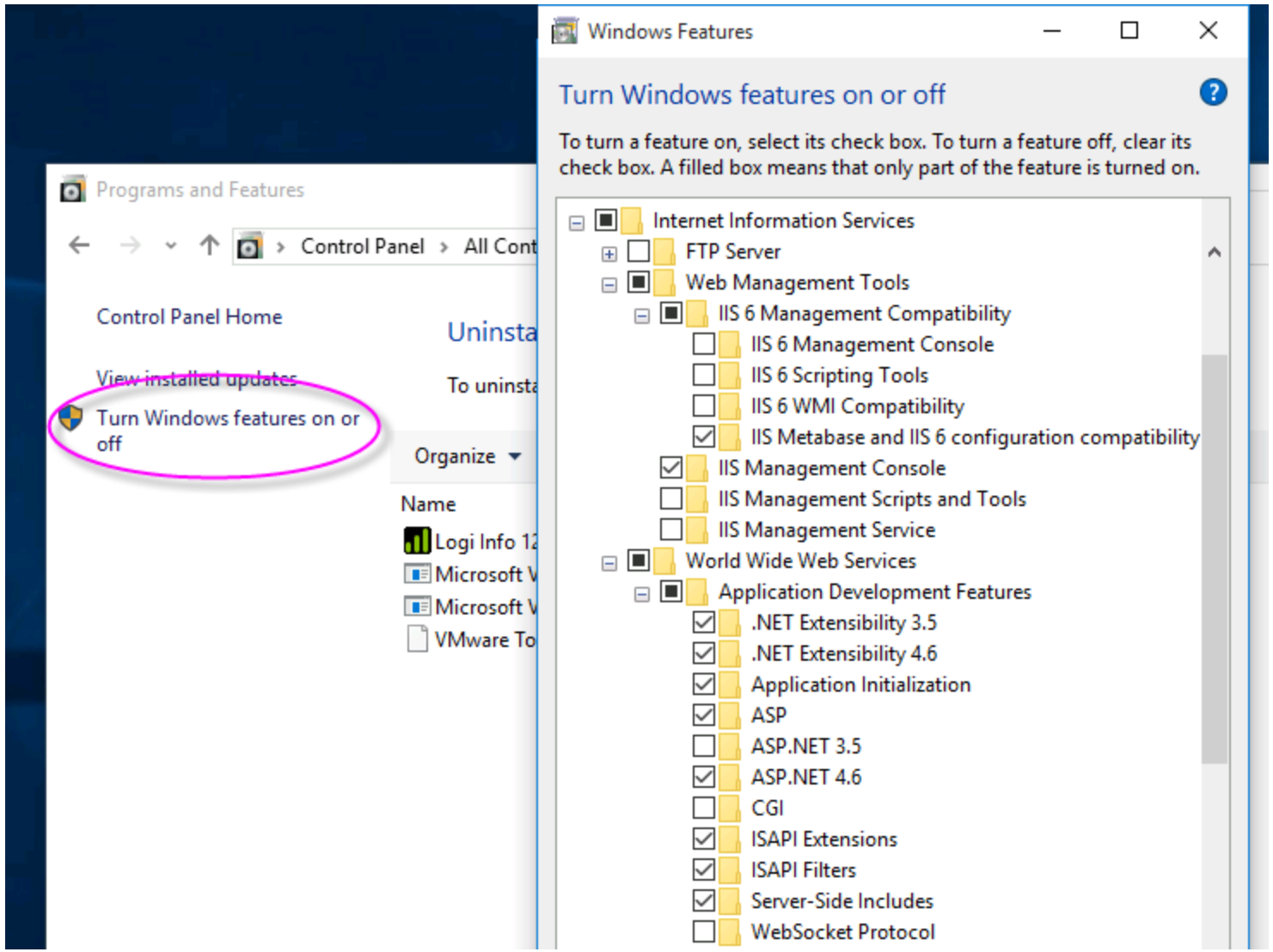
```
C:\Windows\Microsoft.NET\Framework64
```

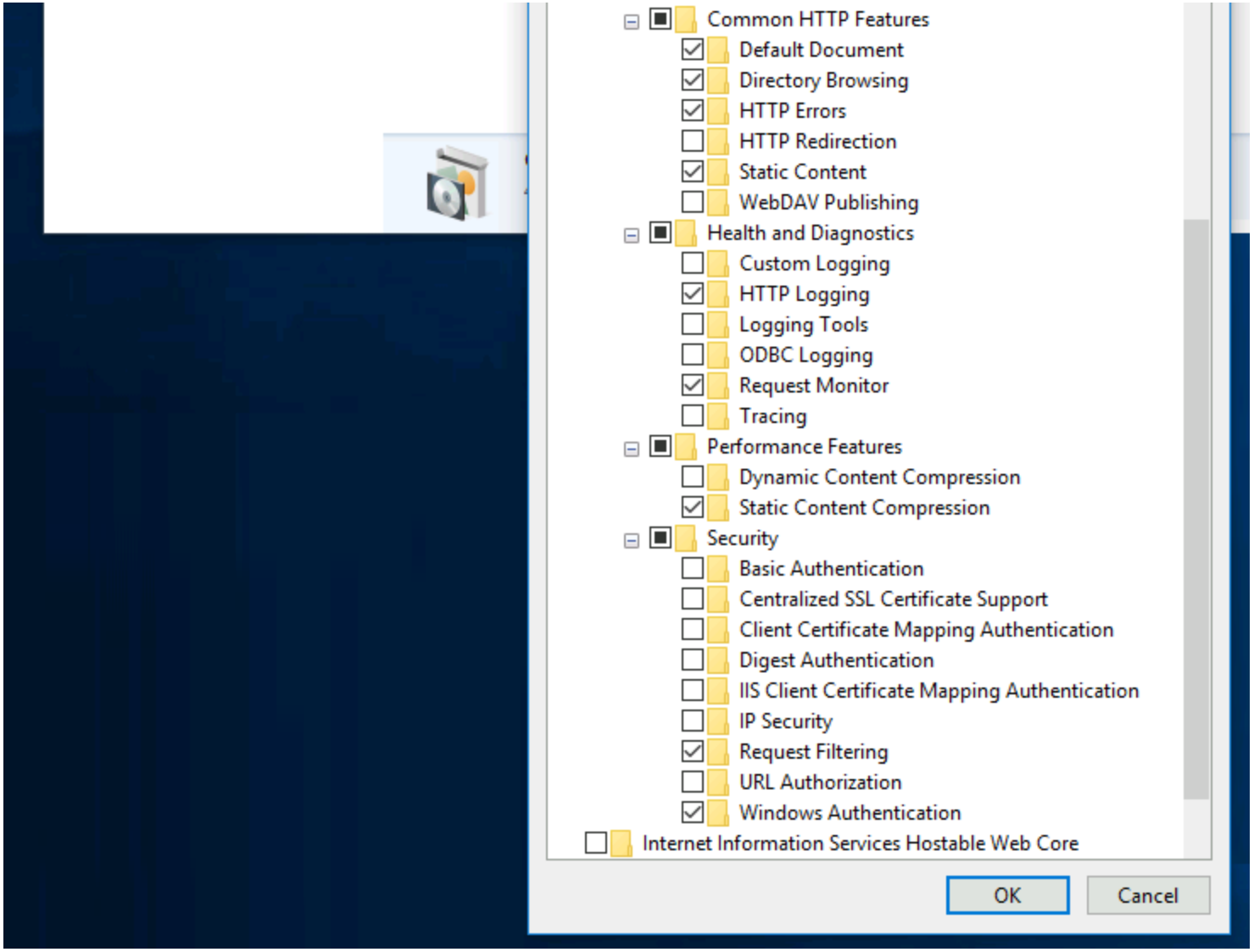
where you should see several folders, such as v4.6.393297, one for each version of .NET installed.

The .NET Framework 4.x is required. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Enable the IIS Web Server

When Windows is installed, the components for the **IIS web server** are installed but are usually not enabled. You must do this manually, *before* you install your Logi product.


Programs and Features
Control Panel › All Cont
Control Panel Home
View installed updates
Turn Windows features on or off
Uninsta
To uninsta
Organize
Name
Logi Info 12
Microsoft V
Microsoft V
VMware To
Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
Internet Information Services
FTP Server
Web Management Tools
IIS 6 Management Compatibility
IIS 6 Management Console
IIS 6 Scripting Tools
IIS 6 WMI Compatibility
IIS Metabase and IIS 6 configuration compatibility
IIS Management Console
IIS Management Scripts and Tools
IIS Management Service
World Wide Web Services
Application Development Features
.NET Extensibility 3.5
.NET Extensibility 4.6
Application Initialization
ASP
ASP.NET 3.5
ASP.NET 4.6
CGI
ISAPI Extensions
ISAPI Filters
Server-Side Includes
WebSocket Protocol


Common HTTP Features
Default Document
Directory Browsing
HTTP Errors
HTTP Redirection
Static Content
WebDAV Publishing
Health and Diagnostics
Custom Logging
HTTP Logging
Logging Tools
ODBC Logging
Request Monitor
Tracing
Performance Features
Dynamic Content Compression
Static Content Compression
Security
Basic Authentication
Centralized SSL Certificate Support
Client Certificate Mapping Authentication
Digest Authentication
IIS Client Certificate Mapping Authentication
IP Security
Request Filtering
URL Authorization
Windows Authentication
Internet Information Services Hostable Web Core
OK
Cancel

This is accomplished in Control Panel, under Programs and Features → Turn Windows features on or off, as shown above in Windows 10. Check the boxes on your machine so they match the image above, then close the dialog boxes.

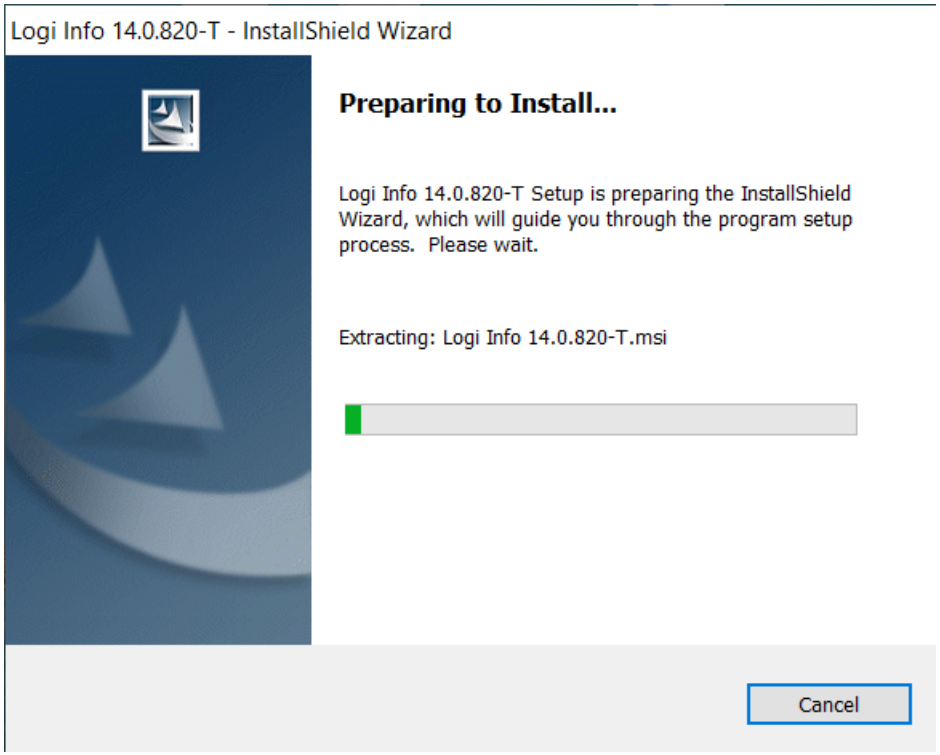
The IIS "Default Web Site"

When it's enabled, IIS creates a "Default Web Site" and the New Application wizard in Logi Studio expects to create all new Logi applications as virtual directories of that web site. If you have renamed, replaced, or disabled the "Default Web Site", or have installed another web server that handles HTTP requests on Port 80, the wizard will fail during its application registration phase. Under these circumstances, you can continue to use Logi Studio to develop Logi applications but you will need to manually register them. This process is described in Windows IIS Configuration.

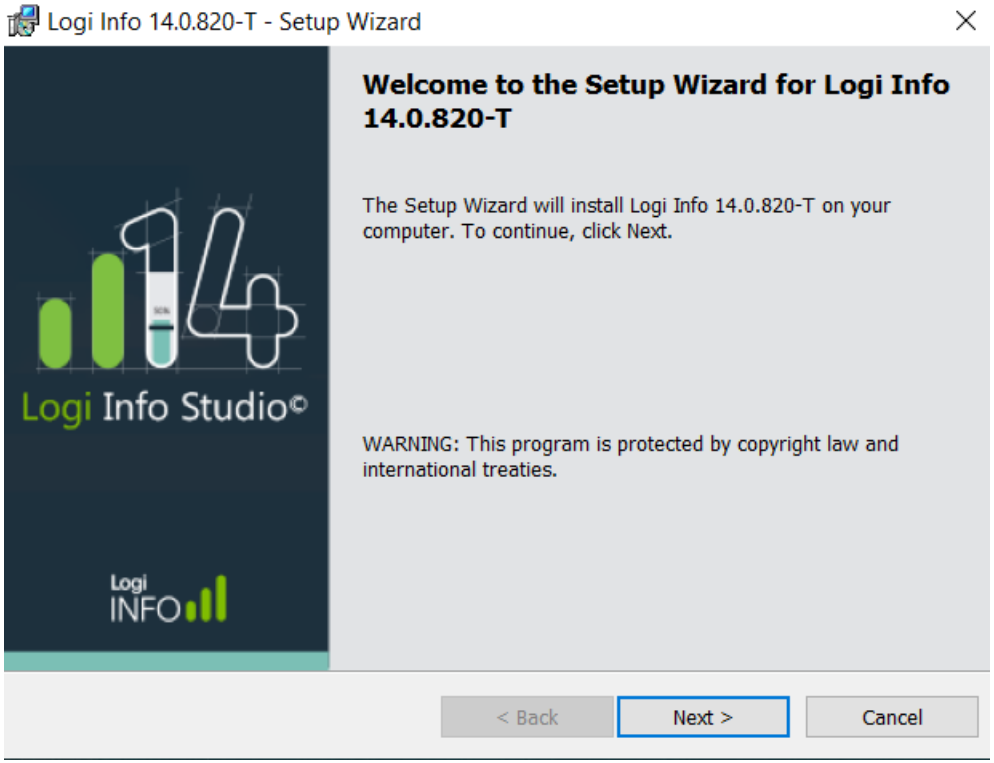
Install Logi Info on Windows 10 - Installing the Software

This topic describes the installation steps and the dialog boxes you'll see when using the interactive tool built into your Logi product installation file. If you need a "silent" (non-interactive) installation, see "Command Line Install" on page 327.

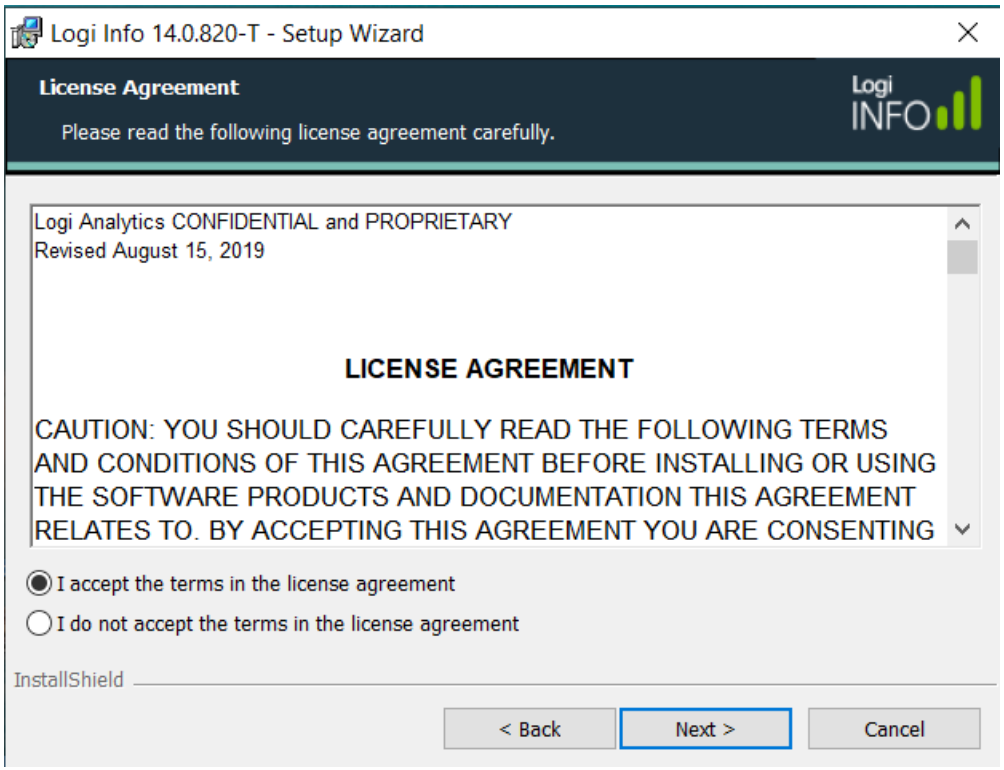
1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.



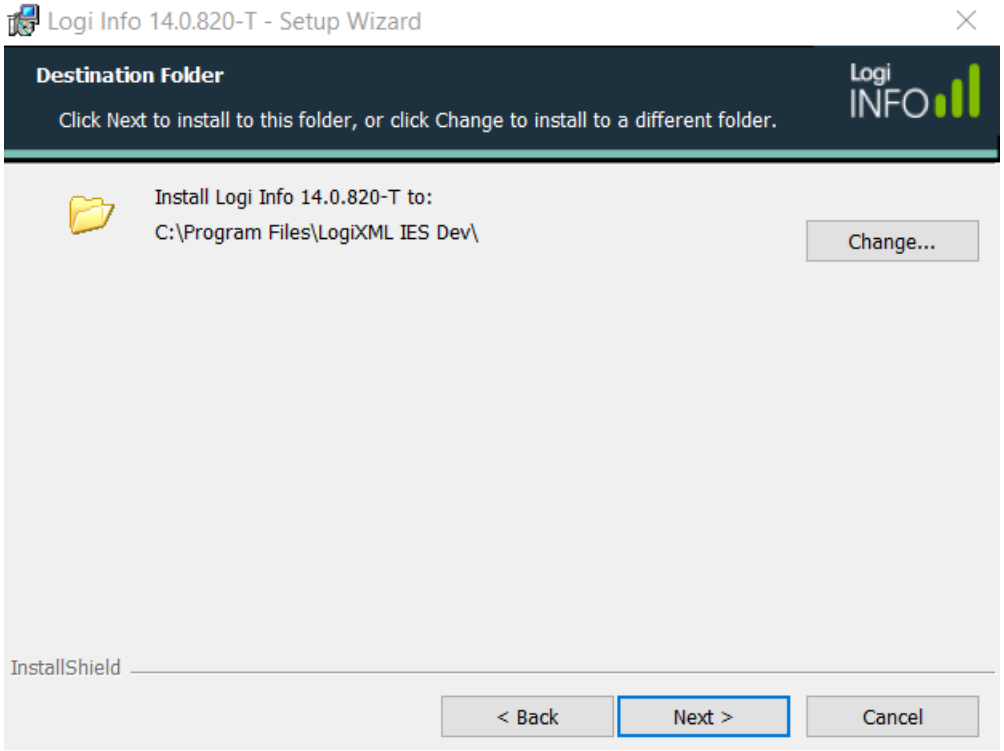
2. When the **Welcome Screen** appears, select **Next:**



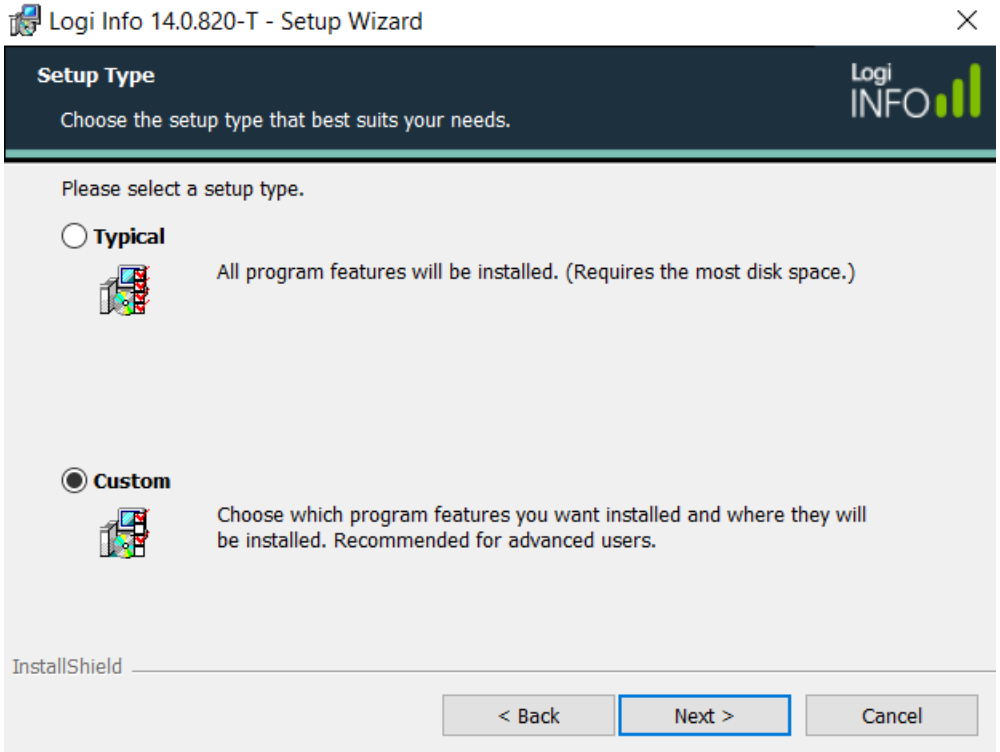
3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and select **Next** to continue:



4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



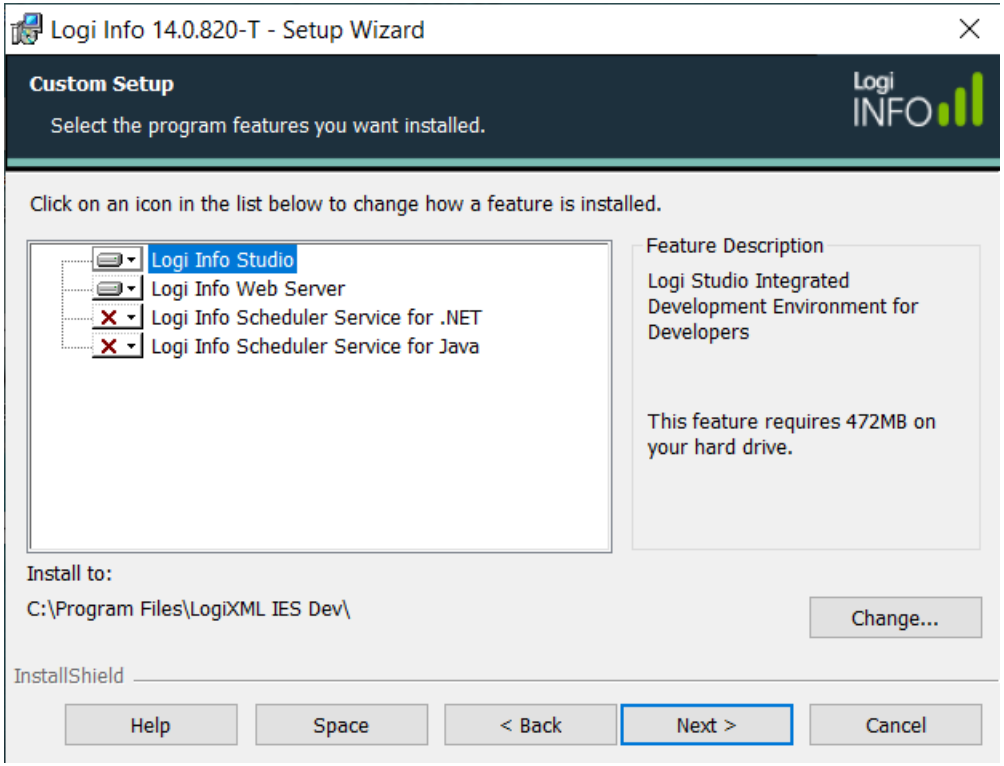
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

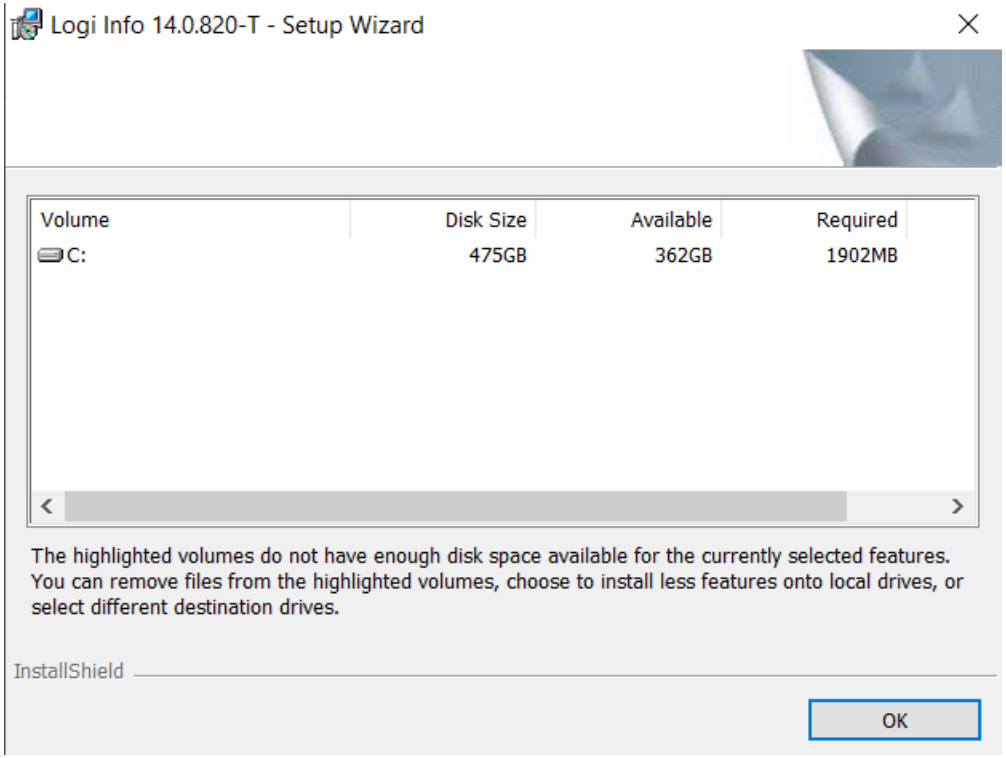
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

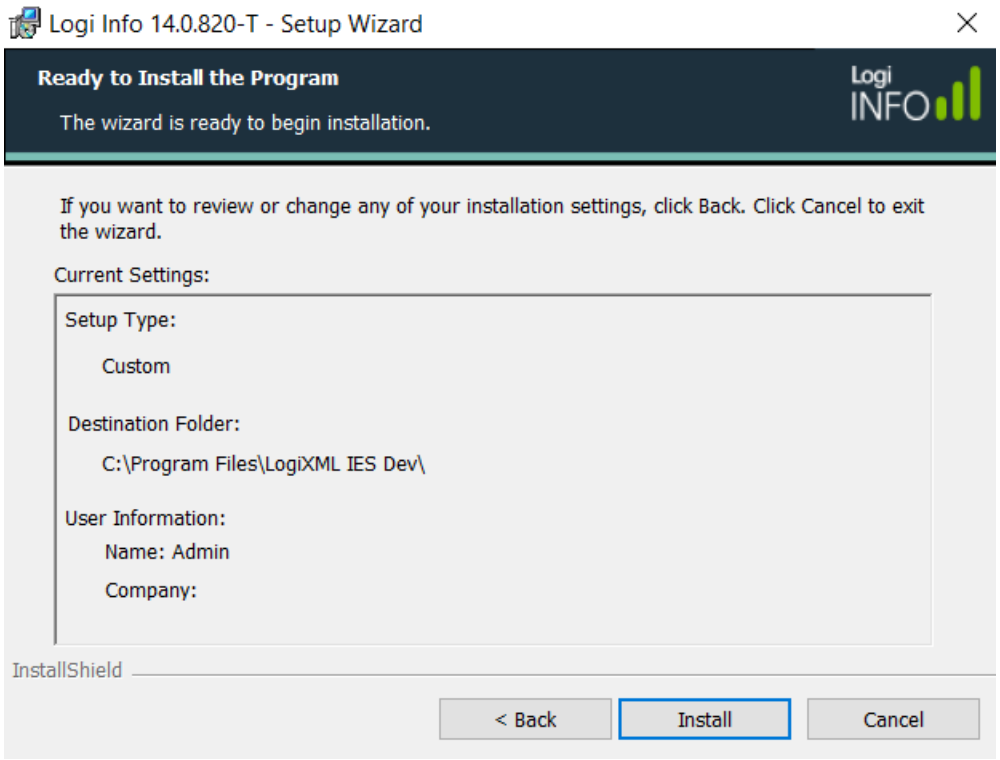
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



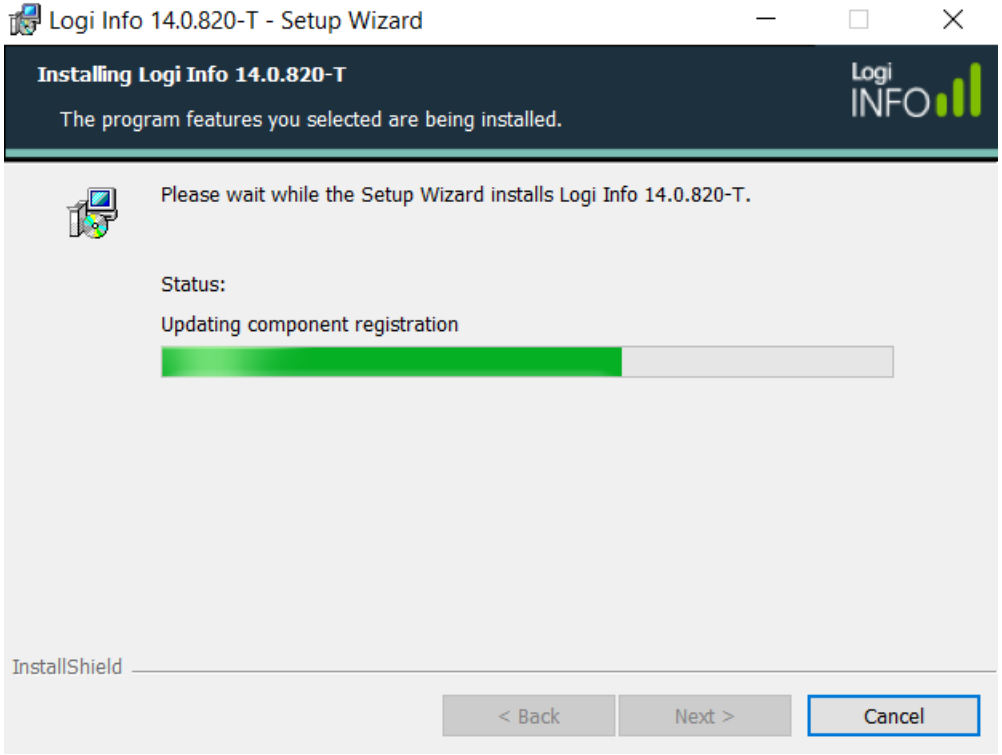
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



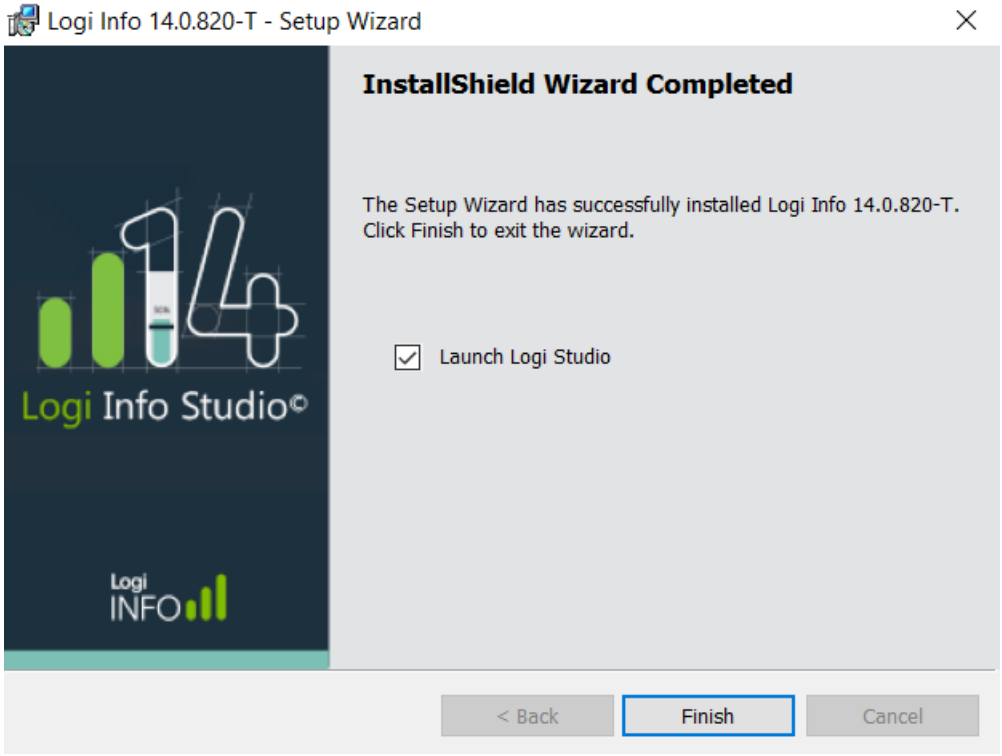
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).



9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

You can also launch Studio by using Start Menu → All Programs → Logi Info or Report → Studio.

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in Using Logi Studio.

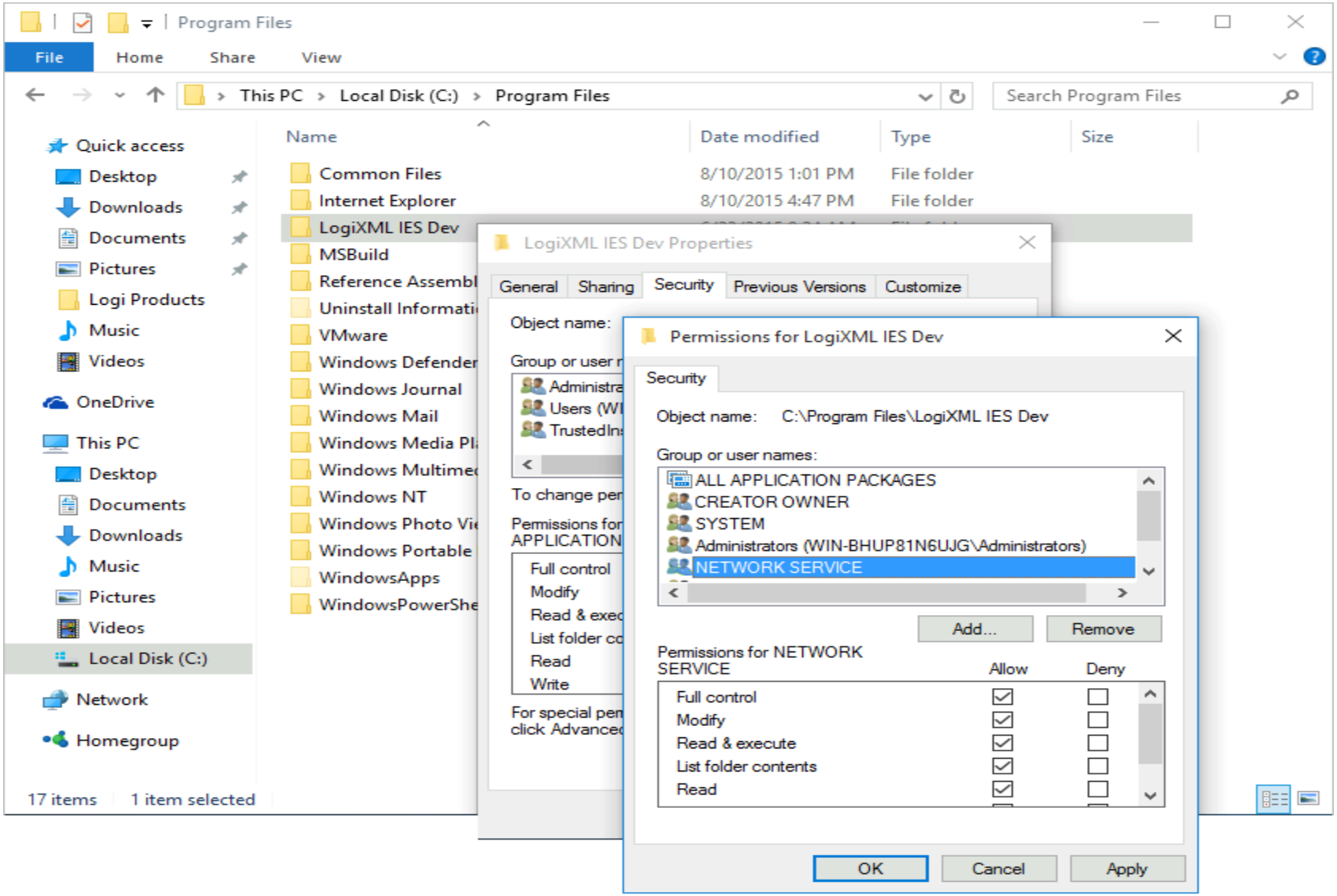
Installation is complete and you may begin to use Studio and/or the Server Engine immediately.

Install Logi Info on Windows 10 - Installing Add-on Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. *Add-on Modules* describes them and includes links to their installation topics.

Install Logi Info on Windows 10 - Setting File Access Permissions

You will need to set file access permissions to allow Logi Studio and IIS to work together smoothly:

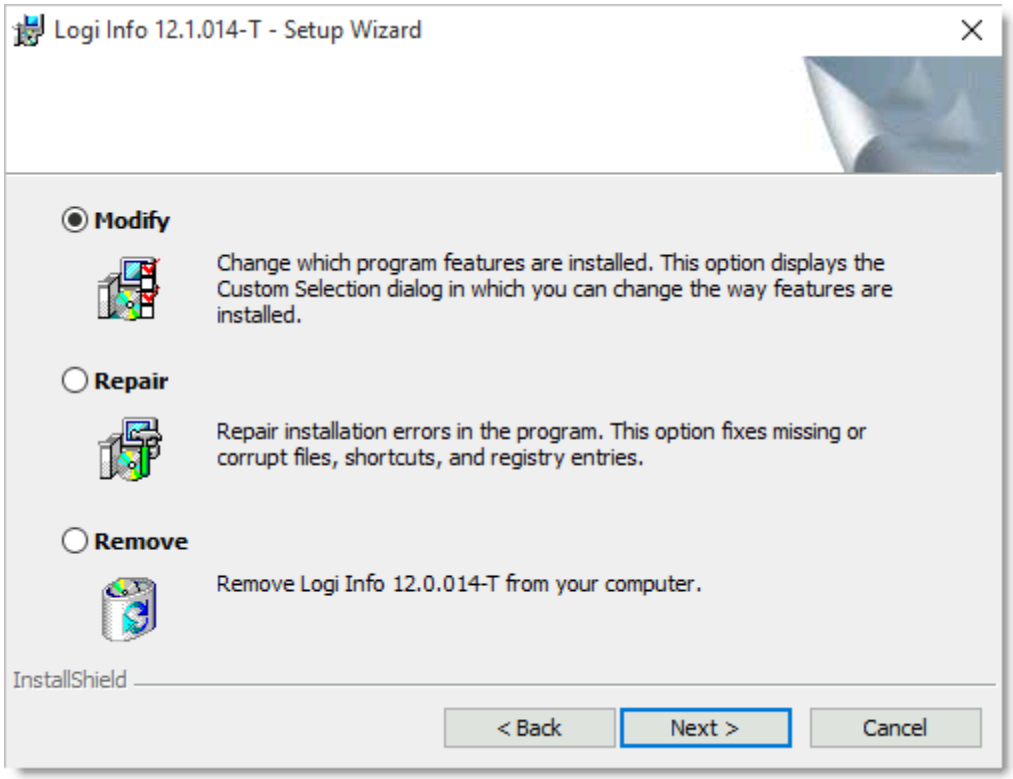


Set the permissions, as shown above, using these steps:

1. Use the File Explorer to navigate to `C:\Program Files\LogiXML IES Dev` and right-click the folder. Select *Properties*.
2. Select the Security tab and click Edit, the Add, and add the account NETWORK SERVICE.
3. Set the Permissions for NETWORK SERVICE to Full Control.
4. Click OK as necessary to exit

Install Logi Info on Windows 10 - Modifying or Repairing an Installation

Suppose you installed Logi Info but didn't initially install the Scheduler, and now you find you want to schedule reports. Or you suspect a Logi Studio .DLL file is missing or is corrupted and you want to fix it.



These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation program file (don't forget to right-click it and use "Run as administrator" to start it). *Do not* use Control Panel → Programs to do this; it will request an .msi file, which is not retained after the original installation.

Install Logi Info on Windows 7-8

The following topics guide you through installing Logi Info on a single computer under the Windows 7 or 8 operating system, for use with the IIS web server:

Install Logi Info on Windows 7-8 - Installation Scenarios

There are two major parts to the Logi product in the installation file you downloaded:

Logi Studio, our development environment, is a Windows application that's typically installed on a Windows development machine and interacts with the local IIS web server for Logi application development and testing.

The **LogiServer Engine** is a set of files that's part of each Logi application and which provides an extension to the IIS web server at runtime. When you build a Logi application, Studio adds the Engine files to the application. The engine also includes a Windows utility application, **Logi Server Manager**, that allows you to perform basic configuration of Logi applications without using Studio.

A very typical installation scenario is to install Studio and the Server Engine on a development machine, and then install only the Server Engine (including Server Manager) on the production web server. This "development and production" dual installation is allowed by our licensing scheme.

Additional copies of Studio may be licensed and installed on additional development machines.

There are a number of ways to *deploy* your Logi applications to production, including using Studio's built-in **Application Deployment** tool.

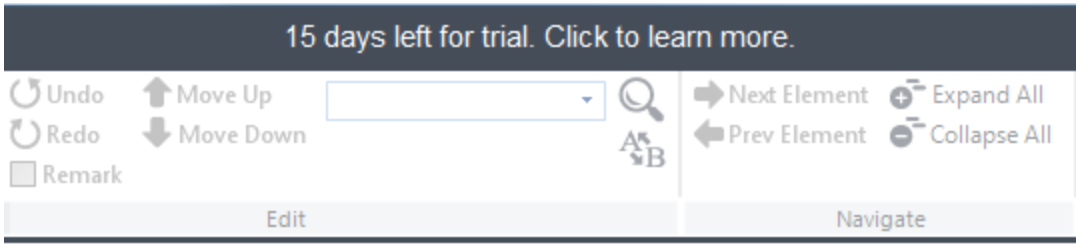
Other installation scenarios, involving shared network drives and team development, can be used. This topic covers installation of Studio and the Server Engine on a development machine, and also explains what's needed to install the Server Engine on a single production server.

32- or 64-bit?

Logi Info versions earlier than v12.2 are available for **32-bit** and **64-bit** systems. Starting with v12.2 only 64-bit versions are available. There are different Logi product distribution files for 32-bit and 64-bit versions, so be sure you've downloaded the right one for your system.

Install Logi Info on Windows 7-8 - Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You don't need to do anything but install the product and you can begin using it immediately. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about **purchasing** a Logi Info license.

After the trial period expires, Studio won't be usable and any Logi reports you may have developed will *no longer run* without a regular license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server. Our licensing scheme allows you to deploy our product on one development machine and on one production server. As mentioned earlier, additional separate licenses for Studio, for additional developers, are also available.

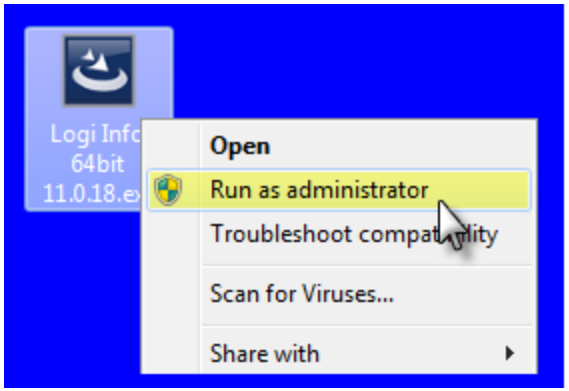
Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Install Logi Info on Windows 7-8 - Preparing to Install

Logi products work with *all* editions of Windows 7. The following sections discuss features that impact the installation and/or operation of Logi Analytics products.

Use the Administrator Account

Windows includes a number of security enhancements that impact software installation and it is critical that installation and configuration occur while using the software with the built-in "Administrator" account.



Even if your personal account has been added to the local Administrators Group, it may not have sufficient privileges, so don't rely on it.

As shown above, the correct practice when running the Logi installation program or using the Command Line to make configuration adjustments is to start the tool by *right-clicking* its icon and selecting "**Run as administrator**" from the menu to start the program. The ensures that appropriate permissions are provided for the installed components.

Don't see a "Run as administrator" option? If the system is in a network domain, your network admin may have created security policies that don't allow you to see this option, in which case you need to consult your IT staff for assistance.

.NET Framework

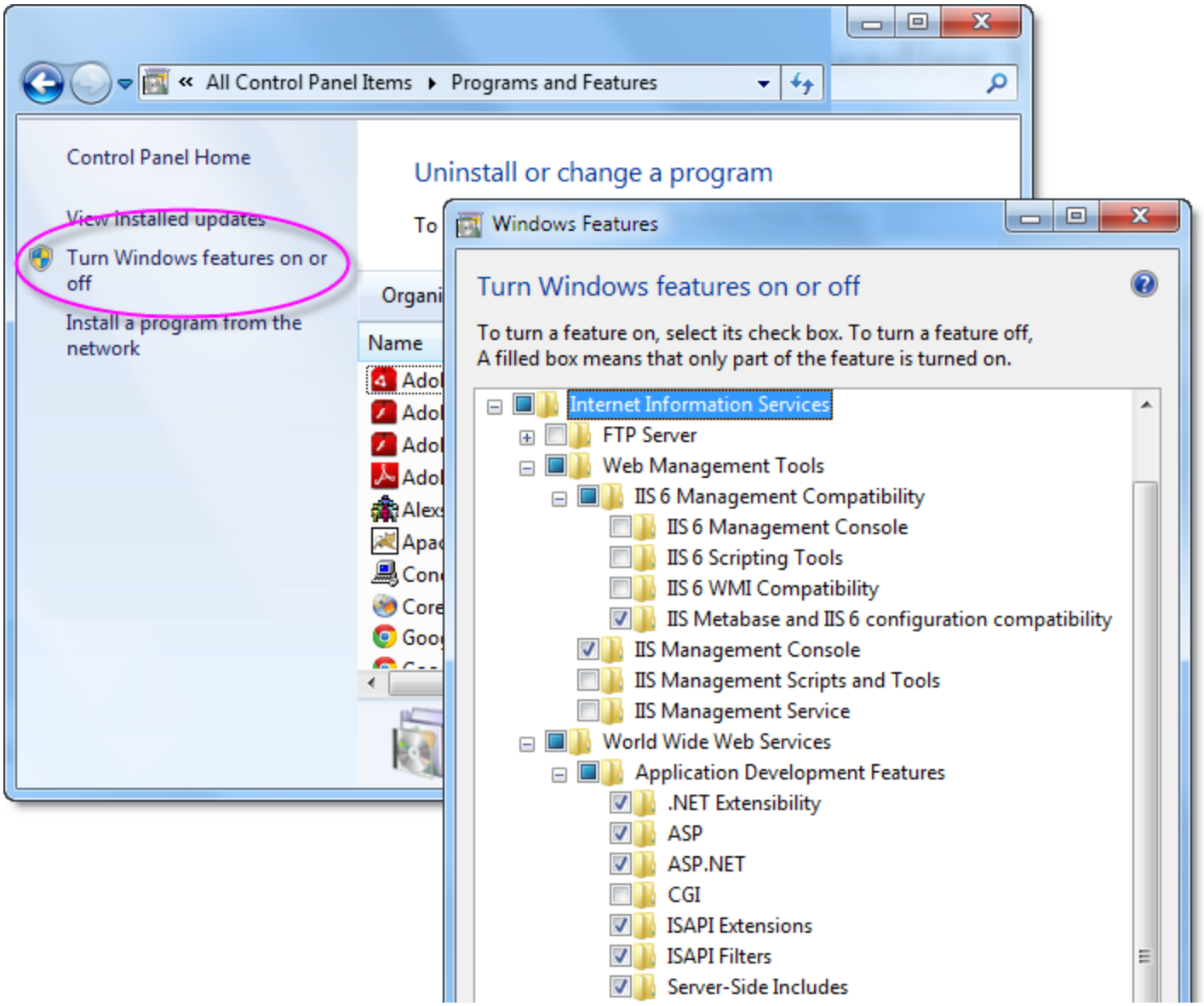
Logi products require the .NET Framework components. When Windows is installed, multiple versions of the framework are typically included. Before doing anything else, ensure that you have .NET installed by using File Explorer to browse to:

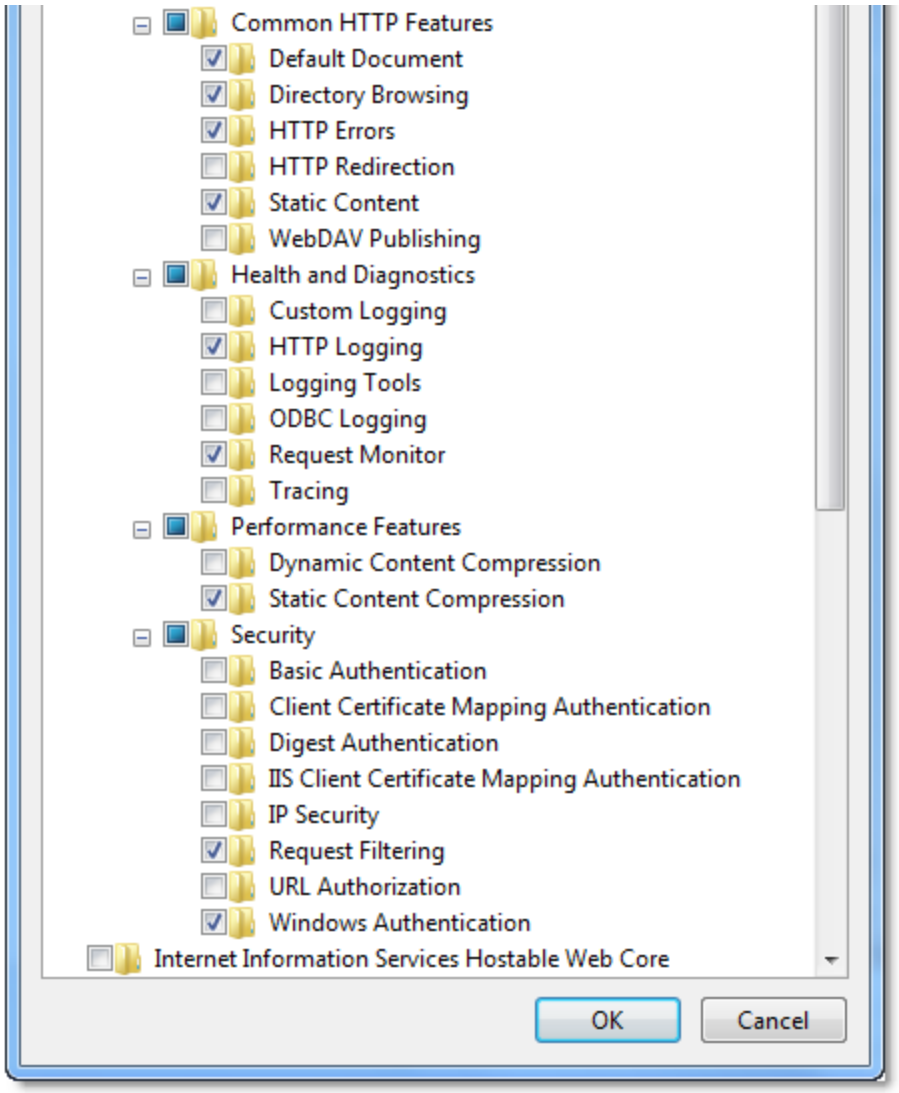
```
C:\Windows\Microsoft.NET\Framework
```

where you should see several folders, such as v4.0.30319, one for each version of .NET installed. The .NET Framework 4.x is required. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Enable the IIS Web Server

When Windows is installed, the components for the **IIS web server** are installed but are usually not enabled. You must do this manually, *before* you install your Logi product.


« All Control Panel Items ▸ Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Uninstall or change a program
Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off,
A filled box means that only part of the feature is turned on.
Internet Information Services
FTP Server
Web Management Tools
IIS 6 Management Compatibility
IIS 6 Management Console
IIS 6 Scripting Tools
IIS 6 WMI Compatibility
IIS Metabase and IIS 6 configuration compatibility
IIS Management Console
IIS Management Scripts and Tools
IIS Management Service
World Wide Web Services
Application Development Features
.NET Extensibility
ASP
ASP.NET
CGI
ISAPI Extensions
ISAPI Filters
Server-Side Includes



This is accomplished in Control Panel, under Programs and Features → Turn Windows features on or off, as shown above in Windows 7. Check the boxes on your machine so they match the image above, then close the dialog boxes.

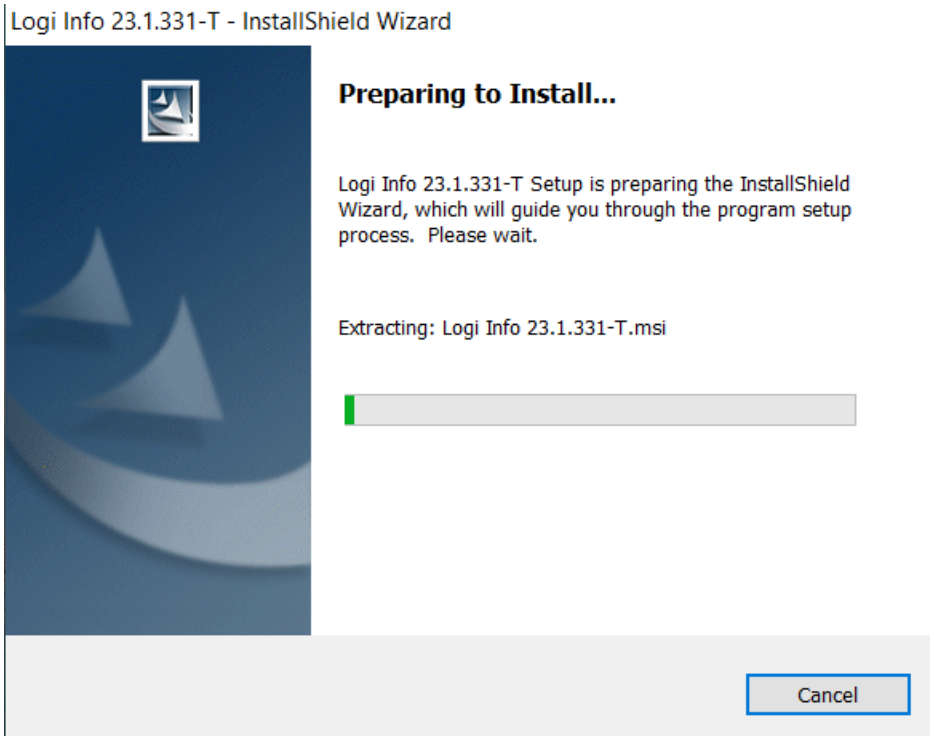
The IIS "Default Web Site"

When it's enabled, IIS creates a "Default Web Site" and the New Application wizard in Logi Studio expects to create all new Logi applications as virtual directories of that web site. If you have renamed, replaced, or disabled the "Default Web Site", or have installed another web server that handles HTTP requests on Port 80, the wizard will fail during its application registration phase. Under these circumstances, you can continue to use Logi Studio to develop Logi applications but you will need to manually register them. This process is described in *Windows IISConfiguration*.

Install Logi Info on Windows 7-8 - Installing the Software

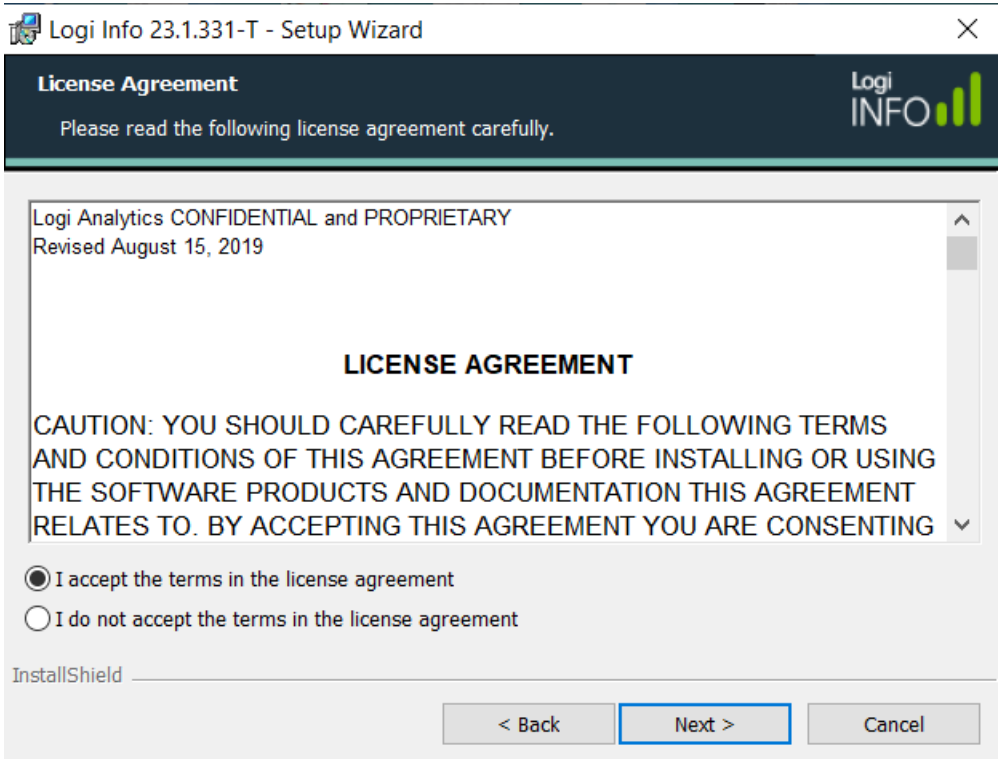
This topic describes the installation steps and the dialog boxes you'll see when using the interactive tool built into your Logi product installation file. If you need a "silent" (non-interactive) installation, see "Command Line Install" on page 327.

1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.

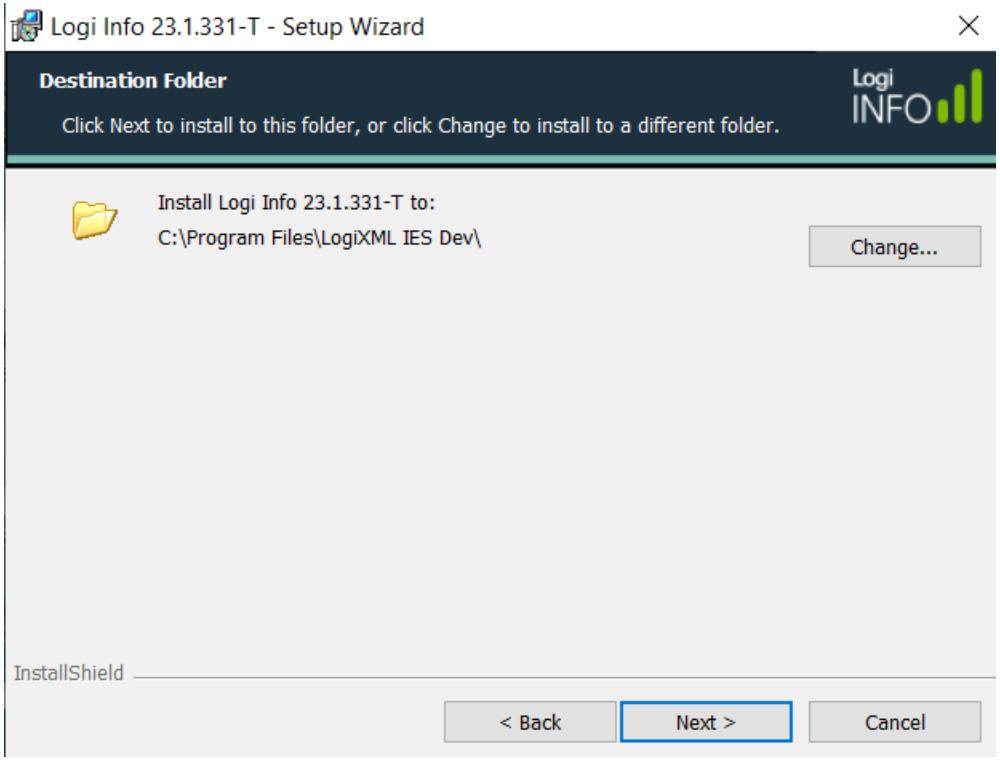


2. When the **Welcome Screen** appears, select **Next:**

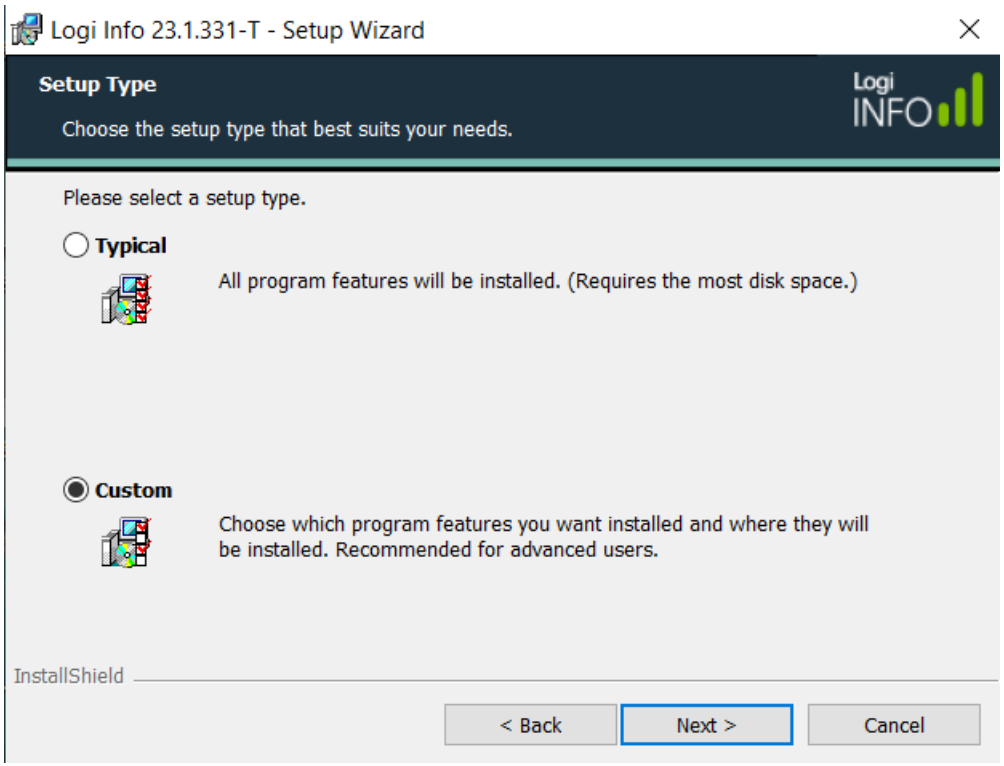
3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and select **Next** to continue:



4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



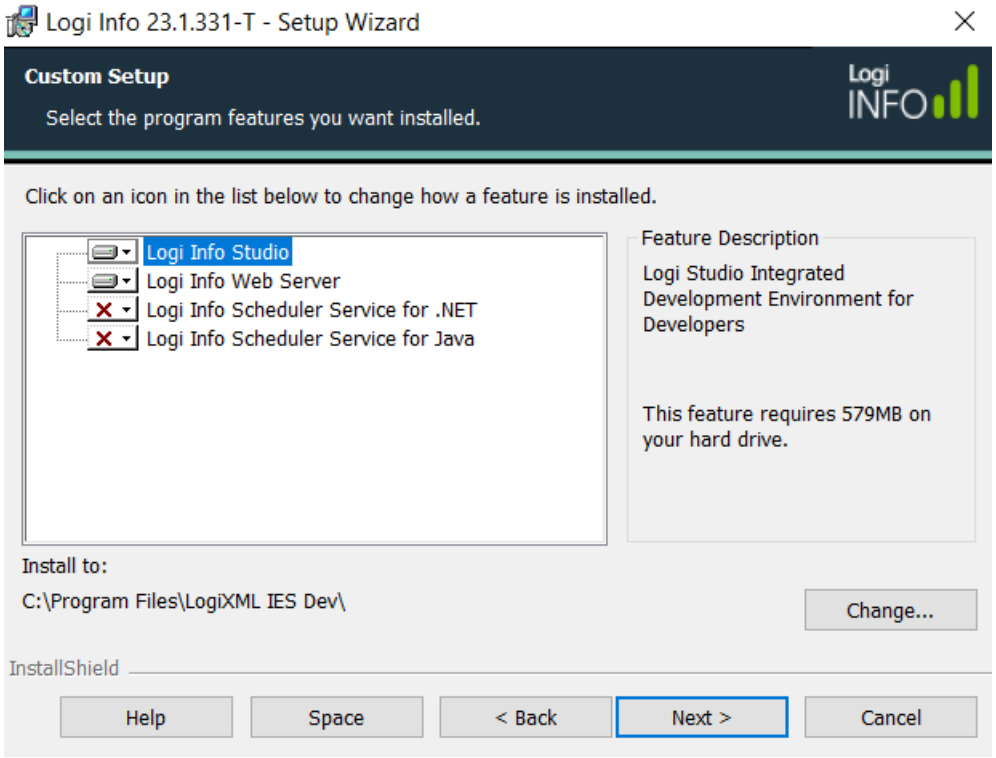
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

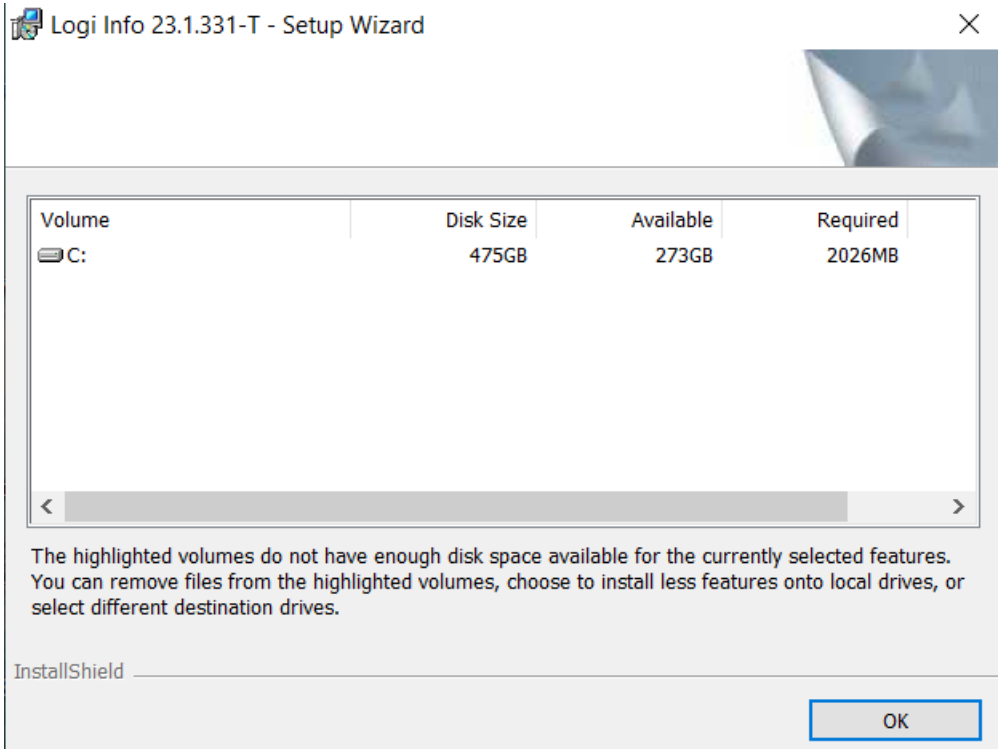
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

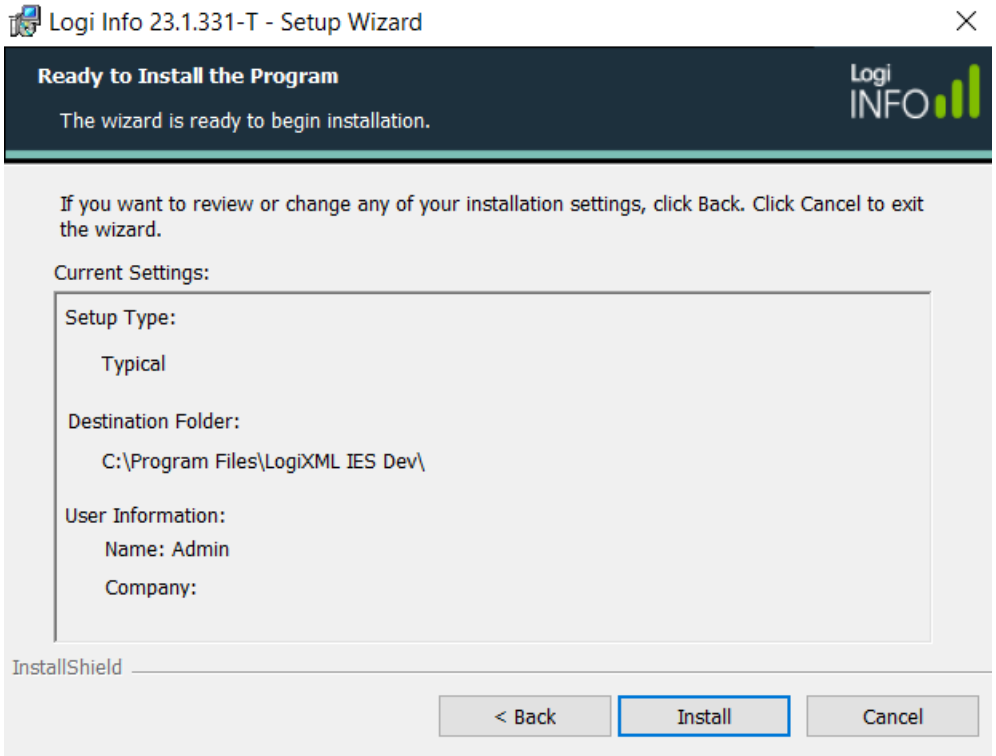
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



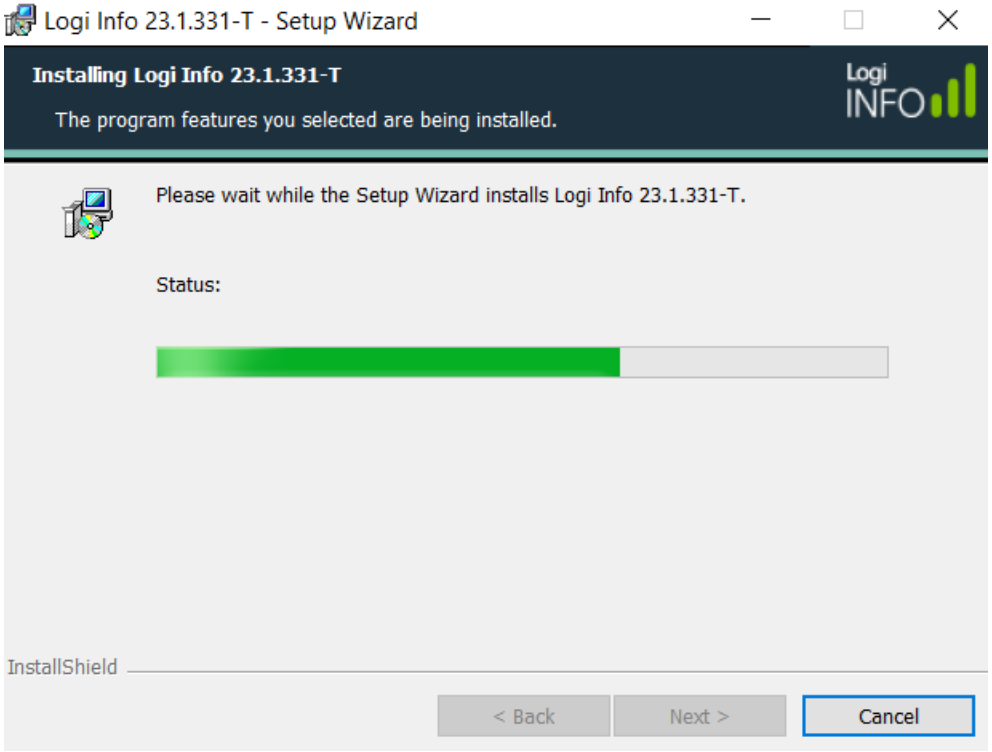
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



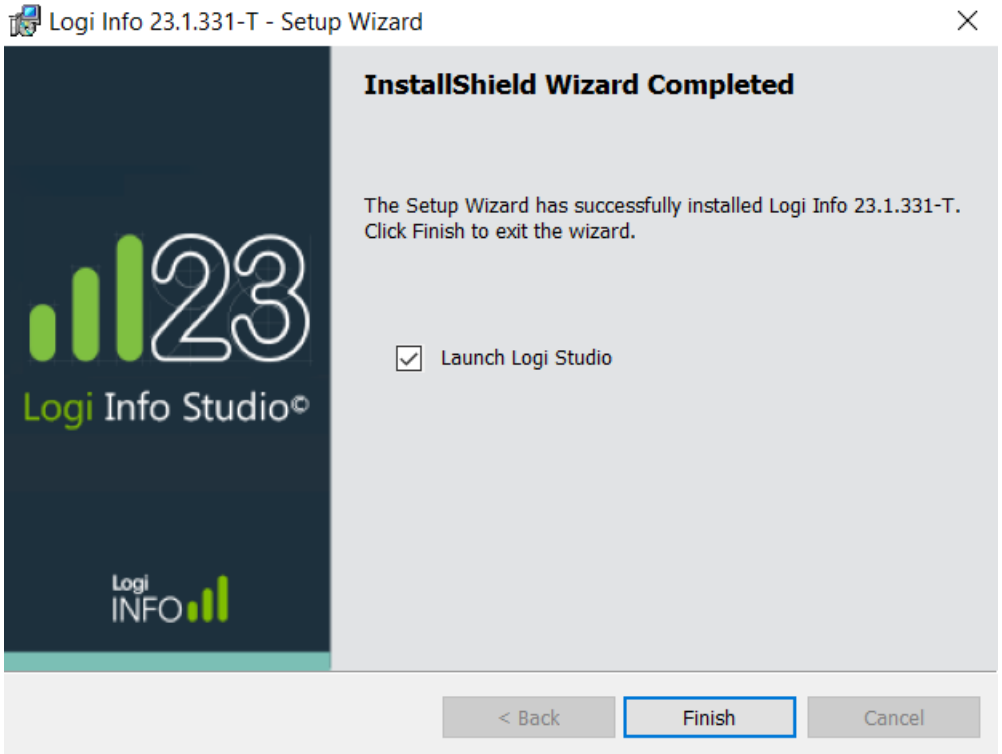
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).

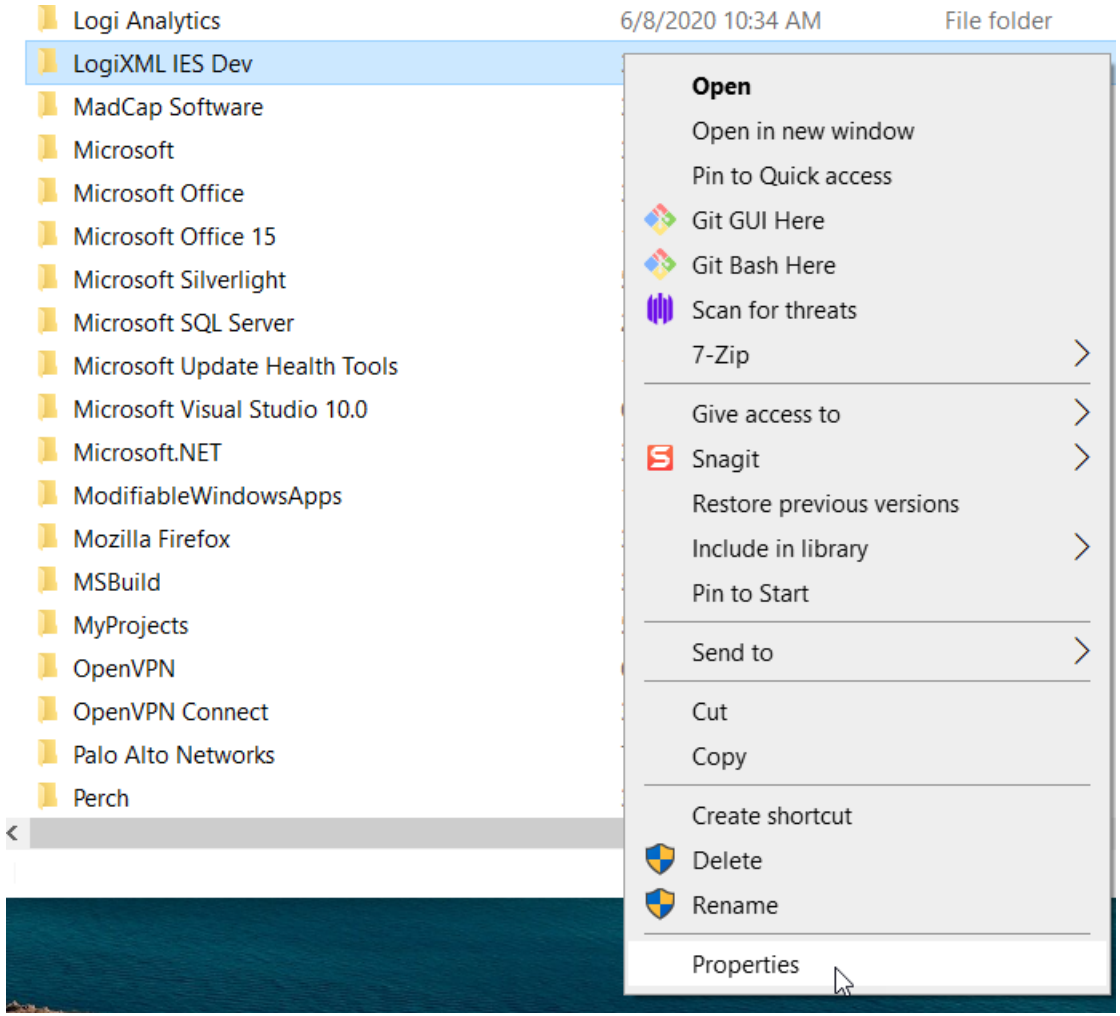


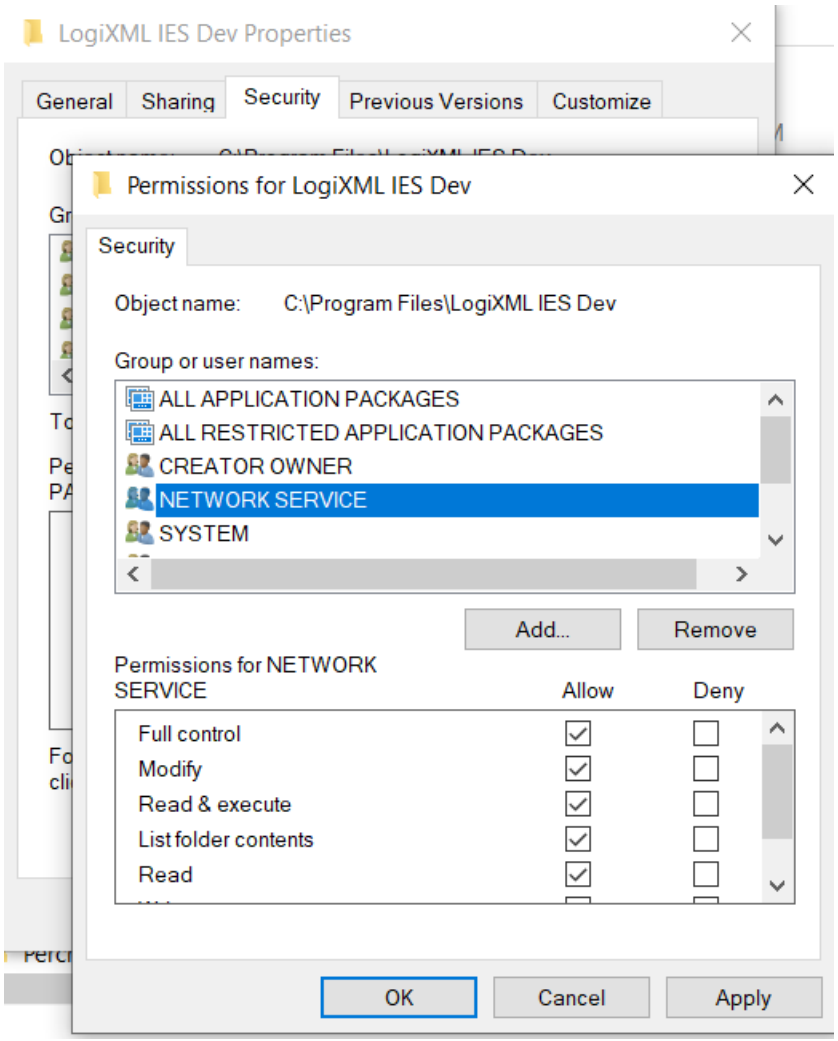
9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

Installation is complete and you may begin to use Studio and/or the Server Engine immediately. *If* you have User Account Control (UAC) *turned off*, you may click **Finish** to exit the installer and launch Studio and skip the next two steps. If UAC is *not* turned off, uncheck the "Launch Logi Studio" check box, then click **Finish** and proceed to the next two steps.

If you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box; just click **Finish**.

10. Use the Computer browser to navigate to `C:\Program Files`. Right-click the `LogiXML IES Dev` folder and select Properties. In the Security tab, add **NETWORK SERVICE** to the list of user names and grant it **Full Control**. Repeat the process, granting Full Control to **your own account** (and to any other Logi developers who will work on this computer).


Logi Analytics
6/8/2020 10:34 AM
File folder
LogiXML IES Dev
MadCap Software
Microsoft
Microsoft Office
Microsoft Office 15
Microsoft Silverlight
Microsoft SQL Server
Microsoft Update Health Tools
Microsoft Visual Studio 10.0
Microsoft.NET
ModifiableWindowsApps
Mozilla Firefox
MSBuild
MyProjects
OpenVPN
OpenVPN Connect
Palo Alto Networks
Perch
Open
Open in new window
Pin to Quick access
Git GUI Here
Git Bash Here
Scan for threats
7-Zip
Give access to
Snagit
Restore previous versions
Include in library
Pin to Start
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties



11. You may now launch your Logi product and you should see a splash screen similar to the one shown below:

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in *Using Logi Studio*.

Installation is complete and Studio and/or the Server Engine is ready for use. See "Install Logi Info on Windows 7-8 - Configuring IIS" on page 185 to complete.

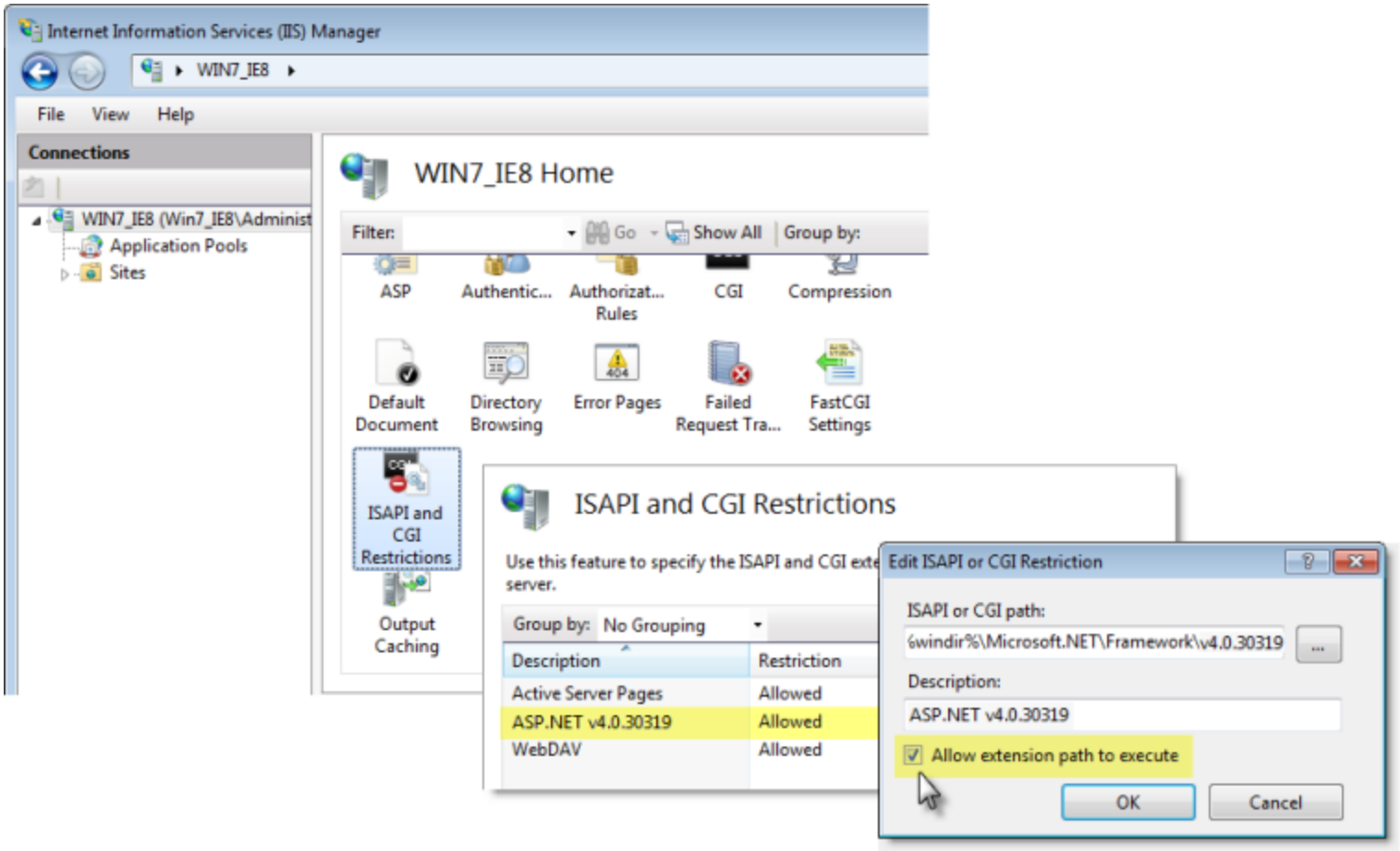
Install Logi Info on Windows 7-8 - Installing Add-on Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. *Add-on Modules* describes them and includes links to their installation topics.

Install Logi Info on Windows 7-8 - Configuring IIS

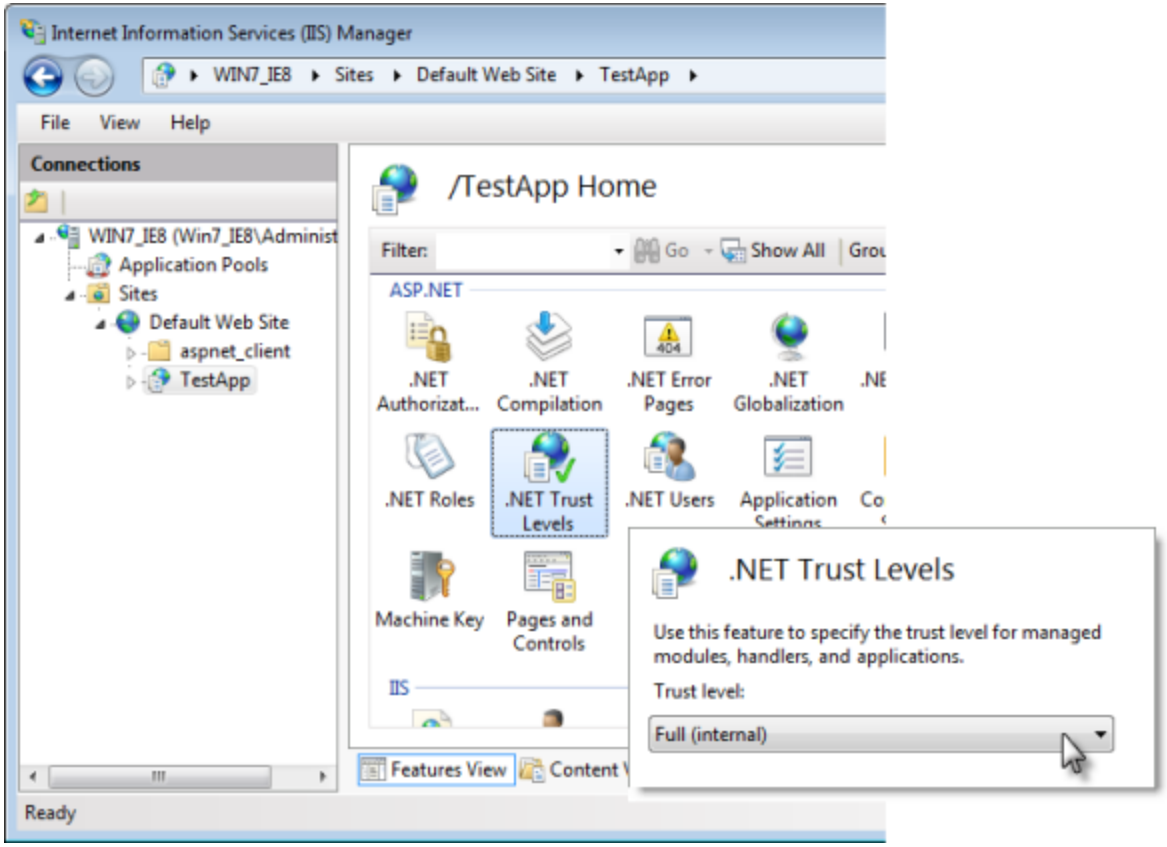
This topic demonstrates how to configure IIS to install Logi Info on Windows 7-8.

First, using the IIS Manager tool provided with the OS, ensure that the ASP.NET extension is allowed to execute:



This is done, as shown above, by selecting the Home item at the left, then double-clicking the **ISAPI and CGI Restrictions** item. If the ASP.NET entry (or entries) is marked *Not Allowed*, edit its settings and check the *Allow extension*... check box.

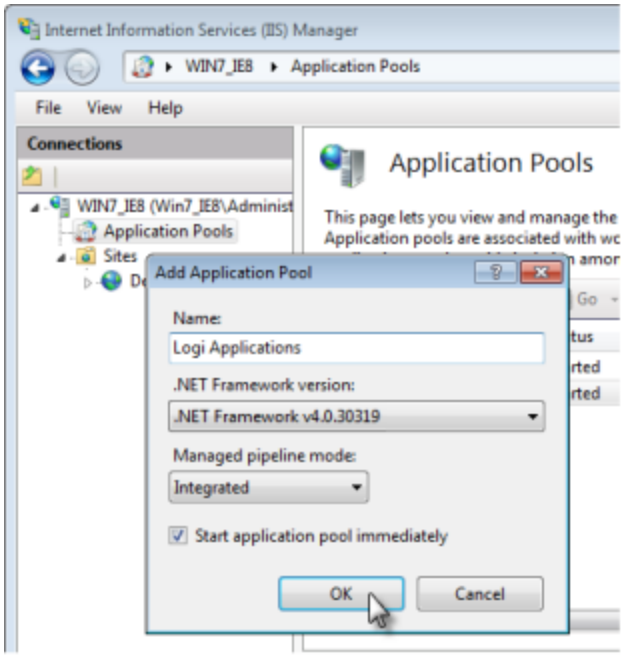
Next, you must ensure that either your Default Web Site, or your individual Logi application virtual directories (when you create them), have the correct **.NET Trust Levels** assigned:

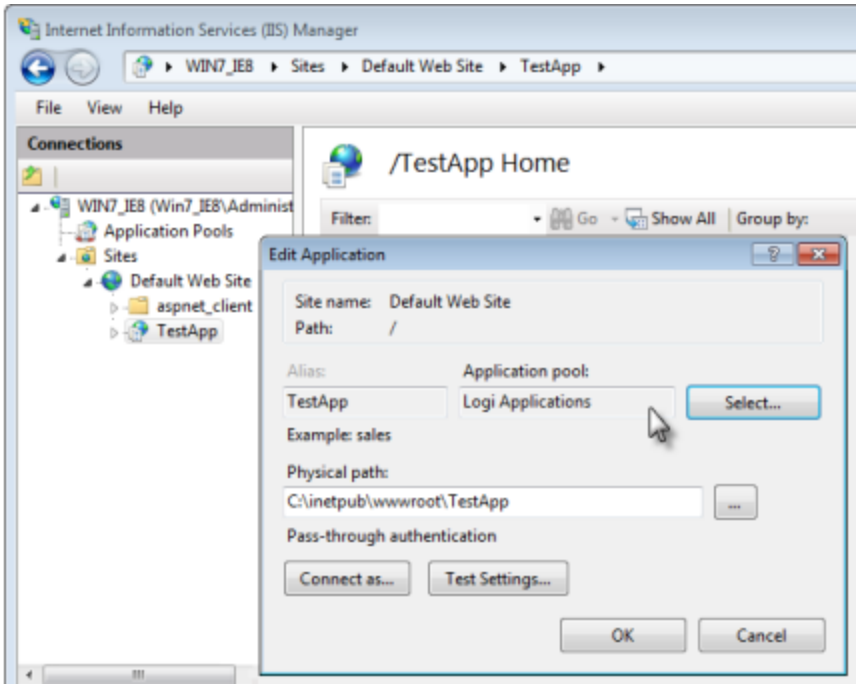


In order to be able to cache data, interact with handlers, and perform other required operations, the .NET Trust Levels must be

set to *Full (Internal)*, as shown above.

Finally, here's a "best practices" tip for configuring IIS for use with Logi applications:

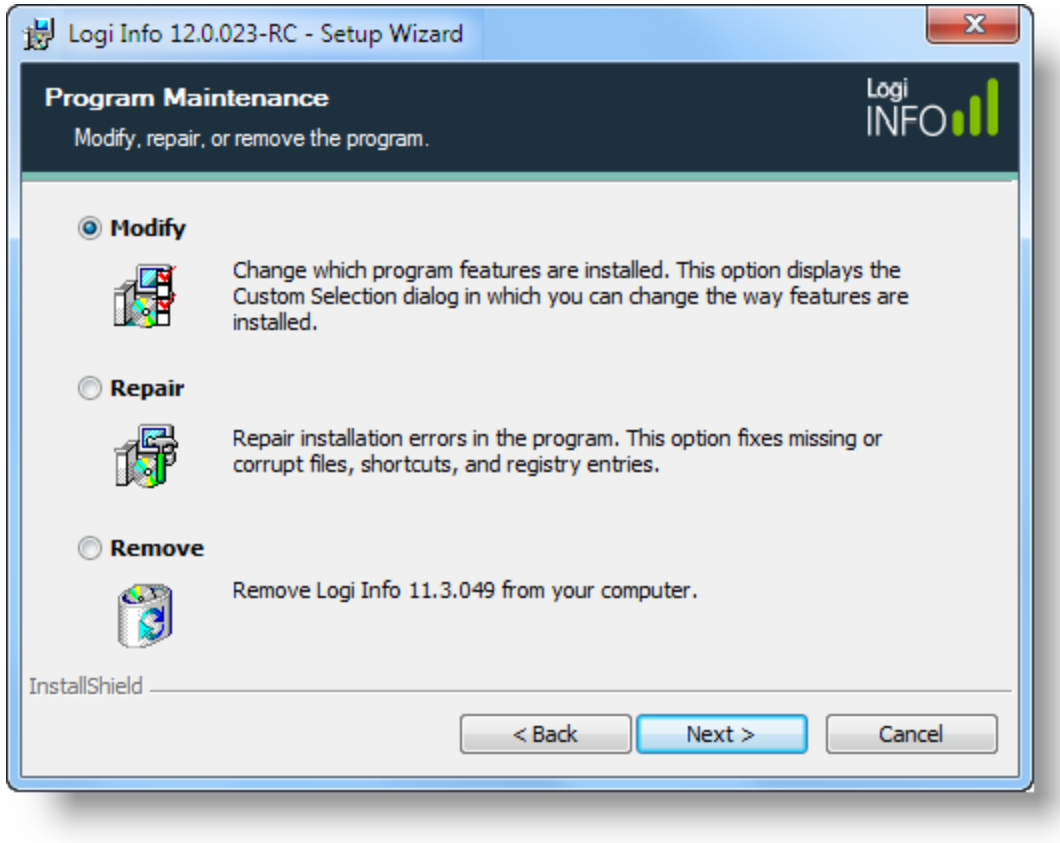




We recommend that you create a new **Application Pool** specifically for your Logi applications, as shown above, left. Then you will need to assign your Logi app virtual directories to it by editing their Basic Settings and selecting the new pool, as shown above, right. This allows you to manage and restart the pool independently of any other non-Logi applications that are also using IIS. Logi apps use the default *Integrated pipeline* mode in application pools, resulting in better performance; Logi apps built using earlier versions must use *Classic* mode.

Install Logi Info on Windows 7-8 - Modifying or Repairing the Installation

Suppose you installed Logi Info but didn't initially install the Scheduler, and now you find you want to schedule reports. Or you suspect a Logi Studio .DLL file is missing or is corrupted and you want to fix it.



These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation program file (don't forget to right-click it and use "Run as administrator" to start it).

Do not use Control Panel → Programs to do this; it will request an .msi file, which is not retained after the original installation.

Install the Self-Service Reporting Module

The Logi Self-Service Reporting Module (SSRM) is an add-on module that enables special elements in Logi Info and adds a complete, pre-built Logi application to your computer.

The following topics guide you through installing the Self-Service Reporting Module on a single computer under the Windows operating system:

- Preparing to Install
- Installing the Software
- Removing or Repairing the Installation

We have a separate topic, "Upgrade the Self-Service Reporting Module" on page 198, that provides guidance for installing an upgrade. Information about which Logi Info and Self-Service Reporting Module versions we recommend using together is available in Release Pairings. For more information, see the Release Notes for each Logi Info version.

About the Self-Service Reporting Module

The special SSRM elements installed can be used with the Analysis Grid in your applications and make it easy for users to configure data queries at runtime, as described in *The Analysis Grid for Developers*.

The included application, **InfoGo**, uses the special SSRM elements and allows users to create and share visualizations, Dashboards, and reports at runtime. We've provided a guide for end-users, *Use InfoGo*. Developers can examine InfoGo for educational purposes and can also customize its branding and other features, if desired. InfoGo can be deployed to .NET and Java servers.

After the SSRM installer finishes, there are several things you need to do to complete the installation. These are explained in *Configure InfoGo for Developers*.

Important Restrictions

Please note these important restrictions on the use of the SSRM:

- Requires Logi Info v12 or later, 64-bit version (there is no 32-bit SSRM version available).
- Works with Chrome, Firefox, and other major browsers, but *does not* work with Internet Explorer 7 or 8.

The **Active Query Builder** and **Metadata** elements and InfoGo application installed with the SSRM only work with these database sources:

- 1010data
- DB2
- Microsoft SQL Server 2005+
- MySQL
- Oracle
- PostgreSQL
- Redshift (Amazon)
- Vertica (HP)
- ODBC- and JDBC-accessible databases that use the same SQL syntax as one of the databases listed above

Ensure that your environment meets these restrictions prior to installation!

Install the Self-Service Reporting Module - Preparing to Install

Logi Info v23.1 must be installed and licensed *before* the SSRM is installed. The SSRM works with your Logi Info license.

 It's critical that installation and configuration occur while running as the built-in "Administrator" account. Start the tool by *right-clicking* its icon and selecting "**Run as administrator**" from the menu to start the program. This ensures that appropriate permissions are provided for the installed components.

Even if your personal account has been added to the local Administrators Group, it may not have sufficient privileges, so don't rely on it.

Don't see a "Run as administrator" option?

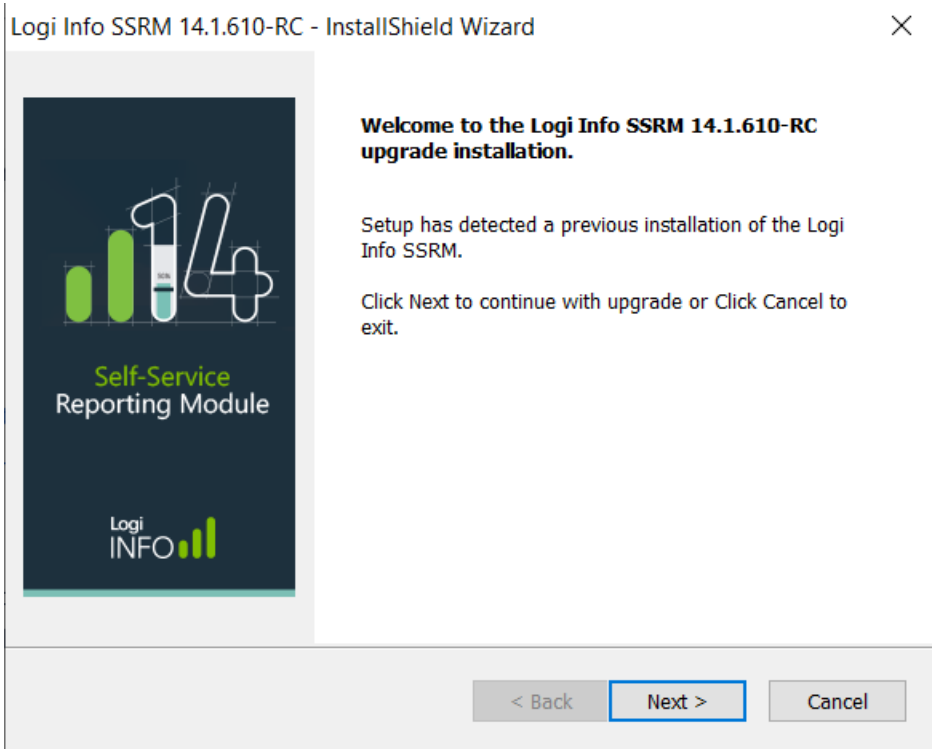
If the system is in a network domain, your network admin may have created security policies that don't allow you to see this option, in which case you need to consult your IT staff for assistance.

Install the Self-Service Reporting Module - Installing the Software

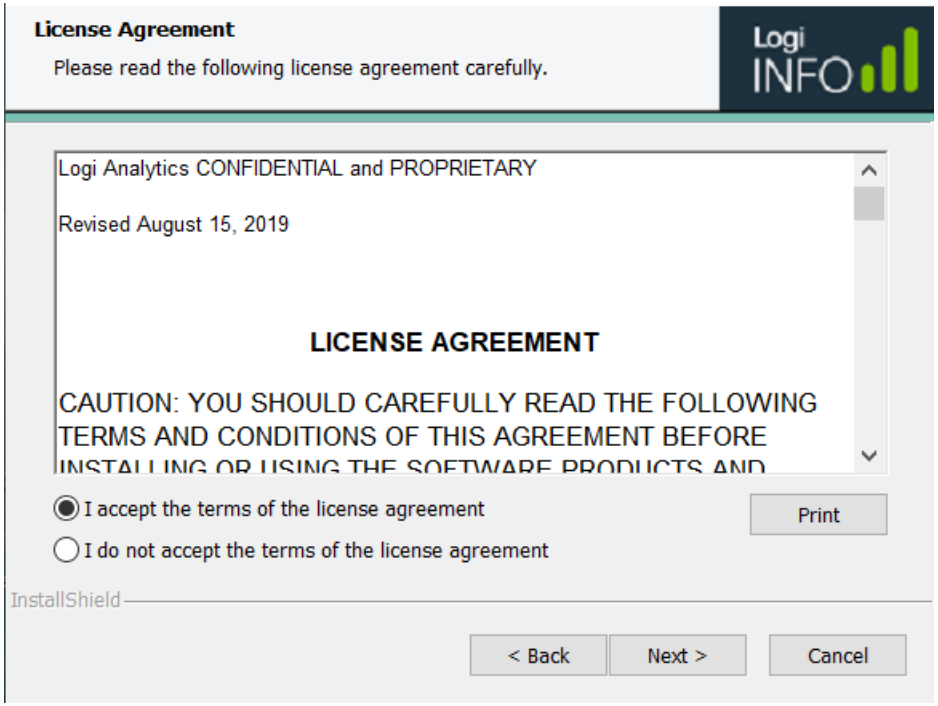
This topic demonstrates how to install Logi Info SSRM 14.1.

As usual, you can click **Back** at any time before the physical installation begins to go back to the previous screen.

1. To start the installation, *right-click* the Logi product installation program icon and select "Run as administrator" to launch the installer. Allow it to complete the installation preparation.
2. When the **Welcome Screen** appears, click **Next**.

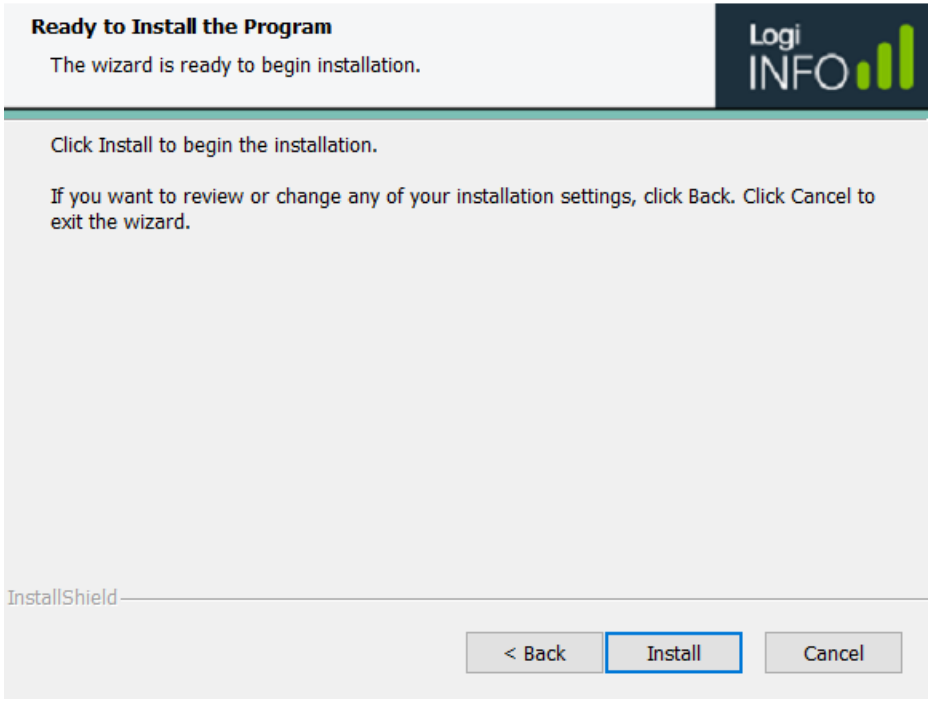


3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and click **Next** to continue.

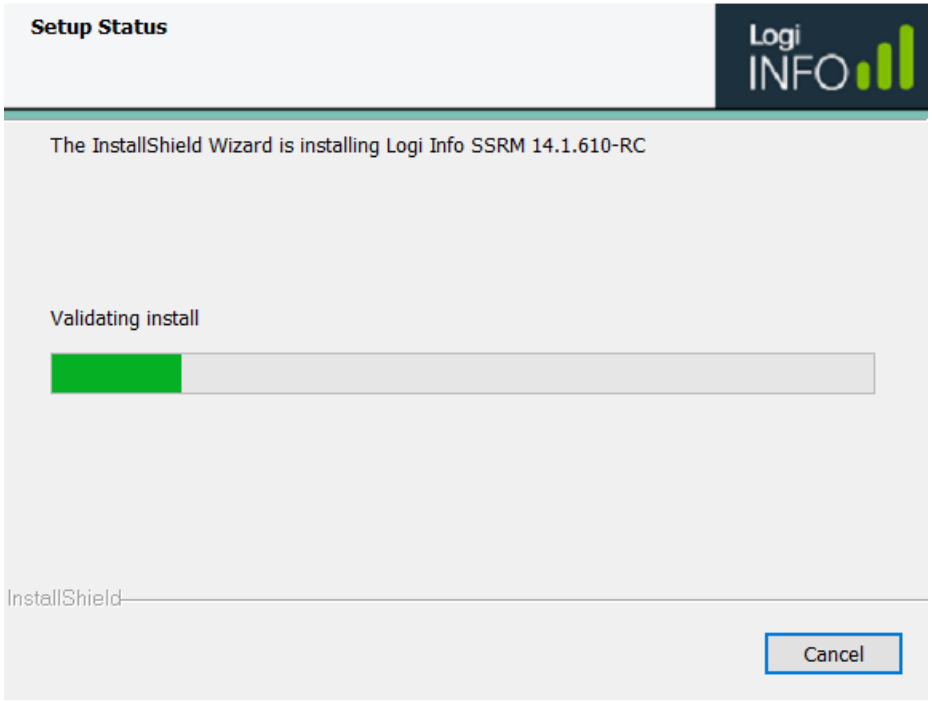


As usual, you can select **Back** at any time before the physical installation begins to go back to the previous screen

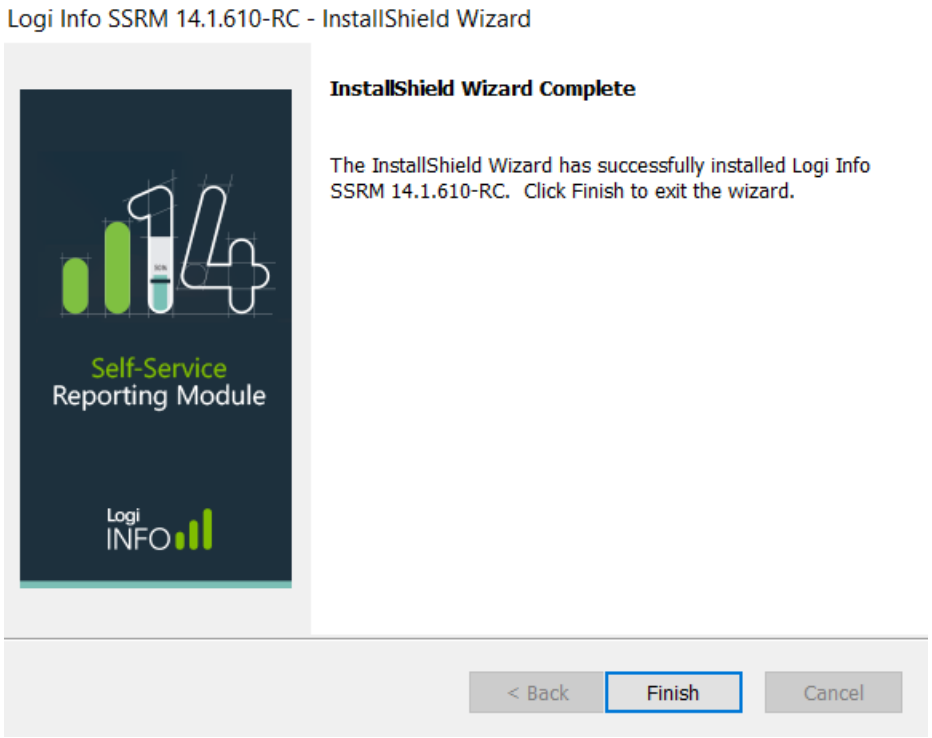
4. **Ready to Install**: Click **Install**.



5. The physical installation will begin and you'll see several **progress indicators** for different tasks.



6. **Installation is complete**: Click **Finish** to exit the installer.

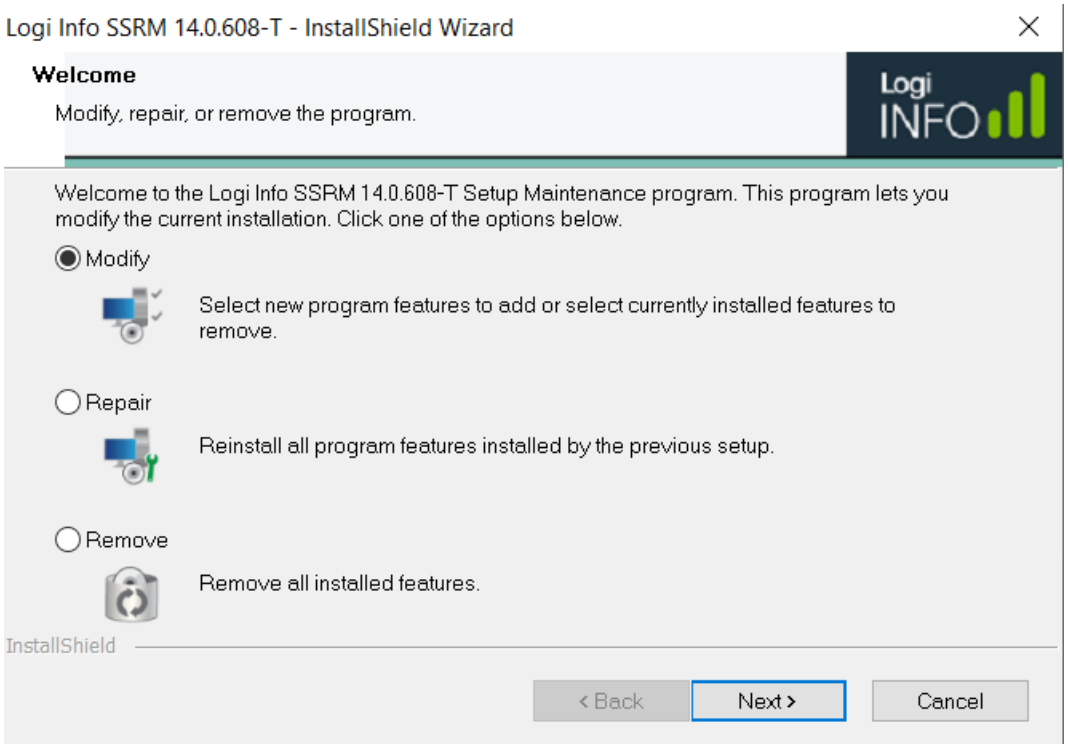


What To Do Next

Before users can use the included InfoGo application, there are several steps you still need to take. For example, you need to register and license it and make a number of configurations in Studio. These steps are explained in *Configure InfoGo for Developers*. If you want to use the Discovery Module 3.0 or Discovery Module 3.1 add-on with the SSRM, you'll need to install it separately and configure the SSRM to enable it. For more information on installing the Discovery Module 3.1, see "Install the Discovery Module - Windows" on page 224.

Install the Self-Service Reporting Module - Removing or Repairing the Installation

Should you ever need to, you can repair damaged SSRM files or remove the SSRM completely by re-running the installer:



Don't forget to right-click it and use "Run as administrator" to start it. You can also use Control Panel → Programsto do this.

Removing the SSRM will remove the special elements *and* the InfoGo application (any InfoGo definition or support files you've modified, however, will be left in place).

Upgrade the Self-Service Reporting Module

The following topics guide you through upgrading the **Self-Service Reporting Module** (SSRM):

Upgrade Considerations

SSRM upgrades are usually associated with a Logi Info upgrade and *we highly recommend that you upgrade both of them together*. Upgrading just one or the other will usually result in undesirable effects. See *Release Pairings* for more information.

If you're upgrading to an early version (pre-v12.5), see "Upgrading to Earlier Versions" on page 218 for important details.

Upgrading to v12.5+

If you're upgrading to the **SSRM v12.5+**, be aware of these important considerations related to the included InfoGo application:

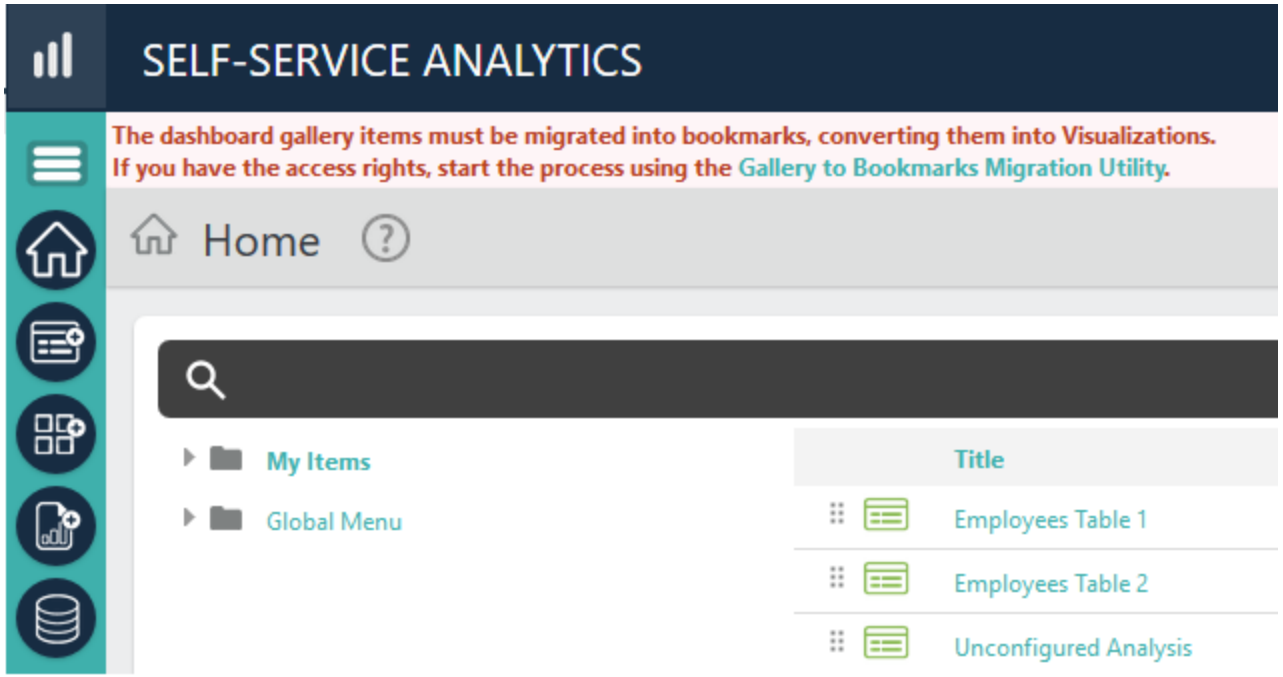
- Upgrading the SSRM will *overwrite* any edited InfoGo files, *except* those in the `goCustomizations`virtual folders under InfoGo → Definitions → Reports and InfoGo → _SupportFiles. The file `goCustomizations.goWebFooter.lgx`will not be overwritten, for example.
- The application's `_Settings.lgx`definition will not be overwritten; however, the new version may require new elements and/or attributes not found in older versions. You'll have to add these manually in the source code if they're missing. See *Configure InfoGo for Developers*for details.
- If you've used your own Definition or Template Modifier Files, identifiers used with them may change from one SSRM version to the next and you may need to adjust your modifier files accordingly.

Making backup copies of any InfoGo files you've edited, prior to applying an upgrade, is always highly recommended.

Upgrading to v12.7

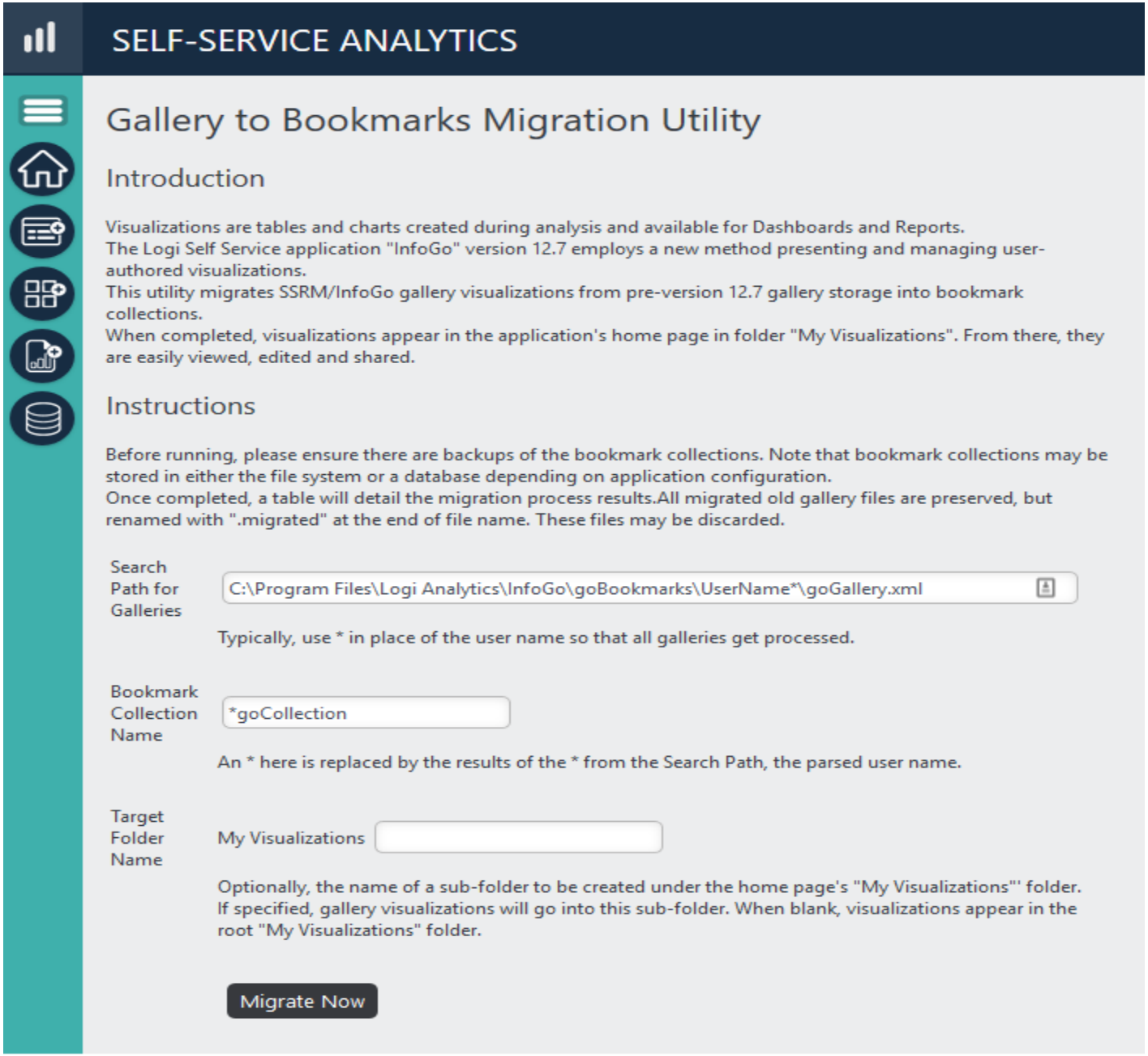
InfoGo v12.7 provides enhanced content management. Visualizations are stored as Bookmarks, which lets them be presented and manipulated in a way consistent with other objects. This makes it easy to edit them, rename them, and duplicate them, and eliminates the need for Gallery and Extra Gallery files.

SSRM 12.7 provides an upgrade tool, the **Gallery to Bookmarks Migration Utility**, for migrating visualizations from gallery files to Bookmarks. When you launch InfoGo for the first time after upgrading to it, you'll see a prompt to run the tool:



You can also run the utility independently by browsing: `/rdPage.aspx?rdReport=rdTemplate/rdGalleryMigration`

You must have the "rdBookmarksAdmin" security right to be able to run the utility. Click the link to run it:


SELF-SERVICE ANALYTICS
Gallery to Bookmarks Migration Utility
Introduction
Visualizations are tables and charts created during analysis and available for Dashboards and Reports.
The Logi Self Service application "InfoGo" version 12.7 employs a new method presenting and managing user-authored visualizations.
This utility migrates SSRM/InfoGo gallery visualizations from pre-version 12.7 gallery storage into bookmark collections.
When completed, visualizations appear in the application's home page in folder "My Visualizations". From there, they are easily viewed, edited and shared.
Instructions
Before running, please ensure there are backups of the bookmark collections. Note that bookmark collections may be stored in either the file system or a database depending on application configuration.
Once completed, a table will detail the migration process results.All migrated old gallery files are preserved, but renamed with ".migrated" at the end of file name. These files may be discarded.
Search Path for Galleries
C:\Program Files\Logi Analytics\InfoGo\goBookmarks\UserName*\goGallery.xml
Typically, use * in place of the user name so that all galleries get processed.
Bookmark Collection Name
*goCollection
An * here is replaced by the results of the * from the Search Path, the parsed user name.
Target Folder Name
My Visualizations
Optionally, the name of a sub-folder to be created under the home page's "My Visualizations"' folder. If specified, gallery visualizations will go into this sub-folder. When blank, visualizations appear in the root "My Visualizations" folder.
Migrate Now

The Migration Utility page is shown above and is fairly self-explanatory, but here are a few key points:

- Make file or database backups *before* you run the utility!
- Gallery files that have been processed are retained in place, with .migrated added to their file names, and can be deleted after ensuring everything migrated properly.
- An asterisk (*) wildcard is used in the input fields to represent the username parts of folder and file names.
- The migrated visualizations will appear in a new folder on the InfoGo Home page, called "My Visualizations".
- You have the option to output the migrated visualizations into a sub-folder of "My Visualizations".

After the migration completes, you'll see the results table appear at the bottom of the page:

Migrated File Name	Target Bookmark Collection	Target folder	Parsed User Name	Visualizations Migrated	Result
C:\Program Files\Logi Analytics\InfoGo \goBookmarks \UserNameLindsay \goGallery.xml	goCollection	My Visualizations	Lindsay	42	✓
C:\Program Files\Logi Analytics\InfoGo \goBookmarks \UserNameJiang \goGallery.xml	JianggoCollection	My Visualizations	Jiang	0	✗

As shown in the example above, the results will indicate whether or not the visualizations were successfully migrated.

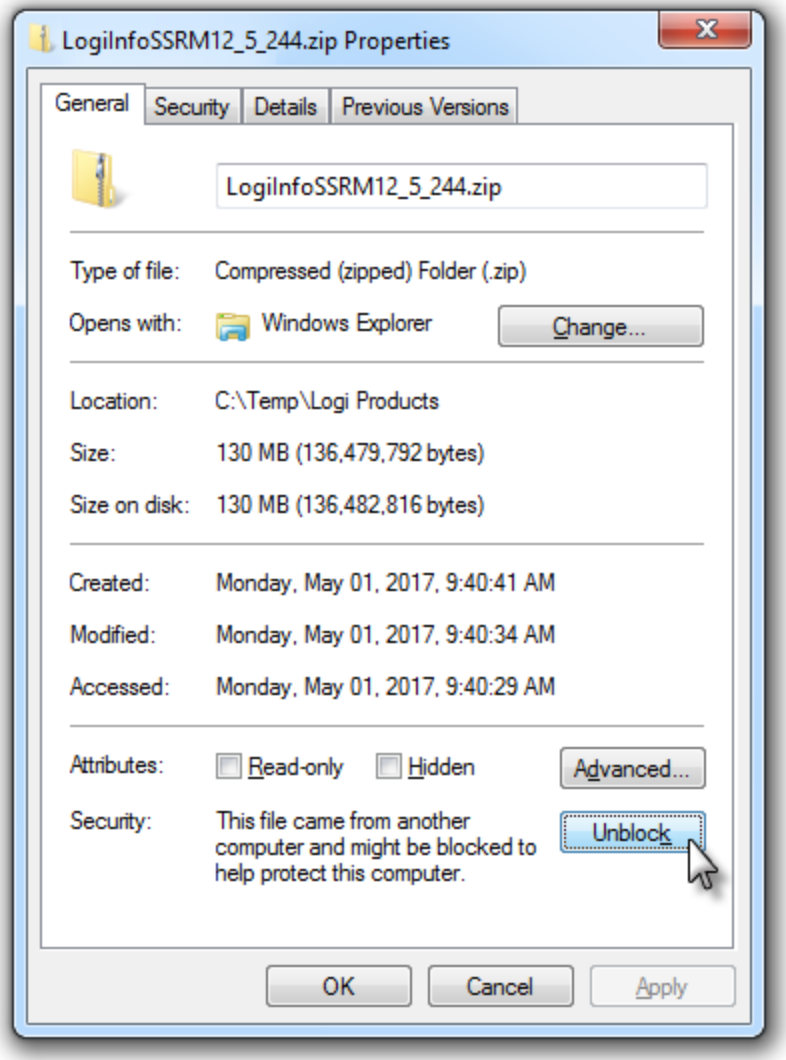
If your developers have customized gallery files to work with special Dashboards, they will most likely need to rework the application to accommodate this change.

After the migration, you may need to perform the following configuration tasks in your InfoGo application:

For Gallery files that have been processed and are retained in place, with .migrated and timestamp added to their file names, those files should be deleted after you are sure everything migrated properly.

Install the New Version

As with any downloaded installation file, please Unblock the file before extracting the installation executable:


LogiInfoSSRM12_5_244.zip Properties
General
Security
Details
Previous Versions
LogiInfoSSRM12_5_244.zip
Type of file: Compressed (zipped) Folder (.zip)
Opens with: Windows Explorer
Change...
Location: C:\Temp\Logi Products
Size: 130 MB (136,479,792 bytes)
Size on disk: 130 MB (136,482,816 bytes)
Created: Monday, May 01, 2017, 9:40:41 AM
Modified: Monday, May 01, 2017, 9:40:34 AM
Accessed: Monday, May 01, 2017, 9:40:29 AM
Attributes: Read-only Hidden
Advanced...
Security: This file came from another computer and might be blocked to help protect this computer.
Unblock
OK
Cancel
Apply

In Windows, right-click the .zip file you received from Logi and select "Properties", then click **Unblock**, as shown above. If the file is not blocked, the Unblock button is not displayed. Then extract the installation executable. In Linux, open the file properties and, in the Permissions tab, mark the file as executable, or use `chmod +x` to make the file executable.

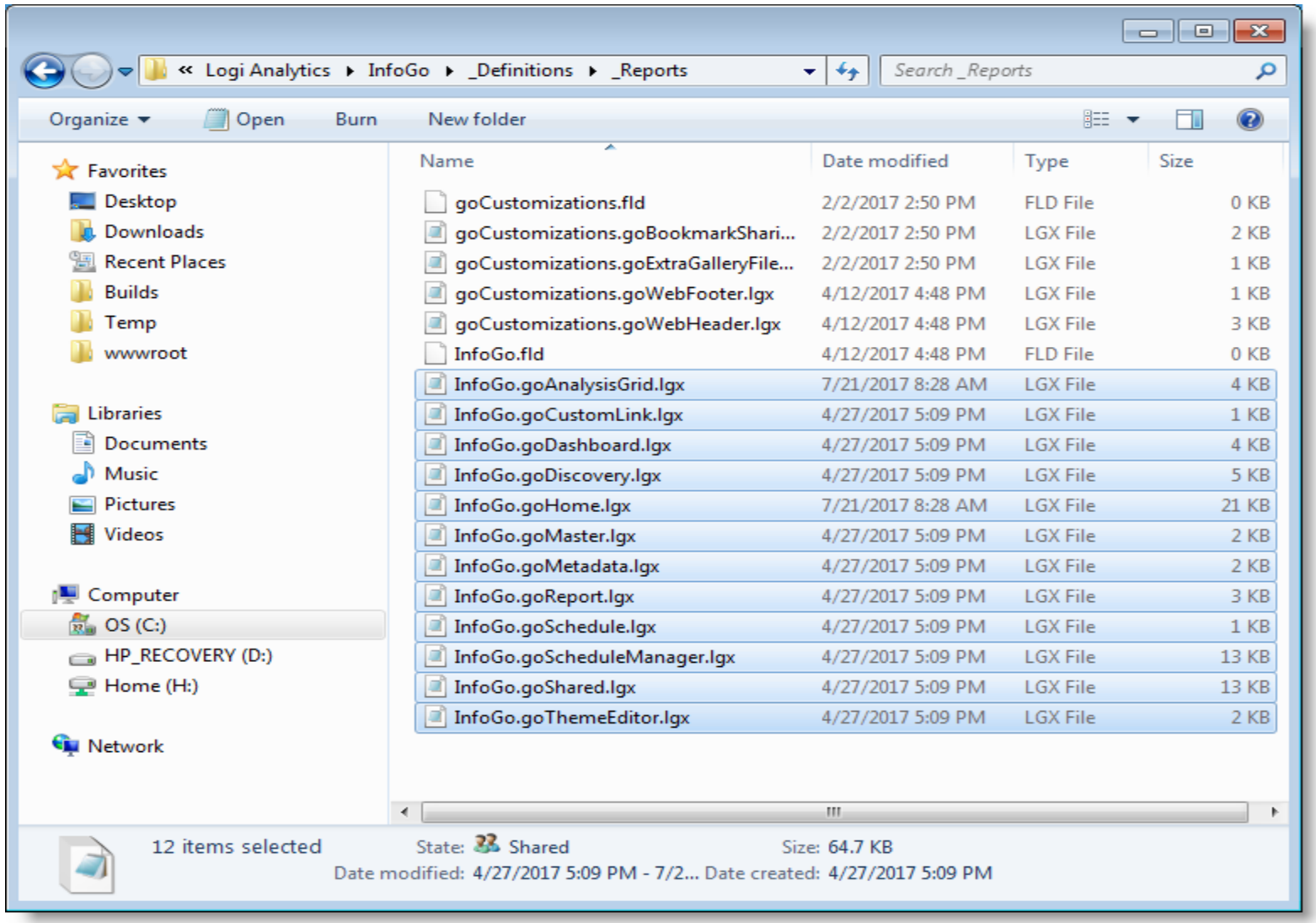
Run the installation executable file as an Administrator (in Windows, right-click and select "Run as administrator"). The installation steps will be the same as those shown in "Install the Self-Service Reporting Module" on page 189. Install the upgrade over the original or previous version, in the same location. You do not need to uninstall previous versions.

The rest of this topic assumes that you followed our Best Practices recommendation that you *not* run InfoGo from the SSRM installation folder (typically `C:\Program Files\Logi Analytics\InfoGo`) and that you copied the application elsewhere to create a working InfoGo instance. In the following instructions, we'll refer to the SSRM installation folder as the "original" folder and your working copy of InfoGo as the "target" folder.

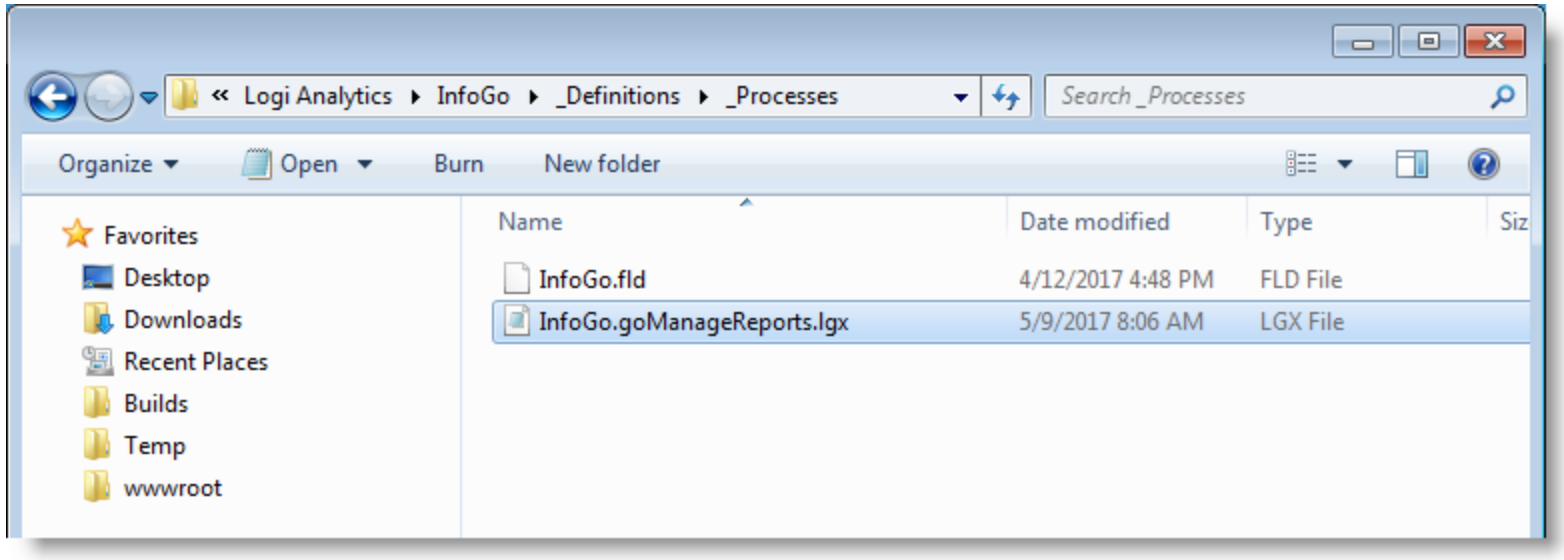
Copy the New Files

Start by copying files from the newly installed version to your working InfoGo copy, as follows:

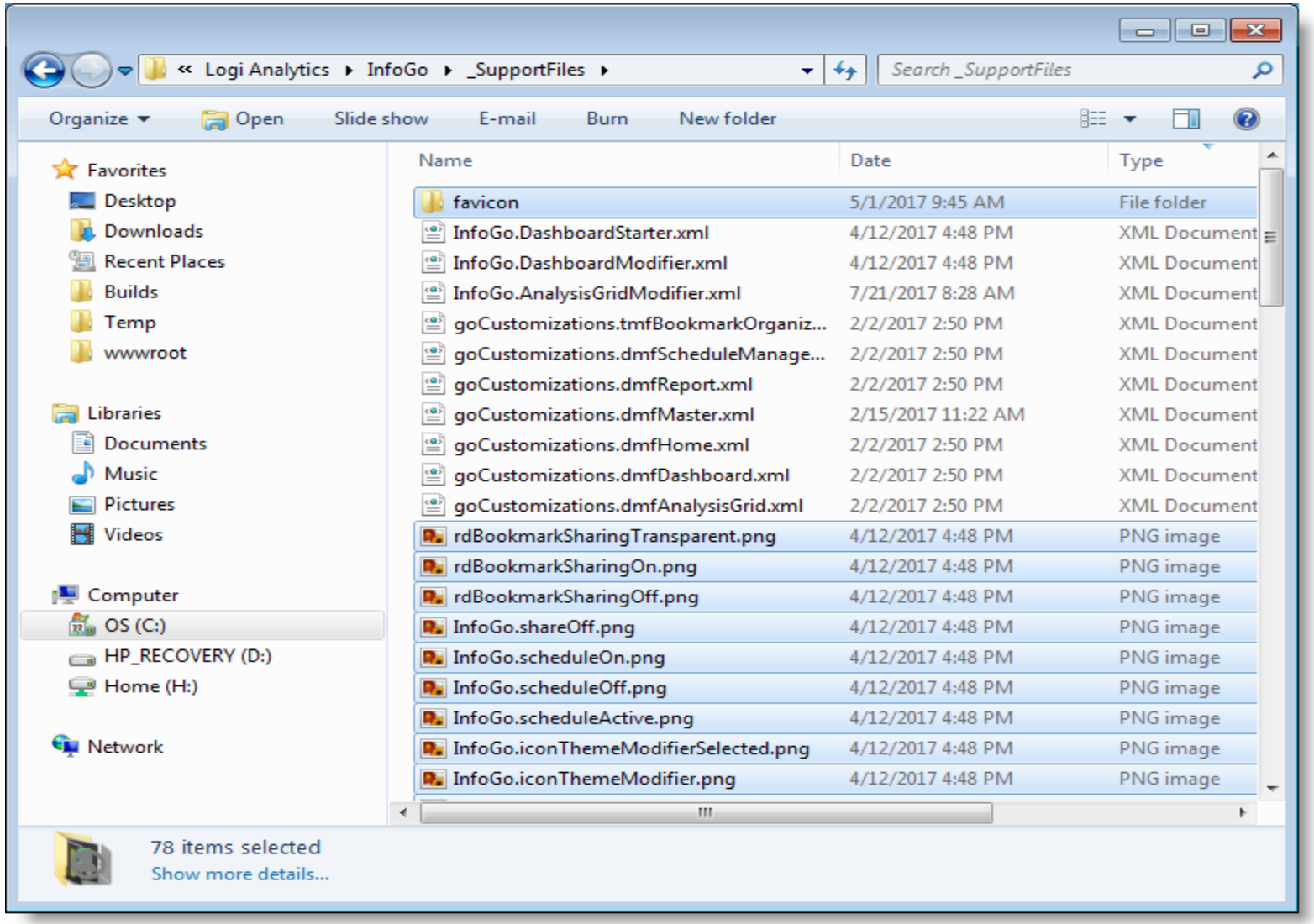
1. Copy all of the Report definition files with the `InfoGo`virtual folder prefix from the original `_Definitions\_Reports`folder to the target `_Definitions\_Reports`folder, overwriting the existing files:



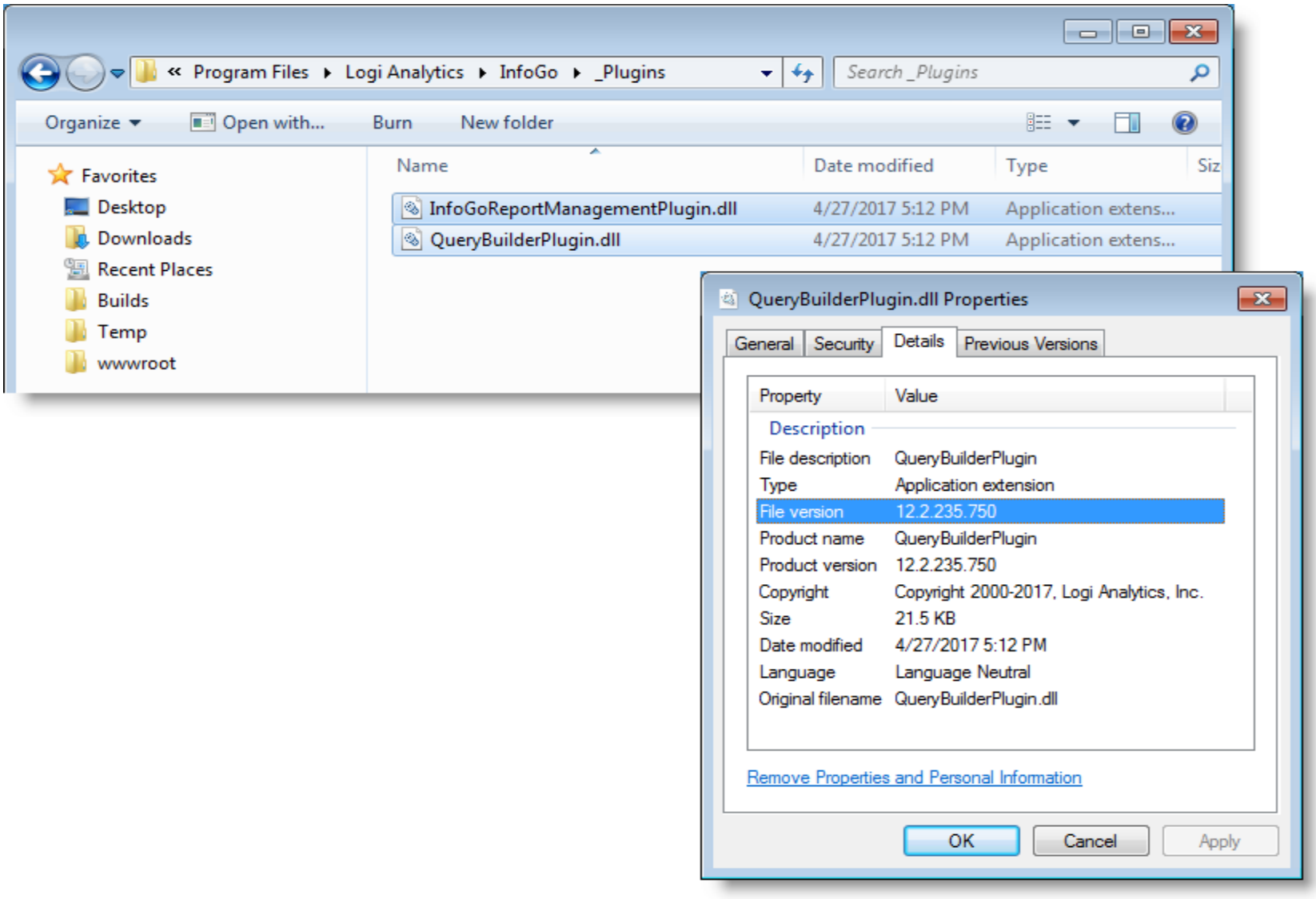
2. Copy the `InfoGo.goManageReports.lgx`file from the original `_Definitions\_Processes`folder to the target `_Definitions\_Processes`folder, overwriting the existing file:



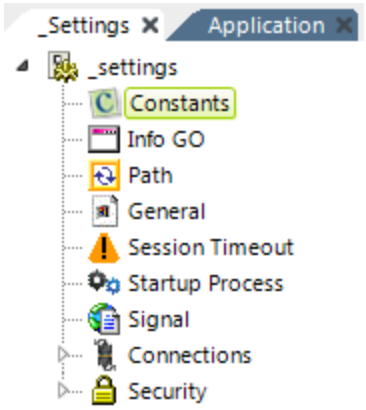
3. Copy all files from the original `_SupportFiles`folder, *except* those with the `goCustomizations`virtual folder prefix and `InfoGo.DashboardModifier.xml` and `InfoGo.AnalysisGridModifier.xml`, to the target `_SupportFiles`folder, as shown below. The image below does not show *all* of the selected files - there should be about 78 of them.


« Logi Analytics ▸ InfoGo ▸ _SupportFiles ▸
Search _SupportFiles
Organize
Open
Slide show
E-mail
Burn
New folder
Favorites
Desktop
Downloads
Recent Places
Builds
Temp
wwwroot
Libraries
Documents
Music
Pictures
Videos
Computer
OS (C:)
HP_RECOVERY (D:)
Home (H:)
Network
Name
Date
Type
favicon 5/1/2017 9:45 AM File folder
InfoGo.DashboardStarter.xml 4/12/2017 4:48 PM XML Document
InfoGo.DashboardModifier.xml 4/12/2017 4:48 PM XML Document
InfoGo.AnalysisGridModifier.xml 7/21/2017 8:28 AM XML Document
goCustomizations.tmfBookmarkOrganiz... 2/2/2017 2:50 PM XML Document
goCustomizations.dmfScheduleManage... 2/2/2017 2:50 PM XML Document
goCustomizations.dmfReport.xml 2/2/2017 2:50 PM XML Document
goCustomizations.dmfMaster.xml 2/15/2017 11:22 AM XML Document
goCustomizations.dmfHome.xml 2/2/2017 2:50 PM XML Document
goCustomizations.dmfDashboard.xml 2/2/2017 2:50 PM XML Document
goCustomizations.dmfAnalysisGrid.xml 2/2/2017 2:50 PM XML Document
rdBookmarkSharingTransparent.png 4/12/2017 4:48 PM PNG image
rdBookmarkSharingOn.png 4/12/2017 4:48 PM PNG image
rdBookmarkSharingOff.png 4/12/2017 4:48 PM PNG image
InfoGo.shareOff.png 4/12/2017 4:48 PM PNG image
InfoGo.scheduleOn.png 4/12/2017 4:48 PM PNG image
InfoGo.scheduleOff.png 4/12/2017 4:48 PM PNG image
InfoGo.scheduleActive.png 4/12/2017 4:48 PM PNG image
InfoGo.iconThemeModifierSelected.png 4/12/2017 4:48 PM PNG image
InfoGo.iconThemeModifier.png 4/12/2017 4:48 PM PNG image
78 items selected
Show more details...

4. Copy the files in the original `_Plugins`folder to the target `_Plugins`folder. Verify that each `.dll`file has been upgraded by right-clicking it, selecting Properties, and checking its version in the Details tab, as shown below. If you encounter an error while copying or the version number is incorrect, you may need to stop and restart your web server to unload the .dlls before trying to copy them again.



5. Open the original and target InfoGo applications and examine their _Settings definitions. Manually compare their Constants element parameters and add any new parameters from the original file to the target file.

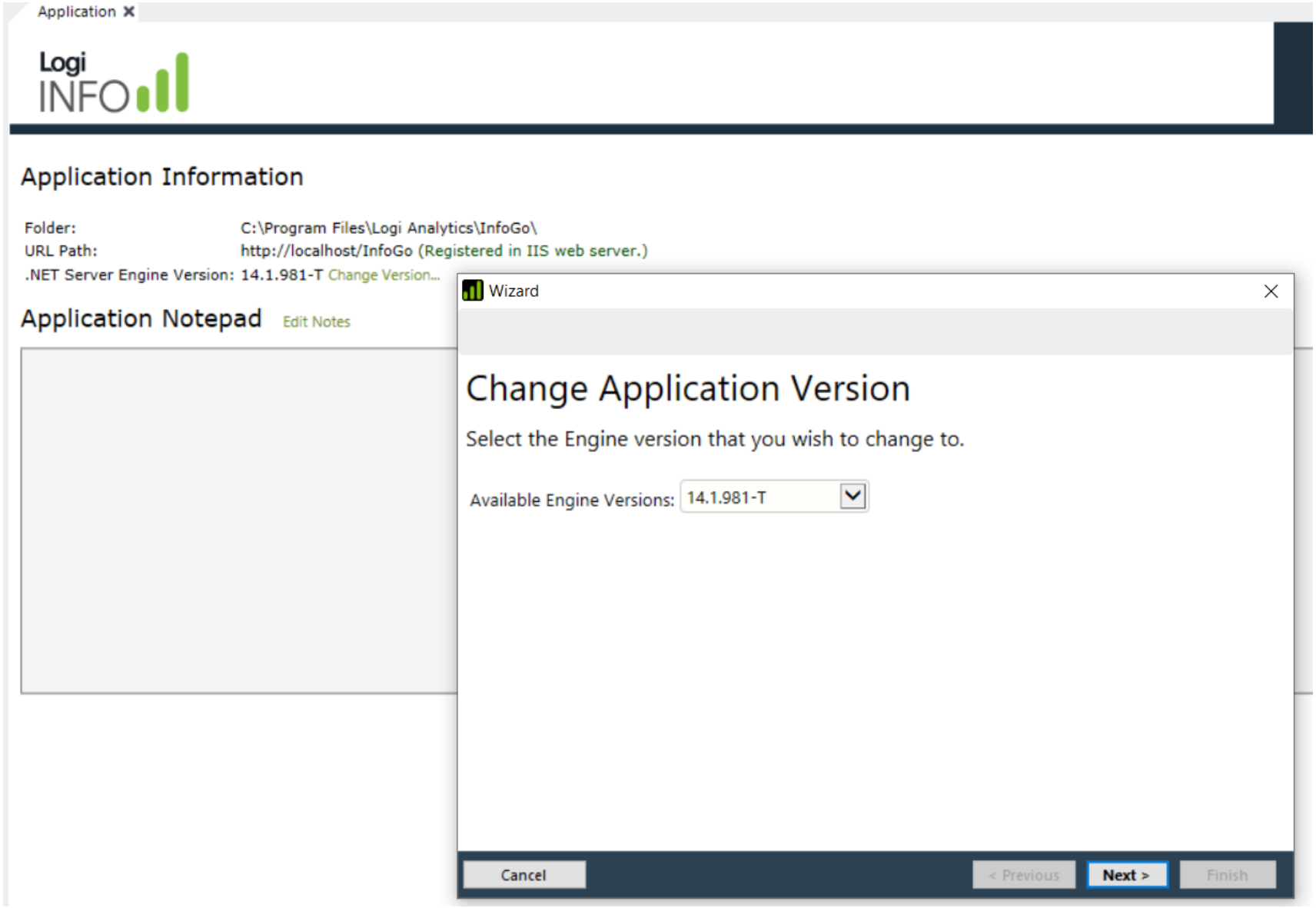


Element - Constants

Parameters	
goAllowSharing	True
goAnalysisType	UserSelectable
goBatchSelection	False
goCustomTableSessionVars	
goDashboardFilters	
goDashboardFiltersFromChartsDisabled	
goDefaultAnalysisName	Untitled Analysis
goDefaultBookmarkCollection	
goDefaultBookmarkID	
goDefaultBookmarkReportName	
goDefaultBookmarkUserName	
goDefaultDashboardName	Untitled Dashboard
goDefaultReportName	Untitled Report
goDefaultSharedBookmarkID	
goHideColumnSelection	False
goHomeName	Home
goMetadataIDsAnalysisGrid	metaNWLarge
goMetadataIDsDiscovery	metaJDBC
goReportStarterFile	
goSchedulerEnabled	True
goThemeEditorEnabled	True
SchedulerApplicationID	InfoGO

Upgrade the Logi Engine

Open the target application in Logi Studio and upgrade it to the desired Logi Engine version:


Application
Logi INFO
Application Information
Folder: C:\Program Files\Logi Analytics\InfoGo\
URL Path: http://localhost/InfoGo (Registered in IIS web server.)
.NET Server Engine Version: 14.1.981-T Change Version...
Application Notepad
Edit Notes
Wizard
Change Application Version
Select the Engine version that you wish to change to.
Available Engine Versions: 14.1.981-T
Cancel
< Previous
Next >
Finish

In Studio's Application tab, click the Change Version... link, as shown above, and then select the desired engine version and click **Next**, then **OK** or **Finish**. You can also do this using the Logi Server Manager, if desired.

Test Upgraded Application

Clear your browser's cache!

Run the application and use the Migration Utility to migrate visualizations into bookmarks, as discussed in "Upgrade Considerations" on page 199.

Run the target application and ensure that all security, sharing, analysis, report, and Dashboard-authoring features work properly. Open a saved analysis and retest.

Upgrading to Earlier Versions

This topic introduces considerations you may need to be aware of if you're upgrading to an earlier SSRM version.

Upgrading to SSRM v12.0

The tool used in InfoGo to create analyses now has the option to apply changes made to chart and table options as a "batch". When turned on, users can make several changes, then click an OK button to apply them. Normally, changes are applied immediately as each change is made. The batch approach may make for a smoother user experience, especially with large data sets. A new constant, goBatchSelection, controls this behavior. If you upgrade, this constant will not be added to the _Settings definition automatically, but you can add it manually in the source code and set it to True if you want to use this feature. You will also have to set the attribute that uses this constant:

1. Open the goAnalysisGrid definition.
2. Under the BodydivGoContainerdivMainContent elements, locate the "ag" element.
3. Set its Batch Selection attribute to @Constant.goBatchSelection~.
4. Save the definition.

In addition, the deployment strategy you've chosen has ramifications when you install SSRM updates. When installing an update on top of an existing version in the standard installation location, the installer program will not overwrite InfoGo definitions you've modified. So, for example, your customizations won't ever be overwritten. However, if you also have other instances of InfoGo, you'll have to copy updated files to them and then you are responsible for deciding what gets overwritten.

Upgrading to SSRM V12.1

Logi Info v12.1 allows developers to create a custom theme for use with InfoGo using a new tool, the Theme Editor. The new version of InfoGo provides an option for Administrators to modify that custom theme at runtime. You will need to configure a

constant and users will need appropriate security rights to make this feature available. These are discussed in the *Configuring InfoGo Constants* and *Customizing InfoGo Files* topics.

InfoGo v12.1 now uses the Logi Info Master Report (see) feature to provide a header, a footer, and a left sidebar menu. Nonetheless, your existing **goWebHeader** and **goWebFooter** definitions *should* work without modification.

However, if you've customized the goWebHeader definition and you'd like to use the new sidebar menu after upgrading, you should review and backup your existing goWebHeader definition, then delete it, *before* installing the new SSRM version. This will ensure that the new goWebHeader definition is installed. You'll then need to customize it as desired, using the old definition as a guide if necessary.

Alternately, if you've developed your own menu system and *do not* want to use the sidebar menu, leave your goWebHeader in place and run the installation. To be thorough, after the installation remark the Left Frame in the goMaster definition.

Upgrading to SSRM v12.2

The InfoGo side-bar menu has changed and there have been some styling changes to the stock Signal theme. A new "Data Manager" tool, based on the Web Metadata Builder, is available along with the Schedule Manager and the Theme Editor and a new Security Right ID has been provided to control access to it.

In the InfoGo application, Dashboard panels no longer resize themselves horizontally to fit the size of the chart they contain. Charts in Dashboard panels are automatically displayed at or near the largest width feasible, and charts no longer have horizontal resizing handles. This ensures good cross-browser and cross-device compatibility.

Install the Discovery Module

The Logi Discovery Module (DM) provides an embedded data discovery experience that works directly with relational data sources using Logi data services. It does this by making several special elements available in Logi Studio. These include the **Thinkspace** element, which provides a highly-interactive analysis experience. In its interface, best-fit visualization suggestions are automatically referred to the user by a built-in "recommendation engine" and data is organized using intelligent technology that clusters data to make it more easily consumable by users.

The DM is installed separately, after Logi Info and, if using it, the SSRM are installed. Please note these important restrictions:

 Advanced features discussed here work with Logi Info v12.5 and later, 64-bit version (there is no 32-bit version available). Earlier and later Info versions may not support them; consult the Release Notesfor specific details.

Databases supported include:

- AWS Redshift
- Infobright
- JDBC database providers
- MS SQL Server
- MySQL
- Oracle
- PostgreSQL
- Snowflake
- Vertica

Logi Info applications on IIS that utilize DM v3.1+ must run using an Application Pool configured for *Integrated* pipeline mode. *Classic* pipeline mode Application Pools will *not* work.

If you're using Internet Explorer 11, the Internet Options → Security setting for applications using this technology must be set to *Medium High* or lower.

Important Java Information

Starting in January 2019, Oracle discontinued its free, long-term support of the **Oracle Java Development Kit** (JDK) for enterprises. The impact of this on Logi products is discussed in detail in "Java Usage Policy" on page 342.

 As a result, Logi no longer distributes the Oracle JRE as part of the Discovery installation. *Prior to installing Discovery 3.2, you must have OpenJDK 12 installed; Oracle JRE is not supported.*

When you install OpenJDK, you must configure your Windows SystemVariables properly. See "Install the Discovery Module - Windows - Setting Windows SystemVariables for OpenJDK" on page 238 for more information.

What Gets Installed

The following components will be installed on the web server:

- An embedded, runtime environment for server-side web applications (node.js)
- An embedded, Java-based relational database (H2)

Node.js and H2 are extremely compact, have a small installation footprint, and do not install a lot of extra tools or management utilities. They're provided as transparent services that require little or no management.

Two Windows services are created during the installation: *Logi Application Service* and *Logi Data Service*.

Important Upgrade Information

If you're upgrading from earlier versions of Discovery, please *uninstall* the earlier version first, using Control Panel, and at the end of the process be sure to respond "Yes" when prompted to remove logs and the database. Then run the Discovery 3.1+ installer to create a fresh installation.

Release Pairings

Information about which Logi Info and DM versions should be used together is available in *Add-on Modules*. For more information, see the Release Notes for each Logi Info version.

DM3.2 Product License

Unlike previous versions, DM v3.1+requires its own license. The installer will provide a built-in **15-day trial license**. You don't need to do anything but install the product and you can begin using it immediately. After the trial period expires, you will not be able to use certain DM functions, such as authoring Dataviews and saving Thinkspace visualizations.

Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licences, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

You *may not* use our products for redistribution with, or embed them in, other products without an OEM license; contact our Sales group for more information if you need OEM licenses.

Install the Discovery Module - Windows

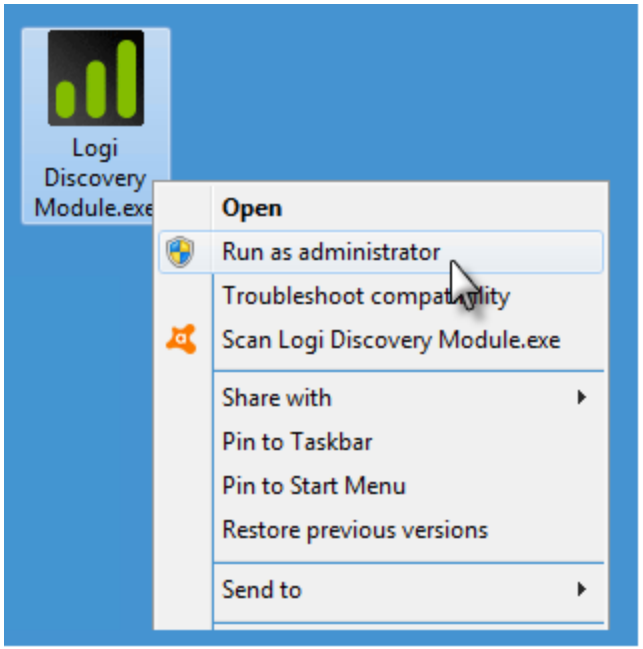
The following topics guide you through installing the Discovery Module on a single computer under the Windows operating systems:

- Preparing to Install
- Installing the Logi Discovery Software
- Customizing Service Settings
- Setting Windows System Variables for OpenJDK
- Removing the Discovery Module Installation

Install the Discovery Module - Windows - Preparing to Install

Logi Info must be installed and licensed *before* the DM is installed. *After* the DM is installed, you'll also need to acquire its separate license file.

It's critical that installation and configuration occur while running as the built-in "Administrator" account.



Even if your personal account has been added to the local Administrators Group, it may not have sufficient privileges, so don't rely on it.

As shown at left, on Windows the correct practice when running the Logi installer or using the Command Line to make configuration adjustments is to start the tool by *right-clicking* its icon and selecting "**Run as administrator**" from the menu to start the program.

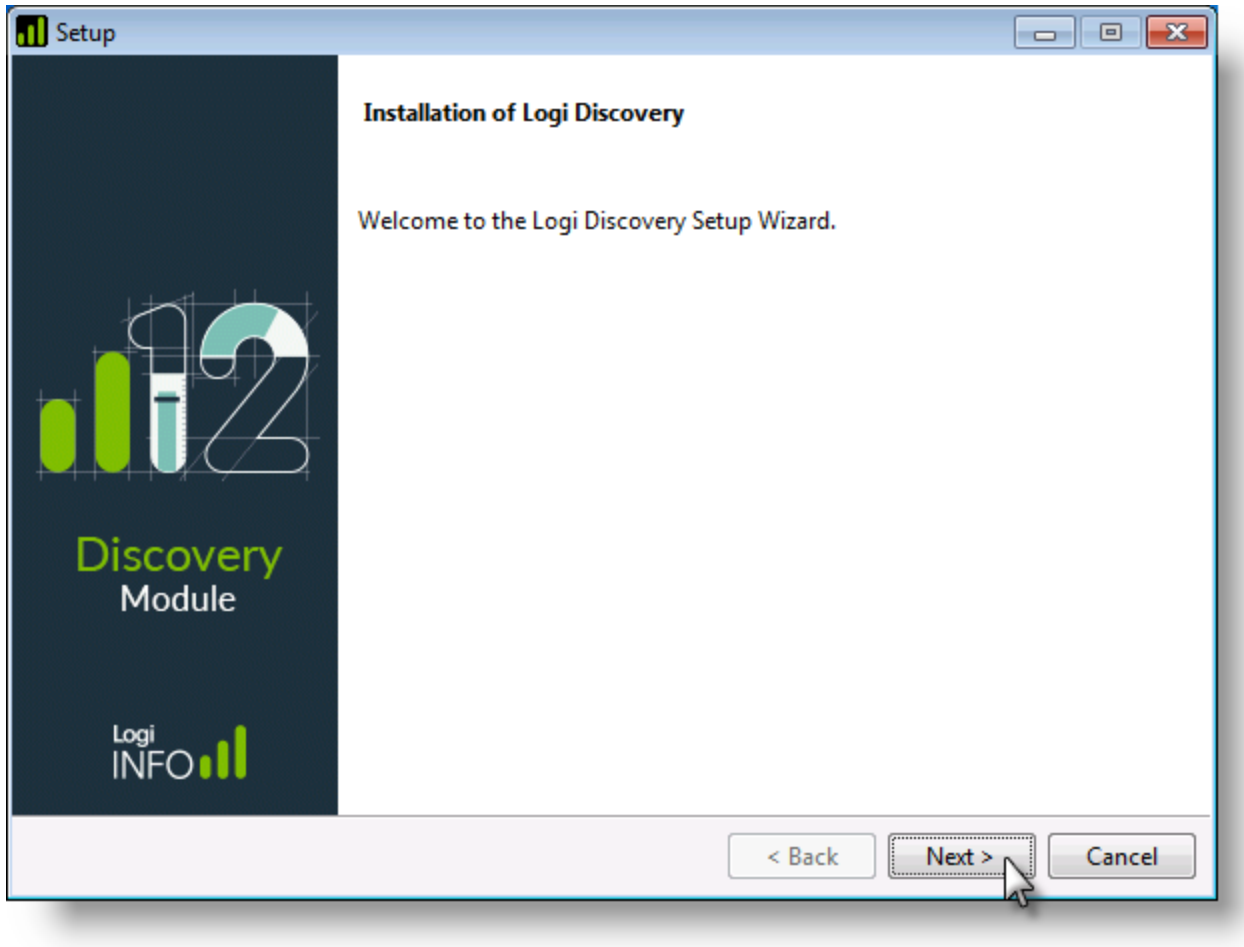
This ensures that appropriate permissions are provided for the installed components.

Don't see a "Run as administrator" option? If the system is in a network domain, your network admin may have created security policies that don't allow you to see this option, in which case you need to consult your IT staff for assistance.

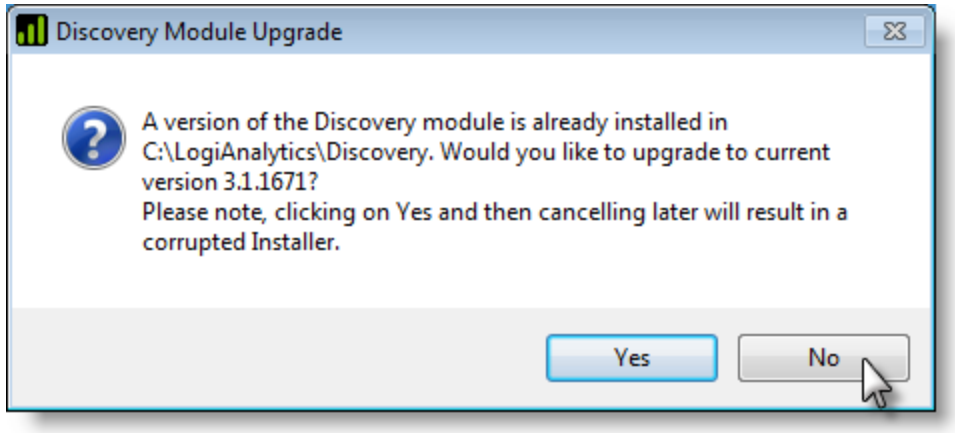
Install the Discovery Module - Windows - Install the Logi Discovery Software

Follow these steps to install the Logi Discovery Software.

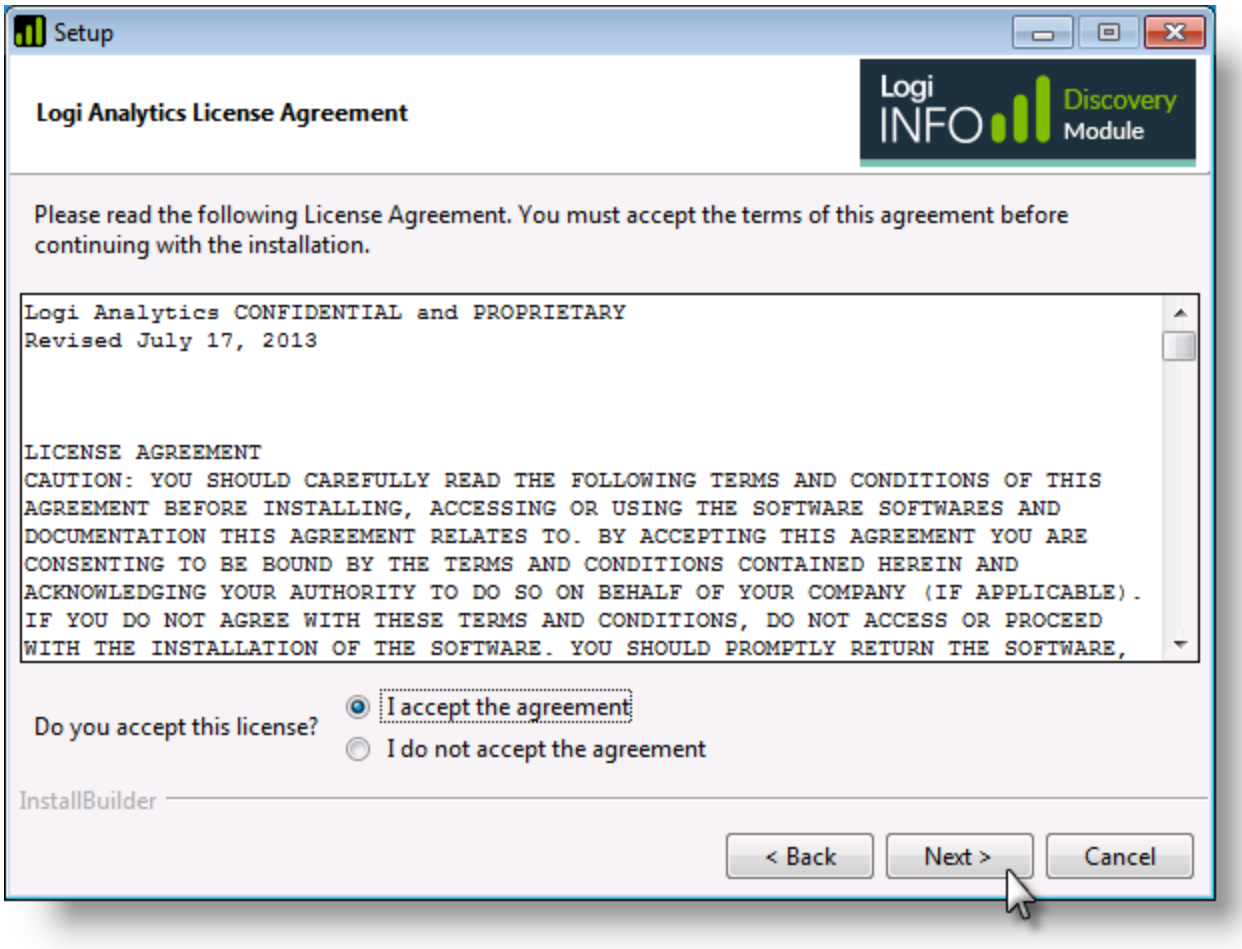
As usual, you can click **Back** at any time before the physical installation begins to go back to the previous screen.



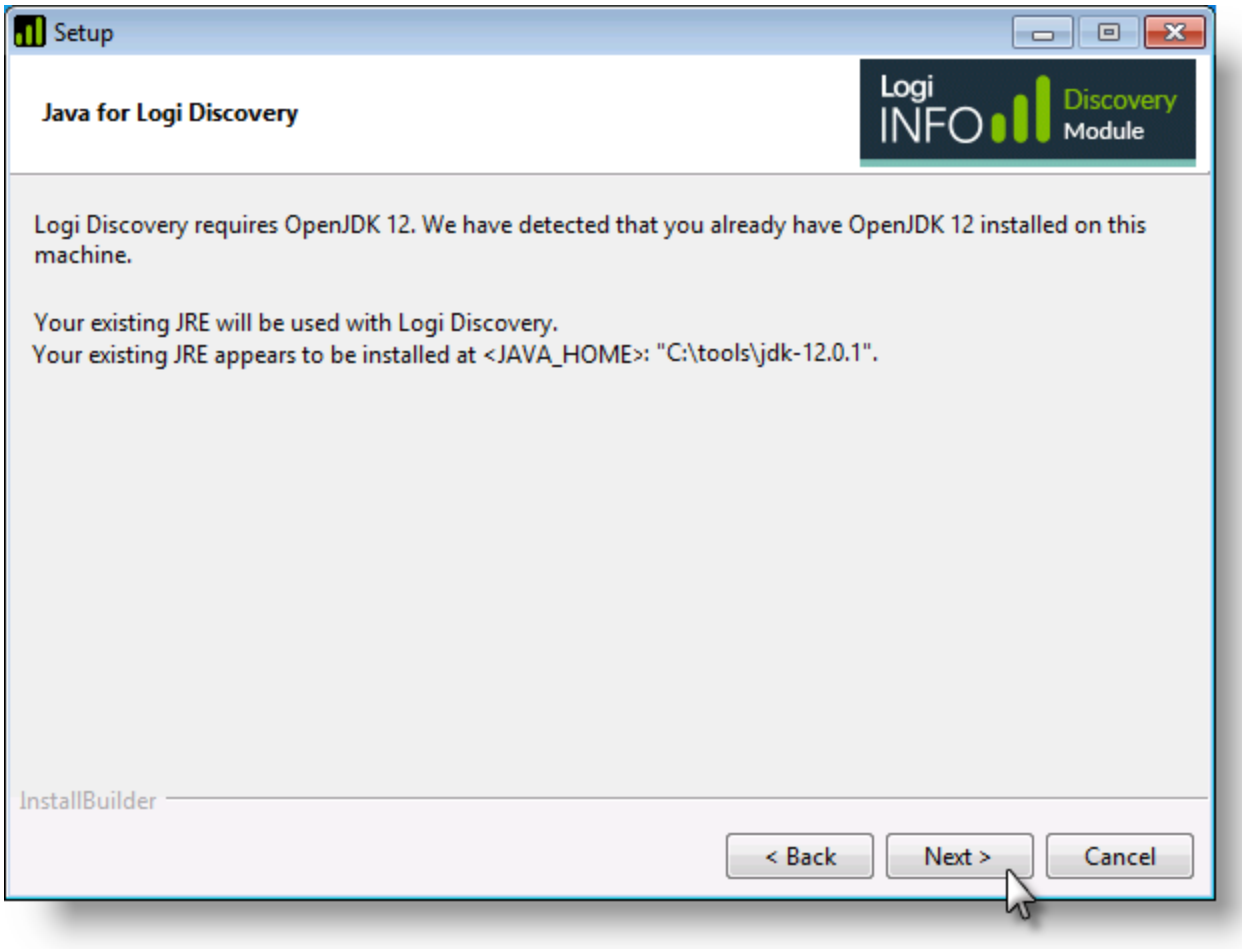
1. To start the installation, *right-click* the Logi product installation file icon and select "Run as administrator" to launch the installer. When the **Welcome Screen** appears, click **Next**.



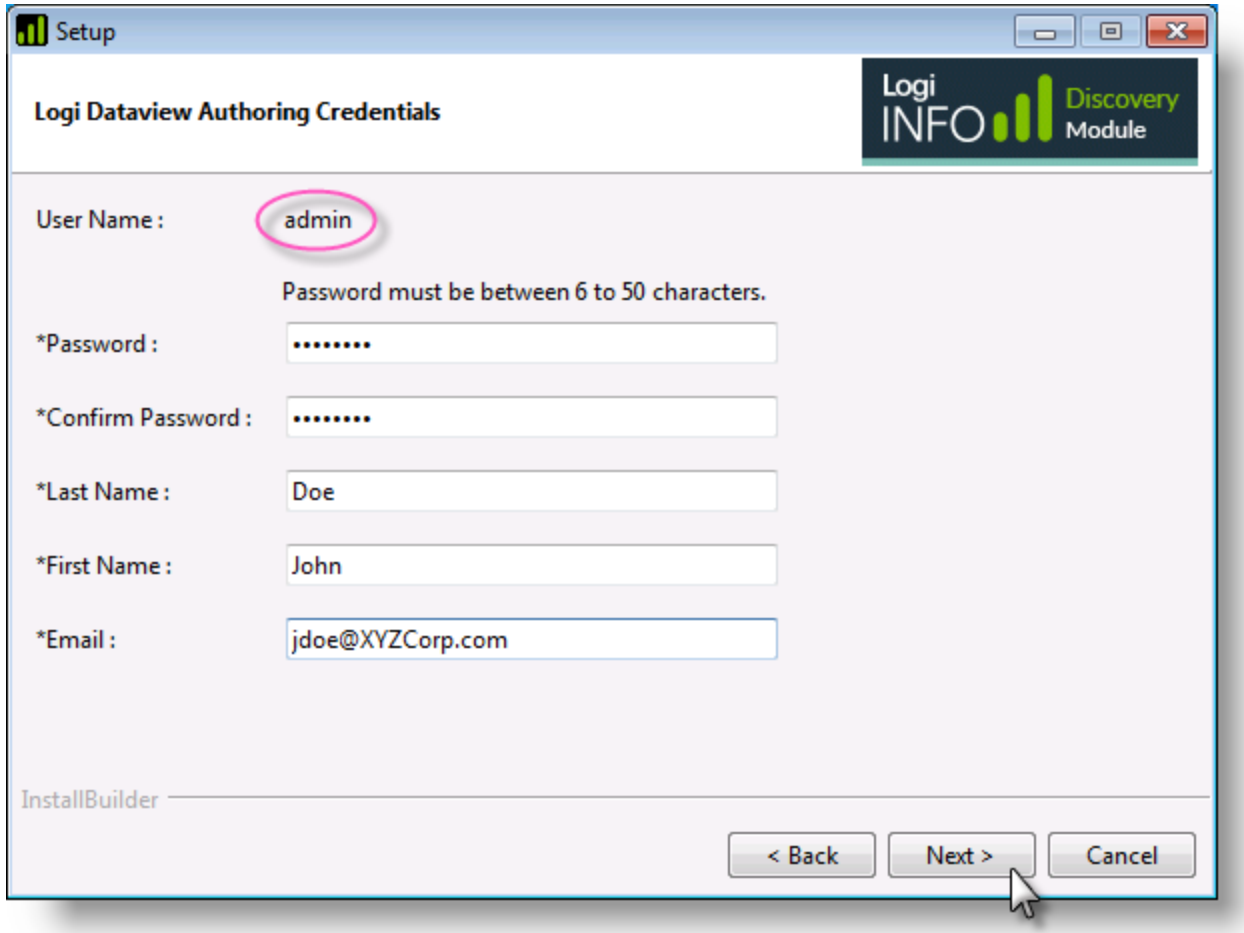
If you see the dialog box shown above, click **No** and the installer will close. Uninstall the existing DM version and then re-run the DM v3.1+ installer.



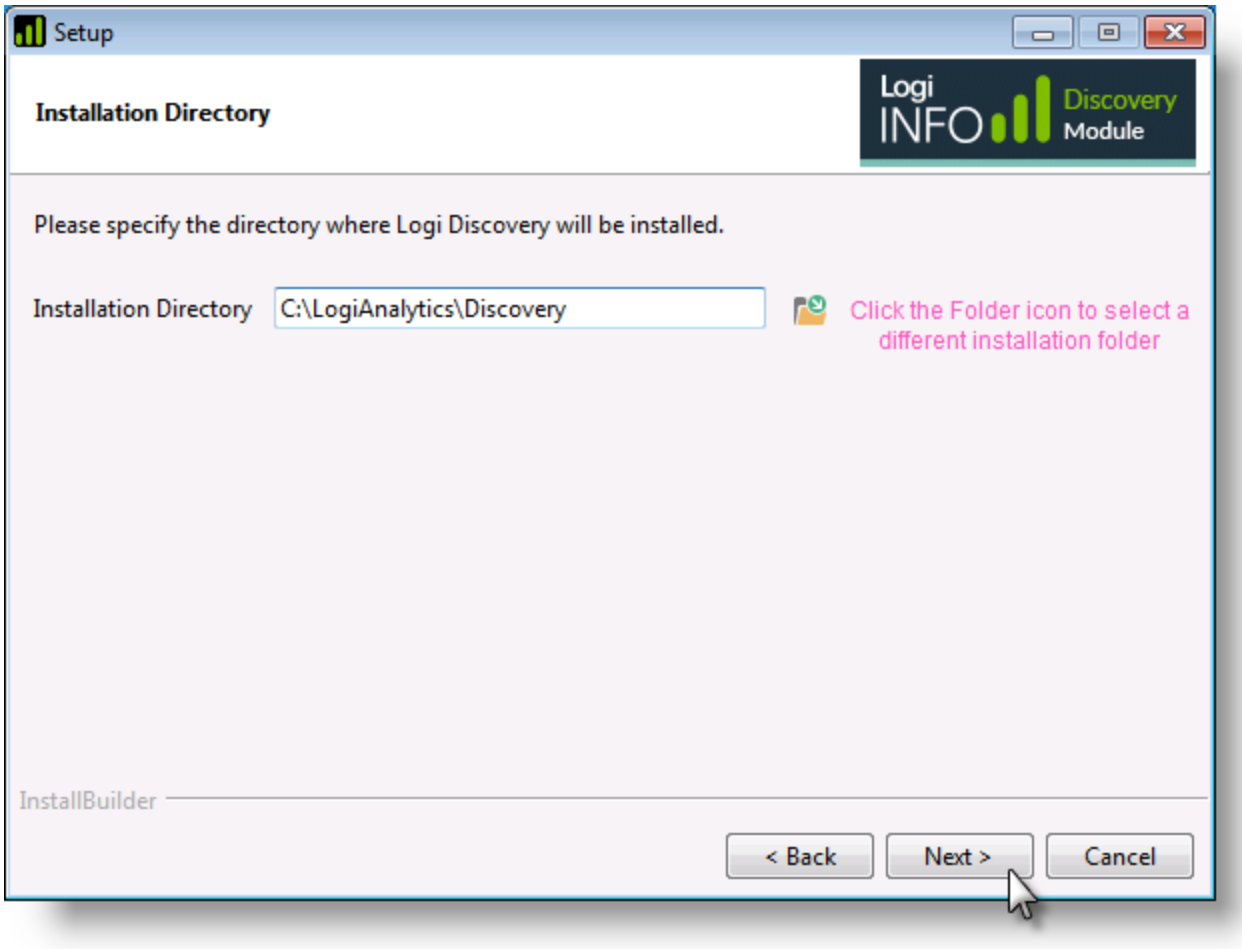
2. **License Agreement**: Select the "I accept the agreement" radio button after reading the license agreement and click **Next** to continue.



3. **Java Acknowledgement:** Next, if you have OpenJDK installed, you'll see the screen shown above. If you don't, an error message is displayed, along with a link to download OpenJDK. If you have it installed but haven't set the related Windows SystemVariables, an error message is displayed. You *must* install it before proceeding. If the link in the window does not resolve correctly, you can go here to get OpenJDK12: JDK12 Download from Archive. Click **Next**.



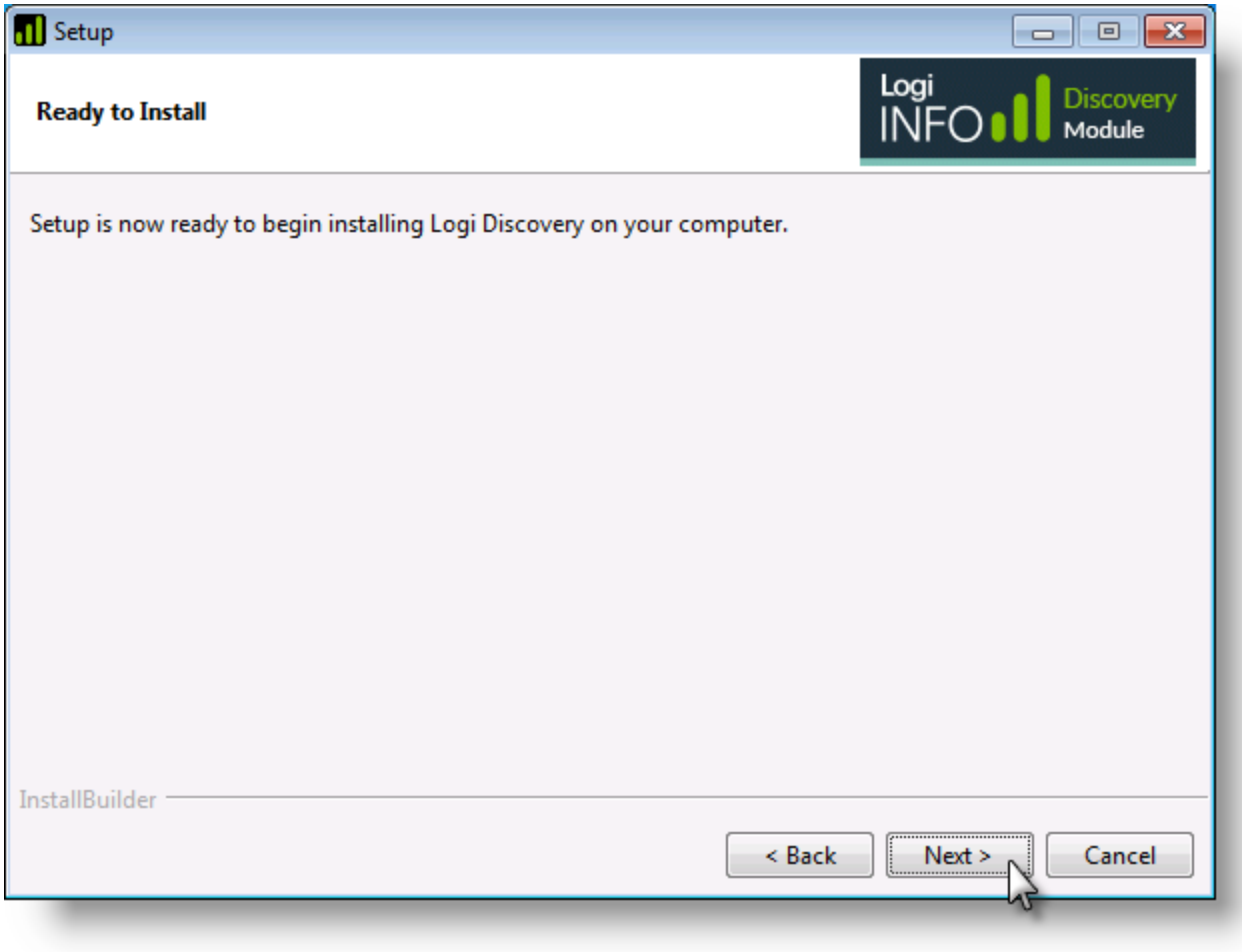
4. **Dataview Authoring Credentials**: (Not shown for upgrades) Provide the required information for the default Dataview Authoring "admin" user, as shown above. Click **Next** to continue.



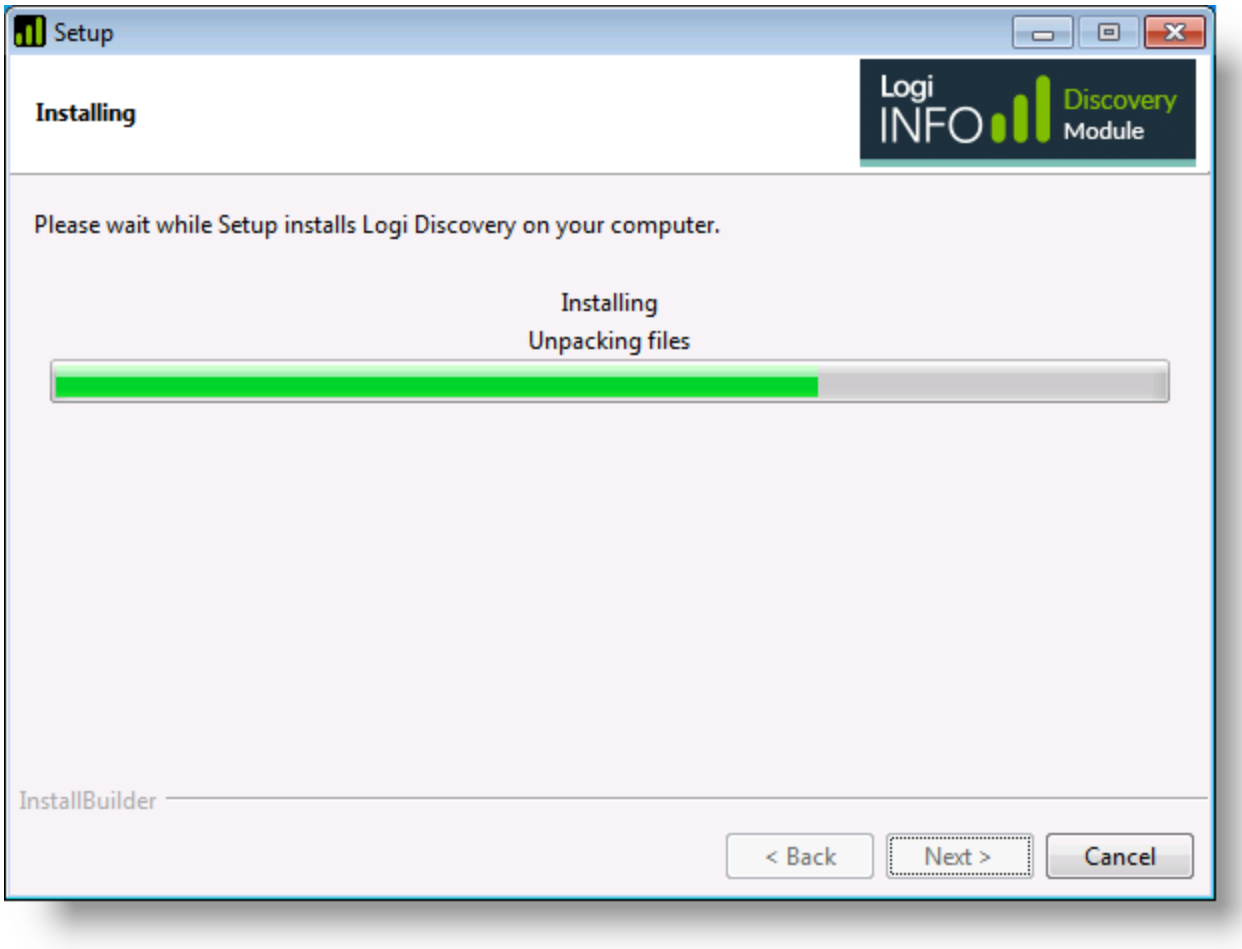
5. **Installation Directory:** Click **Next** to accept the default installation location and continue. *Optional* - click the Folder icon to specify an alternative installation location if you don't like the default location.

   You may not install directly to the root folder of a drive, i.e. `C:\` and if you specify a custom installation directory, the folder names in the path must not contain *spaces*. For example, this path is *not valid*: `C:\Program Files\Discovery`.

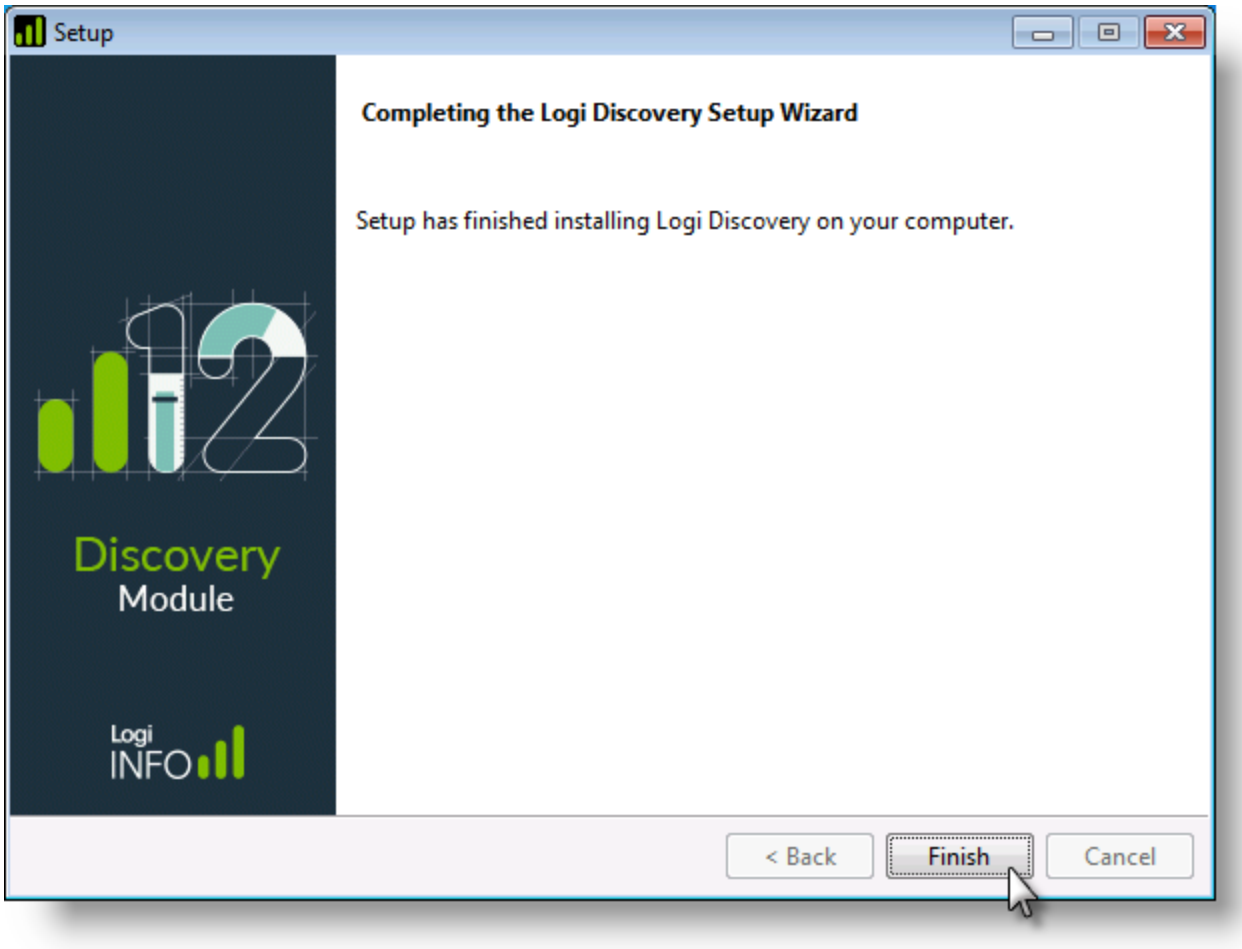
In addition, the *length* of the path must not exceed 260 characters.



6. **Ready to Install**: Click **Next**.



7. The physical installation will begin and you'll see a progress indicator for different tasks.



8. **Installation is complete**: Click **Finish** to exit the installer.

Install the Discovery Module - Windows - Customizing Service Settings

You may wish to customize Logi services settings which, assuming a default installation location, can be found in the LogiApplicationService and LogiDataService files:

```
<installationFolder>\platform\settings\logiApplicationService.json
<installationFolder>\platform\settings\logiDataService.json
```

The beginning of the `logiApplicationService.json`file looks like this:

```
      {
      "disableSchemaValidation": false,
      "logiApplicationService": {
      "description": "Platform
 logiApplicationService Configuration",
      "logLevel": "error",
"corsOriginWhiteList": "*",
      "accessTokenGracePeriod": 120,
      "system": {
      "pollingIntervalSeconds":
 15,
      "pollingTimeoutMinutes":
 -1
      },
```

For example, the required `corsOriginWhiteList` attribute, highlighted above, specifies which servers the service will respond to, allowing you to restrict access to the service, if desired. The default value, "*", tells the service to respond to *all* servers; to restrict access enter a single IP address or server name (as a string) or a JSON array of them, such as:

```
["http://server1","http://server2","http://192.168.1.1"]
```

Changes to some settings may cause applications to fail. For example,

```
39      "port": 3000,
40      "hostname": null,
41      "proxyUrl": null,
42      "platformBaseUrl": "/api/platform"  !!
```

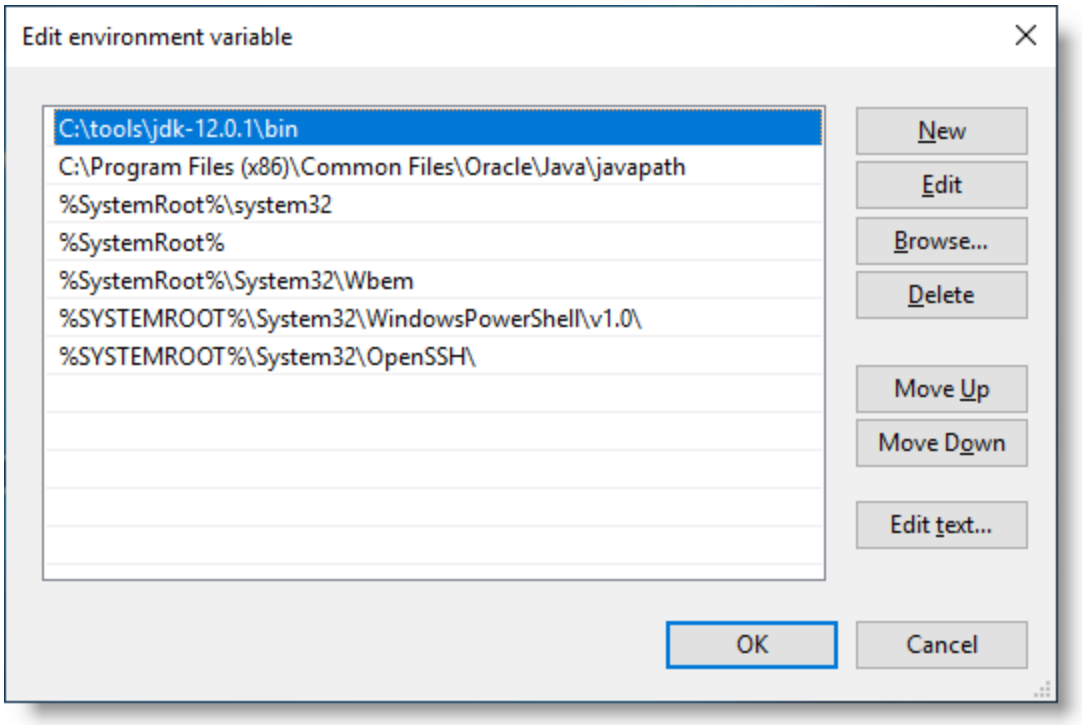
changing the `platformBaseUrl` setting, shown above, may cause integrations with Logi Info applications to fail.

You'll need to stop and restart both services to have any changes to the settings file take effect.

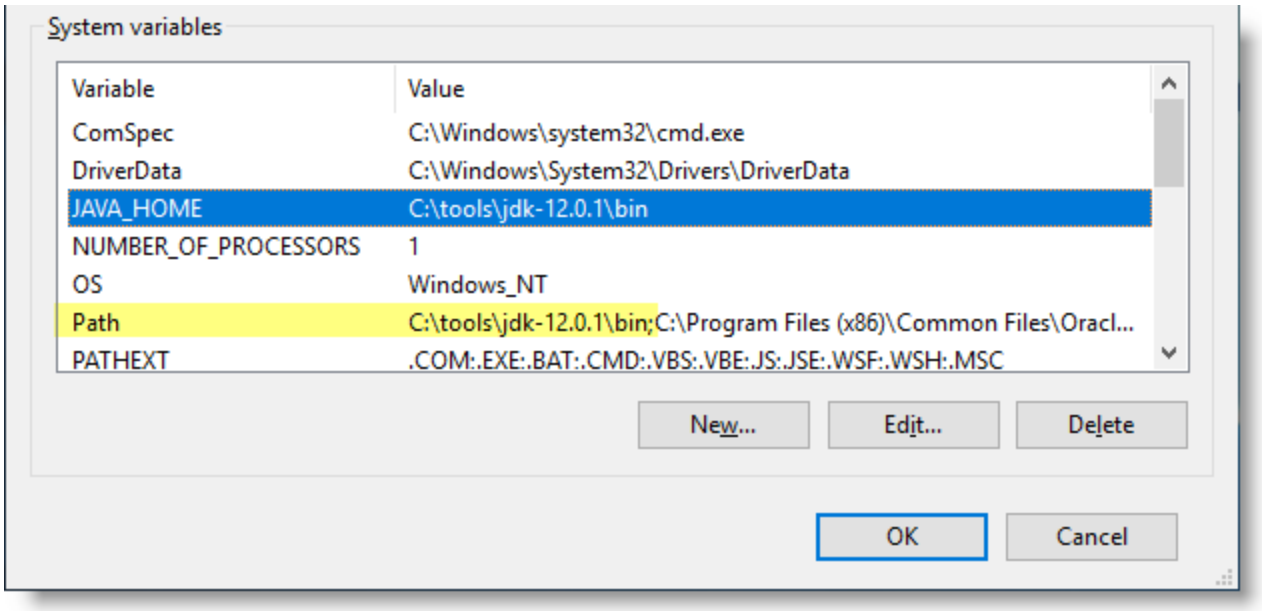
Install the Discovery Module - Windows - Setting Windows SystemVariables for OpenJDK

There is no installer for OpenJDK. You download and unzip its files and place them on your machine, then configure SystemVariables to point to them, as shown in these instructions.

These examples assume you've place the unzipped OpenJDK folder in `C:\tools`.



Navigate to the Environment Variable manager in your OS and add the path for the OpenJDK `bin`folder to the Path system variable, as shown above. Be sure to add the reference to the OpenJDK folders at the top (front) of the Path string, as shown.



Next, create/set the JAVA_HOME system variable to the path for the OpenJDK `bin`folder, as shown above. The Path variable from the previous step has been highlighted for reference.

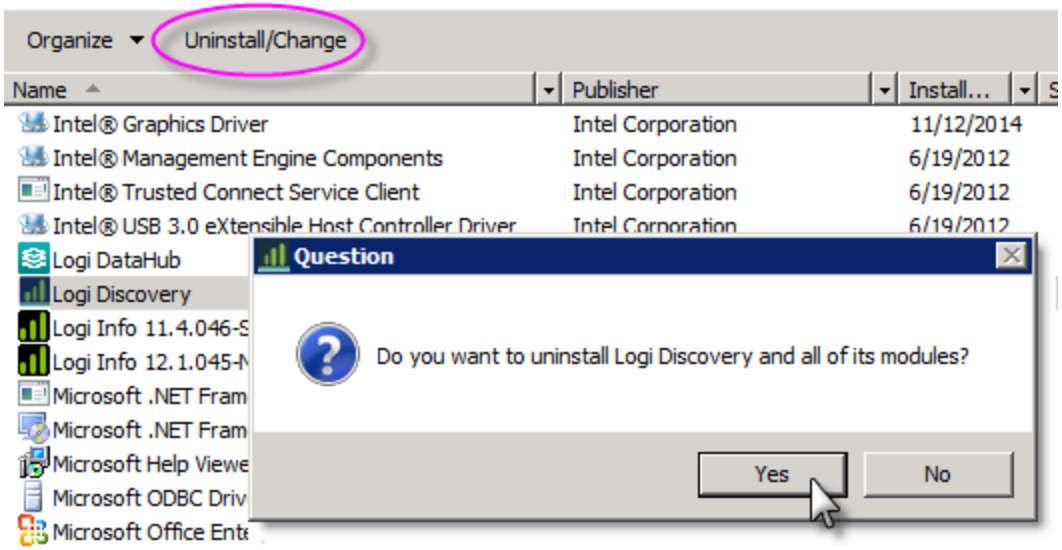
Click **OK** to close any related open windows.

Test your variables by opening a Command window and entering:

```
echo %Path%
echo %JAVA_HOME%
```

Install the Discovery Module - Windows - Removing the Discovery Module Installation

You can remove the DM completely by selecting it in Windows' Control Panel applet, Programs and Features, and clicking **Uninstall/Change**:



Select "Yes", as shown above, when prompted to uninstall the product, and "Yes" again if prompted to remove logs and database files. If JRE 8 was installed as part of the DM installation, it will be uninstalled, but if you had JRE 8 installed prior to installing the DM, it will not be uninstalled as part of this process.

Install the Discovery Module - Linux

The following topics guide you through installing the Discovery Module on a single computer under one of the Linux operating systems:

- Preparing to Install
- Installing the DM Software from a GUI
- Installing the DM Software from a Command-Line
- Configuring the Services (with systemctl)
- Configuring the Services (without systemctl)
- Customizing Service Settings

Install the Discovery Module - Linux - Preparing to Install

Logi Info must be installed and licensed *before* the DM is installed. *After* the DM is installed, you'll also need to acquire its separate license file.

 It's critical that installation and configuration occur while logged-in as the "root" user.

In this topic, we show instructions and examples of installations on **Centos 6.9** and **Ubuntu 16.04** and for systems with, and without, systemctl. Depending on your specific OS and distribution version, some of the commands or steps shown may need to be adapted.

Download the Discovery installer from the link provided to you by the Logi team, which will be similar to:

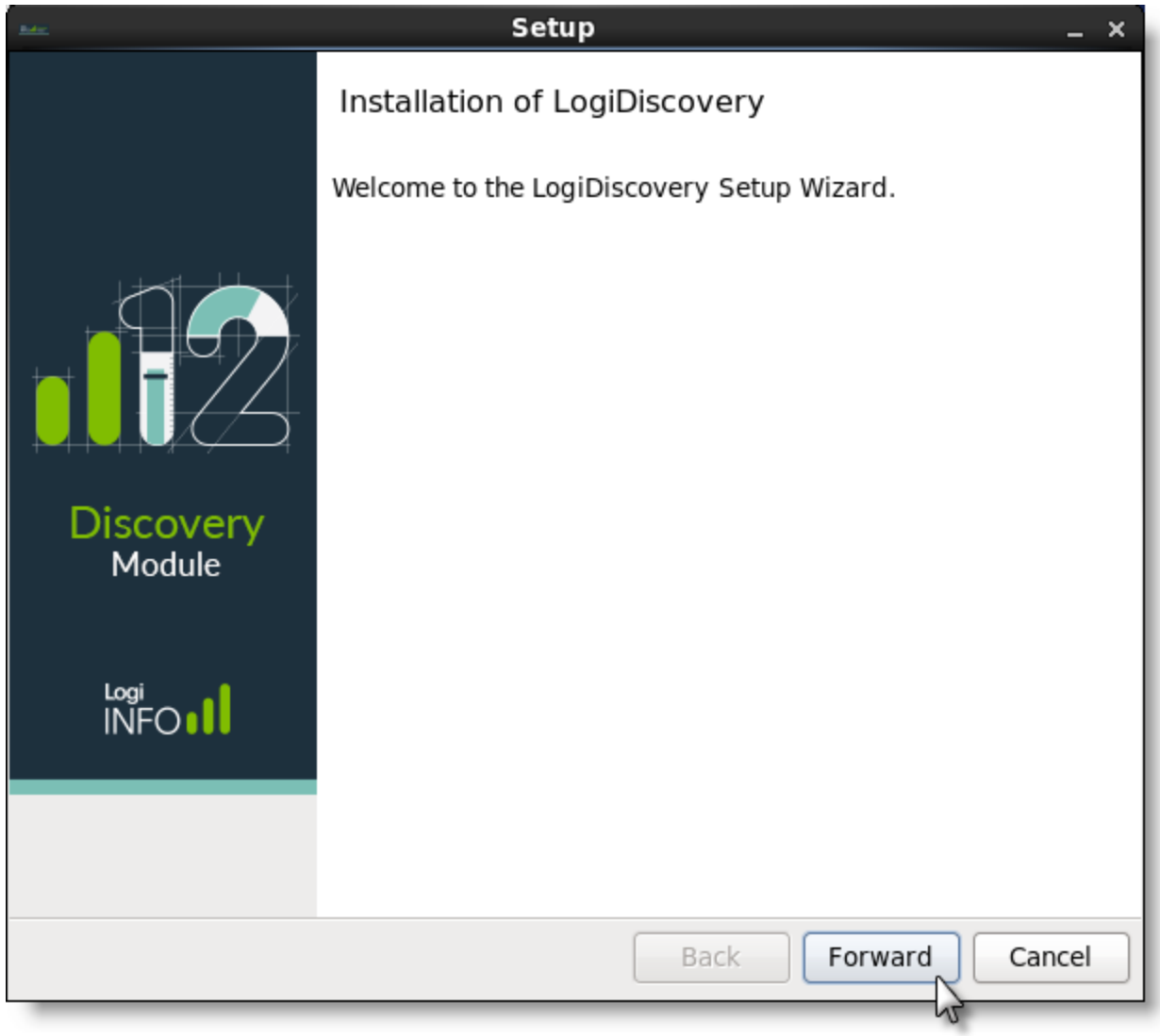
```
curl -O https://downloads.logianalytics.com/info/xxxxx-4538-489e-a5b1-18e1ca714458/Discovery3_02inux_x64in-
staller.run
```

Change the permissions on the Discovery installer so that it can be run:

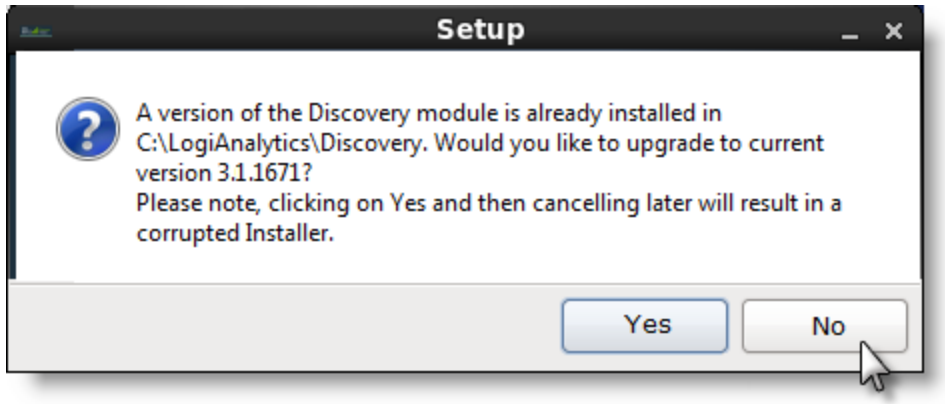
```
chmod +x Discovery3_0linux_x64installer.run
```

Install the Discovery Module - Linux - Installing the DM Software from a GUI

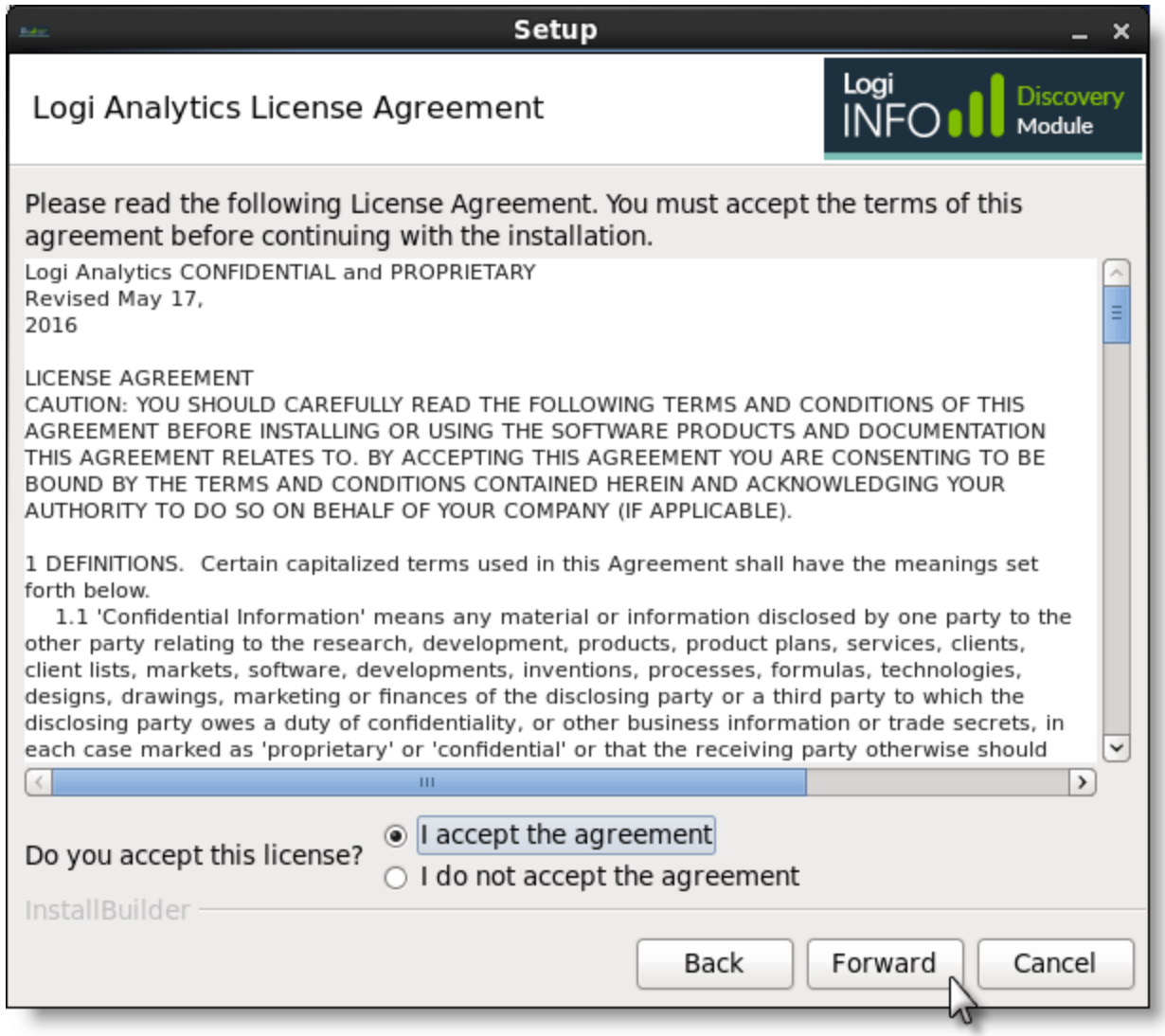
If you prefer to run the installation in a GUI, you can either use "sudo" to run it in a terminal session, or run it directly in the GUI if logged in as "root". These examples are from Centos 6.9. As usual, you can click **Back** at any time before the physical installation begins to go back to the previous screen.



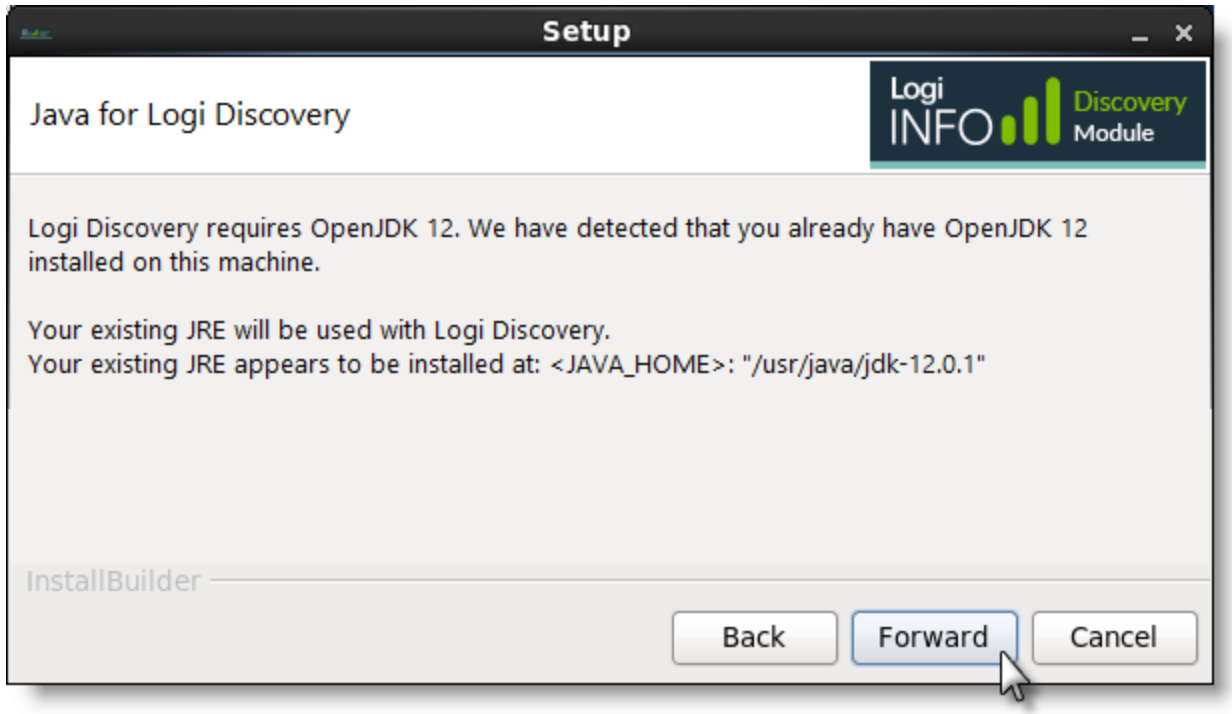
1. Run the installer in the GUI. When the **Welcome Screen** appears, click **Forward**.



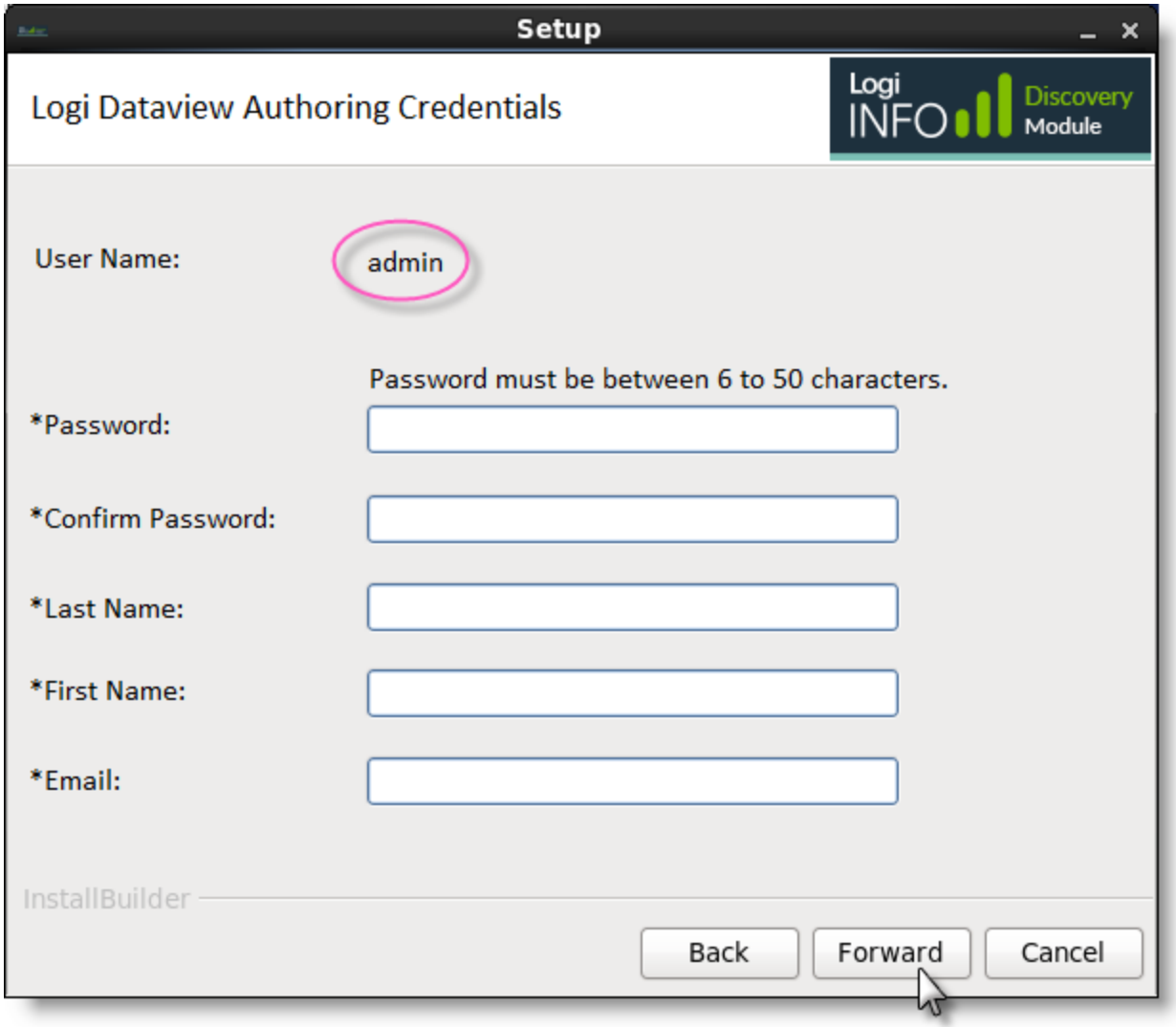
If you see the dialog box shown above, click **No** and the installer will close. Uninstall the existing DM v2.x version and then re-run the DM 3.2 installer.



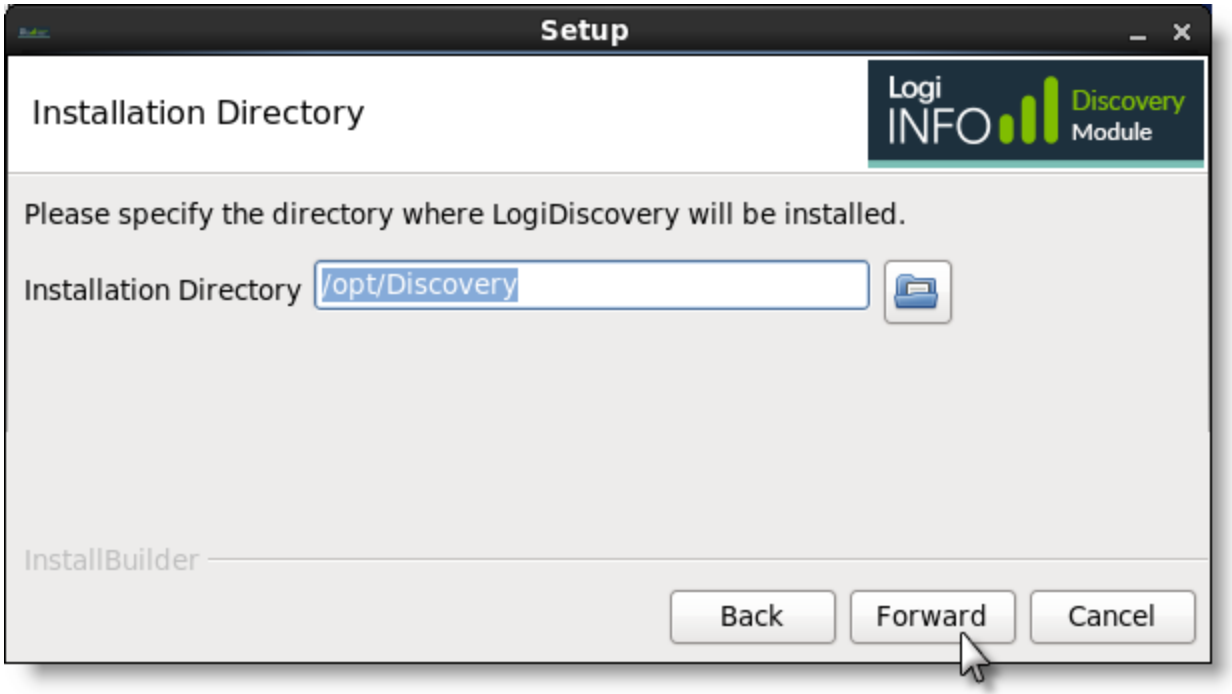
2. **License Agreement**: Select the "I accept the agreement" radio button after reading the license agreement and click **Forward** to continue.



3. **Java Acknowledgement:** Next, if you have OpenJDK installed, you'll see the screen shown above. If you don't, an error message is displayed, along with a link to download OpenJDK. If you have it installed but haven't set the related system variables, an error message is displayed. You *must* install it before proceeding. Click **Forward** to continue.

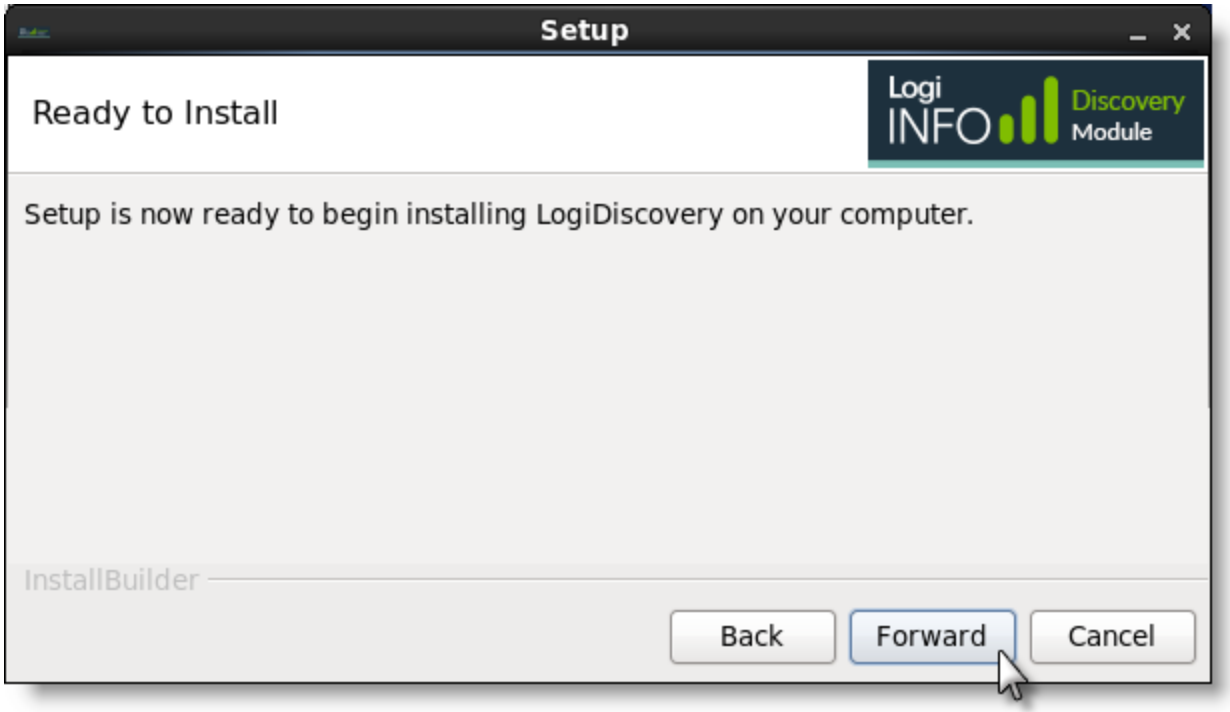


4. **Dataview Authoring Credentials**: (Not shown for upgrades) Provide the required information for the default Dataview Authoring "admin" user, as shown above. Click **Next** to continue.

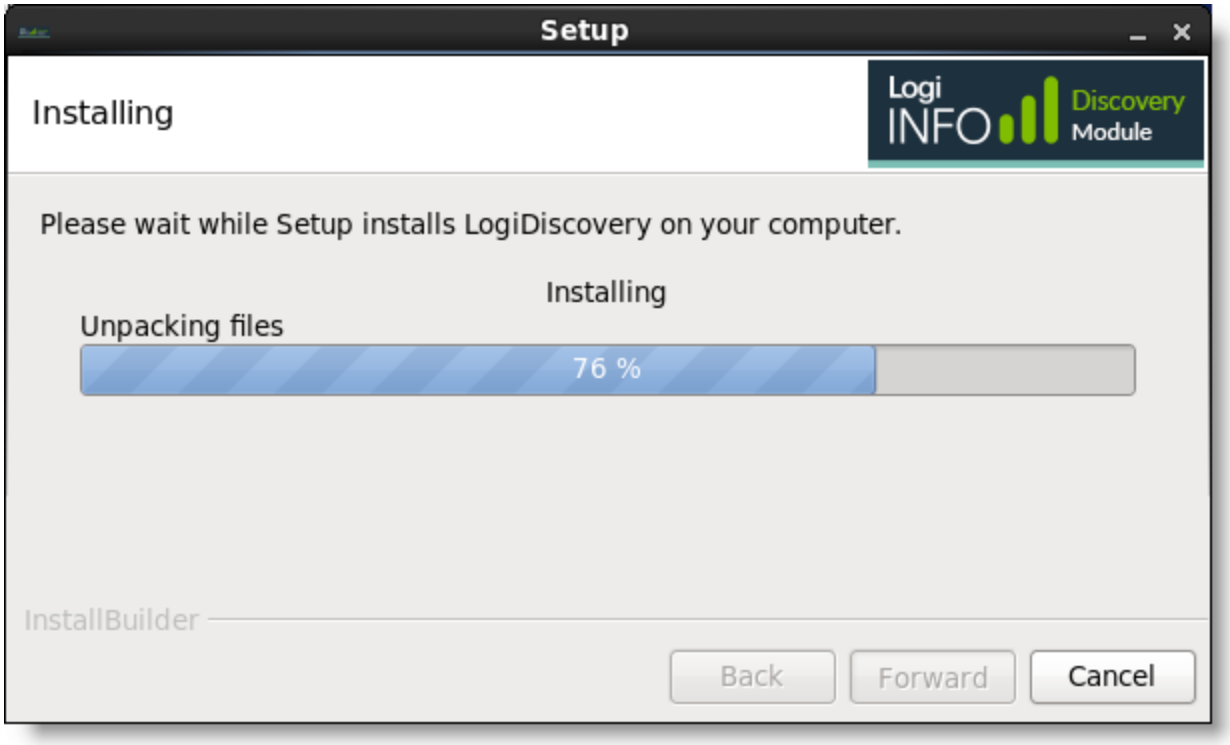


5. **Installation Directory:** Click **Forward** to accept the default installation location and continue. *Optional* - click the Folder icon to specify an alternative installation location if you don't like the default location.

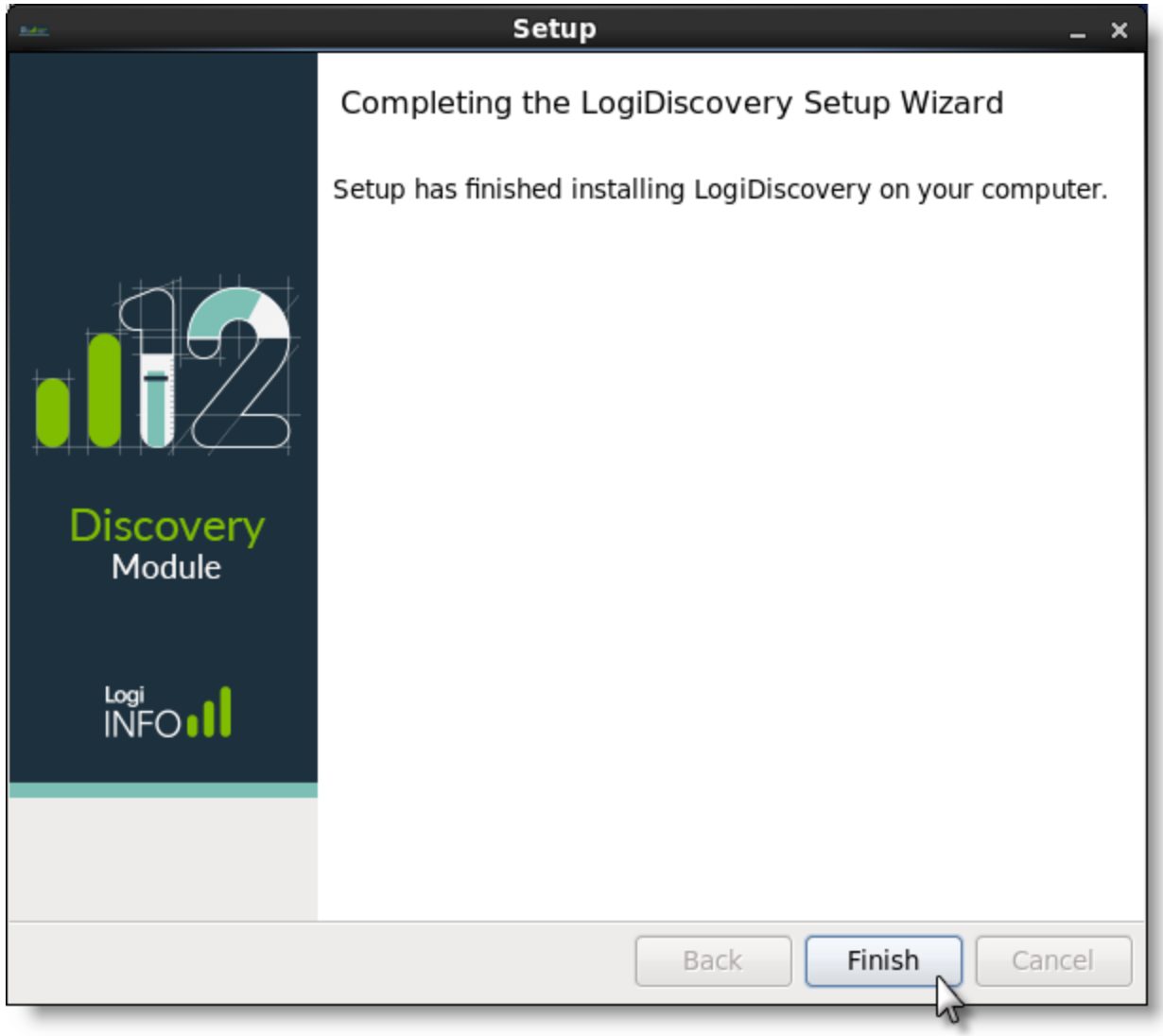
   If you specify a custom installation directory, the folder names in the path must not contain *spaces*. For example, this path is not valid: `/opt/Logi Discovery`. In addition, the *length* of the path must not exceed the OS PATH_MAX length (4,096 by default, usually).



6. **Ready to Install**: Click **Forward**.



7. The physical installation will begin and you'll see a progress indicator for different tasks.



8. **Installation is complete**: Click **Finish** to exit the installer.

Once installation is complete, skip down to one of the following sections about Configuring Services.

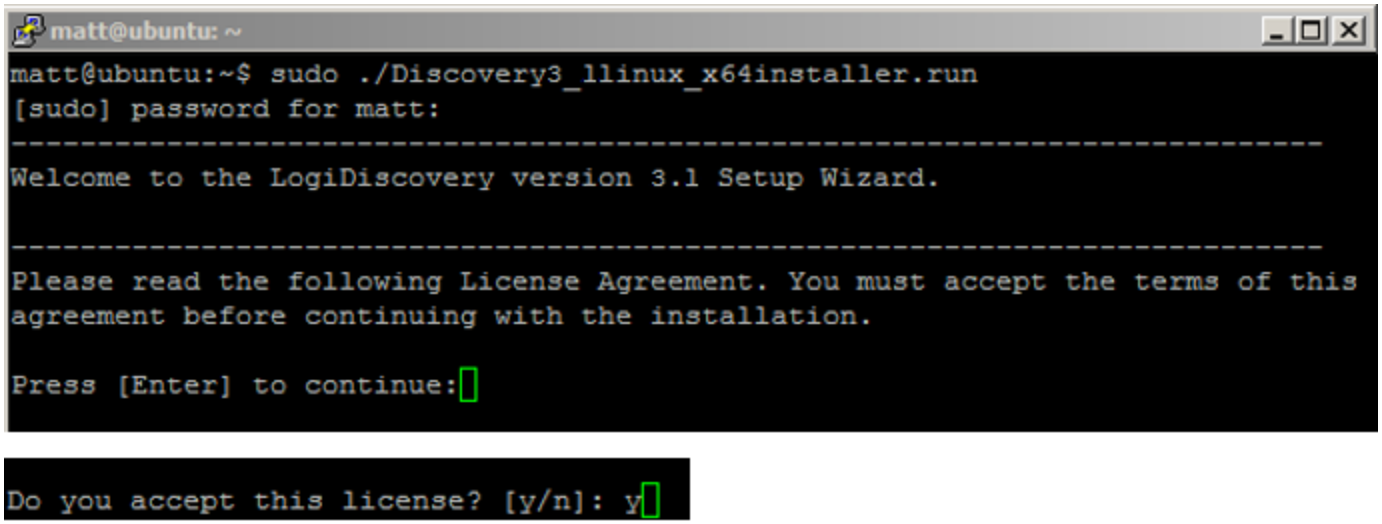
Install the Discovery Module - Linux - Installing the DM Software from a Command-Line

If you do not have a GUI available or prefer to install from a Command Line, use the following instructions:

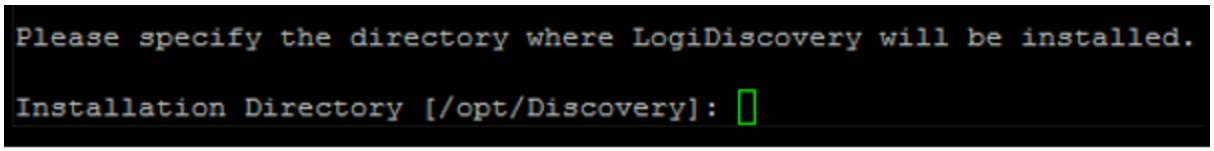
1. Once the file has been downloaded and you've changed its permissions, run the ".run" file, as root.

```
sudo ./Discovery3_2linux_x64installer.run
```

2. When prompted, accept the licenses:

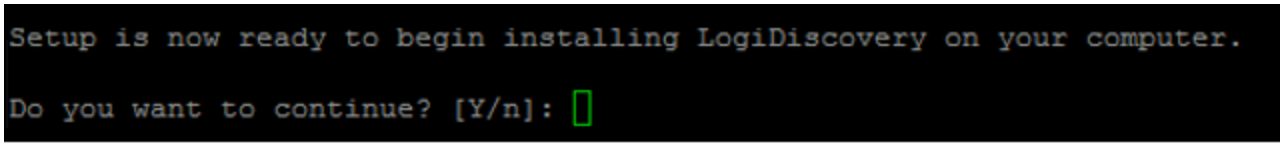


3. Next, you'll be prompted to specify an installation directory - we'll use default location (recommended):



If you specify a custom installation directory, the folder names in the path must not contain *spaces*. For example, this path is not valid: `/opt/Logi Discovery`.

4. You'll be prompted to start the physical installation:



A progress indicator will be updated as the installation proceeds:

And installation is complete. Next you need to configure the related services (daemons).

Install the Discovery Module - Linux - Configuring the Services (with systemctl)

If your OS provides access to systemctl, follow these instructions. We want the Discovery services (daemons) to run as their own user. To do this:

1. Create a new User and Group named "Discovery", as follows:

   ```
   sudo groupadd Discovery
   ```

   ```
   sudo useradd -M -s /bin/nologin -g Discovery -d /opt/Discovery Discovery
   ```

2. Then change the owner of the Discovery directory to the new "Discovery" user:

   ```
   sudo chown Discovery:Discovery /opt/Discovery -R
   ```

3. If you look in Discovery's "bin" directory (`/opt/Discovery/platform/bin`) you'll see two scripts: `logiDataService.sh` and `logiApplicationService.sh`. Both of these scripts need to be run to start the full Discovery server. In this next section, we'll configure a service to start and stop the Discovery server.

   First, we'll configure the application service:

   ```
   sudo nano /etc/systemd/system/LogiApplicationService.service
   ```

Include these items in the `LogiApplicationService.service` file:

```
[Unit]
Description=Logi Application Service

[Service]
Environment=JAVA_HOME=/opt/Discovery/platform/java/jre1.8.0_102
Environment=NODE_HOME=/opt/Discovery/platform/nodeis
Environment=LOGI_HOME=/opt/Discovery
ExecStart=/opt/Discovery/platform/bin/logiApplicationService.sh

User=Discovery
Group=Discovery
```

Next, for the data service:

```
sudo nano /etc/systemd/system/LogiDataService.service
```

Include these items in the `LogiDataService.service` file:

```
[Unit]
Description=Logi Data Service

[Service]
Environment=JAVA_HOME=/opt/Discovery/platform/java/jre1.8.0_102
Environment=NODE_HOME=/opt/Discovery/platform/nodejs
Environment=LOGI_HOME=/opt/Discovery
ExecStart=/opt/Discovery/platform/bin/logiDataService.sh

User=Discovery
Group=Discovery
```

4. Finally, run this command so systemd picks up the changes:

```
sudo systemctl daemon-reload
```

And then start the services:

```
sudo service LogiDataService start
sudo service LogiApplicationService start
```

Now run this test to see if the server responds:

```
curl localhost:3000
```

If you receive a message that says "Cannot GET /" then the installation was successful. Congratulations!

Start Services on Boot

If you'd like to have both services start on boot, do this:

sudo systemctl LogiApplicationService enable
sudo systemctl LogiDataService enable

Install the Discovery Module - Linux - Configuring the Services (without systemctl)

If your OS *does not* provide access to systemctl, follow these instructions. The examples are from Centos 6.9. Start by logging in as root, then download from DevNet and save the file logiDiscovery.sh.

Move the file to `/etc/init.d` and make it executable:

mv logiDiscovery.sh /etc/init.d
chmod +x logiDiscovery.sh

Use a text editor to open the file and edit the DISCOVERYHOME, NODE_HOME, and JAVA_HOME values to match your installation location. For example, these are the values for the default location:

```
DISCOVERYHOME = /opt/Discovery
NODE_HOME = /opt/Discovery/platform/node.js
JAVA_HOME = /opt/Discovery/platform/java/jre1.8.0_102 (JRE directory may vary with different Discovery releases)
```

You should now be able to immediately start the services, from within the `/etc/init.d` directory, with:

./logiDiscovery.sh start

Now run this test to see if the server responds:

curl localhost:3000

If you receive a message that says "Cannot GET /" then the installation was successful. Congratulations!

Start Services on Boot

To ensure that the services will start on their own if the server is restarted, edit the top of the `logiDiscovery.sh` file, changing:

```
#!/bin/bash
#
```

To:

```
#!/bin/bash
#
# Discovery 3.2
#
# chkconfig: - 10 90
#
```

Then, as root, run:

```
chkconfig --add logiDiscovery.sh
chkconfig logiDiscovery on
```

Install the Discovery Module - Linux - Customizing Service Settings

You may wish to customize the Logi services settings which, assuming a default installation location, can be found in these files:

```
<installationFolder>\platform\settings\logiApplicationService.json
<installationFolder>\platform\settings\logiDataService.json
```

The beginning of the `logiApplicationService.json` file looks like this:

```
{
"disableSchemaValidation": false,
"logiApplicationService": {
"description": "Platform logiApplicationService Configuration",
"logLevel": "error",
"corsOriginWhiteList": "*",
"accessTokenGracePeriod": 120,
"system": {
"pollingIntervalSeconds": 15,
"pollingTimeoutMinutes": -1
},
```

For example, the `corsOriginWhiteList` attribute, which is required, specifies which servers the service will respond to, allowing you to restrict access to the service, if desired. The default value, "*", tells the service to respond to *all* servers; to restrict access enter a single IP address or server name (as a string) or a JSON array of them, such as:

```
["http://server1","http://server2","http://192.168.1.1"]
```

Changes to some settings may cause applications to fail. For example,

```
39      "port": 3000,
40      "hostname": null,
41      "proxyUrl": null,
42      "platformBaseUrl": "/api/platform" !!
```

changing the `platformBaseUrl` setting, shown above, may cause integrations with Logi Info applications to fail.

You will need to stop and restart both services to have any changes to the settings file take effect.

Special thanks to contributing writer Matt Grogan.

Installing Discovery Module Instances Manually

The following topics guide you through manually installing single or multiple instances of the **Discovery Module** on a single computer under the Windows or Linux operating systems:

- Manual Installations under Windows
- Manual Installations under Linux

About Manual Installations

Advanced features discussed here work with Logi Info v12.5 and later. Earlier and later Info versions may not support them; consult the Release Notesfor specific details.

The Logi Discovery Module (DM) provides an embedded data discovery experience that works directly with relational data sources using Logi data services. It does this by making several special elements available in Logi Studio. These include the **Thinkspace** element, which provides a highly-interactive analysis experience. In its interface, best-fit visualization suggestions are automatically referred to the user by a built-in "recommendation engine" and data is organized using intelligent technology that clusters data to make it more easily consumable by users.

A "manual" installation does not use the usual installer program. Instead, you begin with a .zip file and manually accomplish the steps that are normally completed by the installer. Why would you do this? You might manually install a *single* instance of the DM in order to gain a better understanding of how it works, or to customize the installation.

A manual installation is also necessary for situations like a "multi-instance installation". That's a scenario where you may need to have multiple, separate DM installations on the same server. You may want to do this for a variety of reasons, including ensuring that multiple "tenants" are completely isolated from one another. The standard DM installers, however, are not engineered for

this scenario. When they run and find an existing installation, they assume that you're trying to apply an upgrade and respond accordingly. A manual installation lets you install as many instances as you need.

Requirements

Generally, we recommend that you use the standard installers for your first installation instance, then follow the instructions provided here for subsequent instances. However, as mentioned earlier, you can manually install a single or first instance, if you prefer.

Information about requirements, licensing, and what get's installed is provided in the standard "Install the Discovery Module - Windows" on page 224and "Install the Discovery Module - Linux" on page 241 installation topics and is not repeated here.

You have the choice of whether or not to use a *separate* instance of the OpenJDK 12 and Node.js 8.11.4 for each DM instance. These should be installed before you begin the instructions provided below. If Node.js is installed using the OS package manager, its files may be distributed across multiple folders, which is unacceptable for Discovery. The correct process is to download Node.js and keep in contained in one folder, setting the $NODE_HOME variable to that folder, e.g. `/home/someUser/node-v8.11.4-linux-x64`.

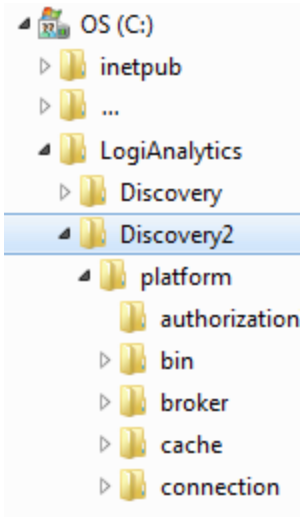
The Home folders and the `logiDiscovery.bat` (Windows) or `logiDiscovery.sh` (Linux) should be named differently for each instance.

You may not install directly to the root folder of a drive, i.e. `C:\` and the folder names in the path must *not* contain *spaces*. For example, this path is not valid: `C:\Program Files\Discovery`. In addition, the *length* of the path must not exceed 260 characters.

Manual Installations Under Windows

To install the DM:

1. Ensure that you're logged into your computer as an administrator.
2. Download the Discovery .zip file using the link provided by Logi staff and save it to a convenient location.
3. Create a new folder under `C:\LogiAnalytics` for the new instance and give it a name that's relevant to you. The default for a single installation would be: `C:\LogiAnalytics\Discovery` and for a multi-instance installation we'll use, `C:\LogiAnalytics\Discovery2` in these instructions.

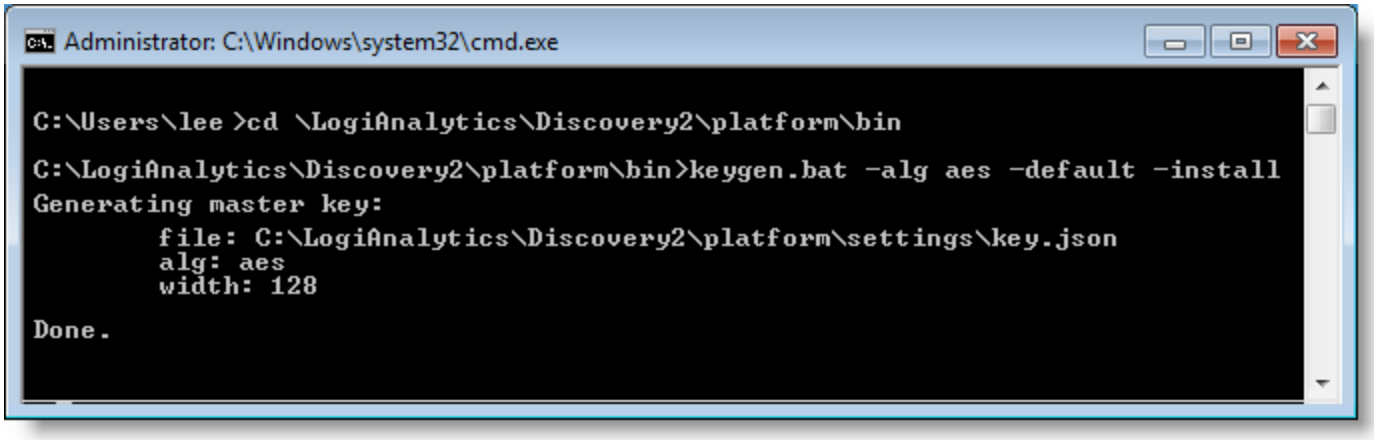


4. Extract the downloaded .zip file contents into that new folder - it requires 600+ MB of space and will take several minutes. The resulting folder structure should be similar to that shown above. To conserve space when referring to the installation folder, in the following steps we'll shorten "C:\LogiAnalytics\Discovery2" to just "Discovery2".

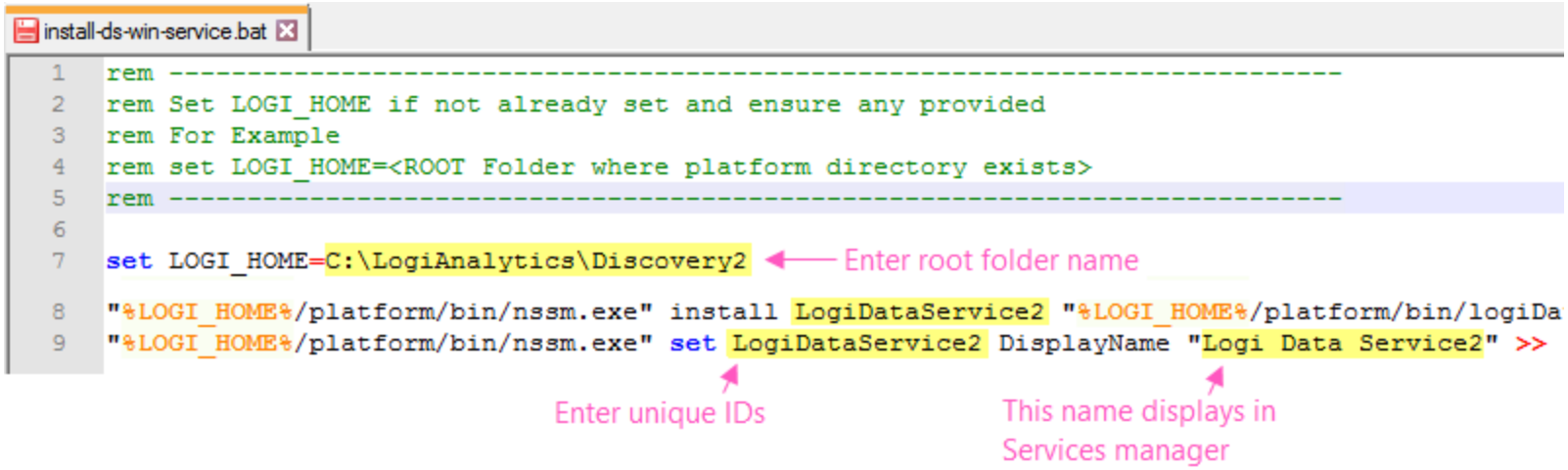
logiconfig.bat

```
rem ------------------------------------------------------------------------
rem Set LOGI_HOME, JAVA_HOME and NODE_HOME if not already set and ensure any provided
rem settings are valid and consistent with the selected start-up options.
rem For Example
rem set LOGI_HOME=<ROOT Folder where platform directory exists>
rem set JAVA_HOME=%LOGI_HOME%\platform\java\jdk-12.0.1
rem set NODE_HOME=%LOGI_HOME%\platform\nodejs
rem ------------------------------------------------------------------------
set LOGI_HOME=C:\LogiAnalytics\Discovery2
set JAVA_HOME=C:\LogiAnalytics\Discovery/platform/java/jdk-12.0.1
set NODE_HOME=C:\LogiAnalytics\Discovery/platform/nodejs
```

5. Once the extraction completes, use a text editor like NotePad++ to open the `Discovery2\platform\bin\logiconfig.bat` file and edit the three environment variable paths, as shown above. If you want to use the JRE and Node.js installed for your first instance by the standard installer, edit their paths here to point to their folders, as shown above. Or, if you want separate instances of them, too, you can download them, install them, and edit the paths here appropriately.



6. Generate Master Key data by opening a Command Window, navigating to the `Discovery2\platform\bin` folder, and executing `keygen.bat -alg aes -default -install`. A successful execution will look like the image shown above. You'll use the Command Window again later, so leave it open.



7. If you're installing a single instance and wish to use default configuration values, skip ahead to Step #9. Otherwise, open the `Discovery2\platform\bin\install-ds-win-service.bat` file and edit the lines shown above.

install-app-win-service.bat

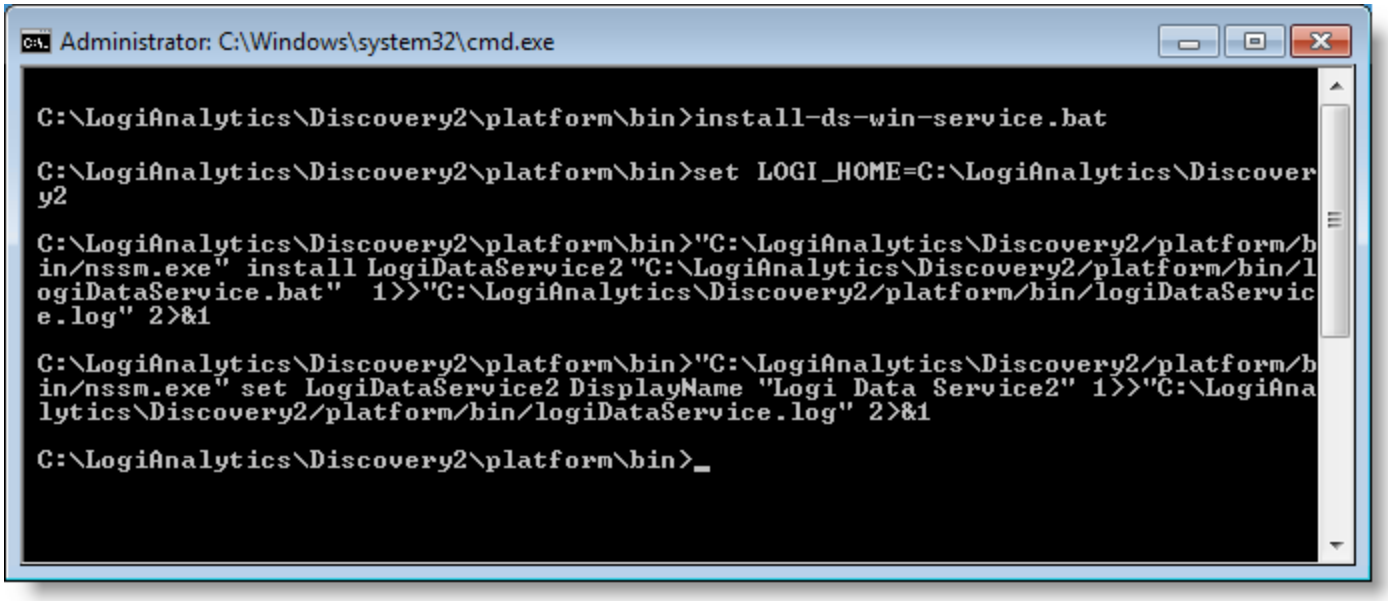
```
1  rem ------------------------------------------------------------------
2  rem Set LOGI_HOME if not already set and ensure any provided
3  rem For Example
4  rem set LOGI_HOME=<ROOT Folder where platform directory exists>
5  rem ------------------------------------------------------------------
6
7  set LOGI HOME=C:\LogiAnalytics\Discovery2
8  "%LOGI_HOME%/platform/bin/nssm.exe" install LogiApplicationService2 "%LOGI_HOME%/platform/bin/logiApplica
9  "%LOGI_HOME%/platform/bin/nssm.exe" set LogiApplicationService2 DisplayName "Logi Application Service2"
```

Enter root folder name

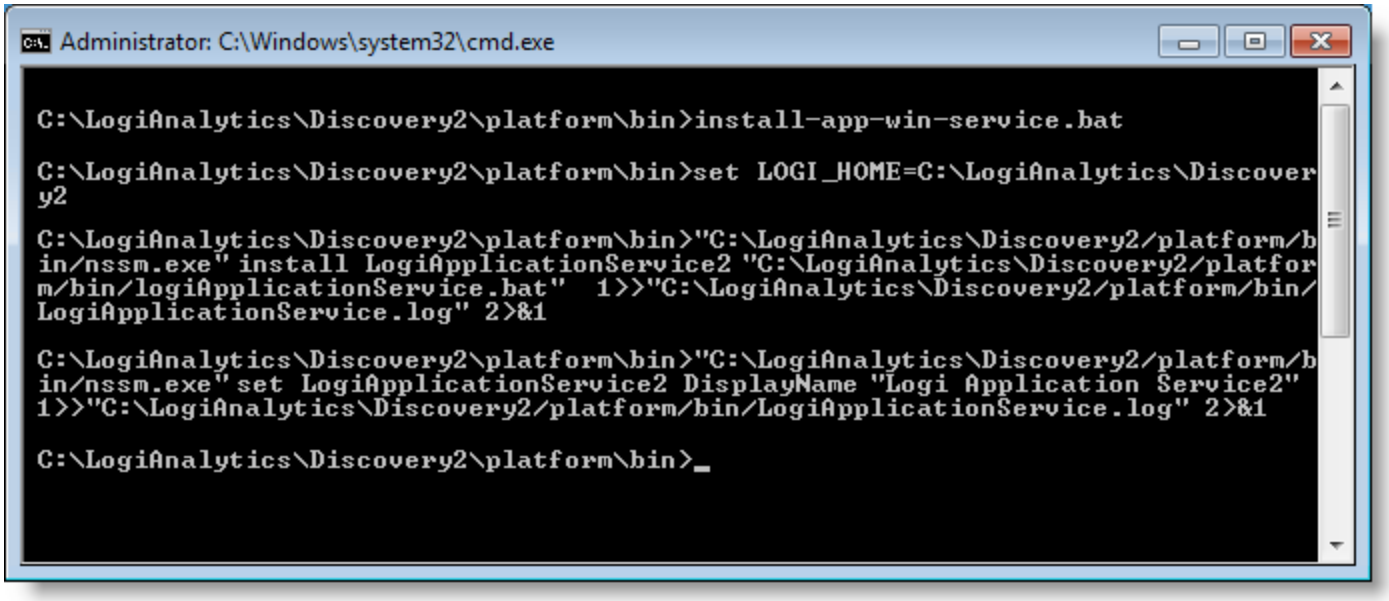
Enter unique IDs

This name displays in Services manager

8. If you're installing a single instance and wish to use default configuration values, skip ahead to Step #9. Otherwise repeat the process with the `Discovery2\platform\bin\install-app-win-service.bat` file, as shown above.



9. Return to your Command Window (its current folder should still be `Discovery2\platform\bin`) and execute `install-ds-win-service.bat`, as shown above. For clarity, the image above has been edited to remove some remarks that you may see.



10. Execute `install-app-win-service.bat`, as shown above. For clarity, the image above has been edited to remove some remarks that you may see. Leave the Command Window open for later use.
11. An installation of the DM is configured to use a number of communication ports. If you're installing a single instance and would like to use the default settings, skip ahead to the next step. These ports must be unique for each DM instance and so you'll need to configure them if you're installing multiple instances. These ports must be above 5000 and, obviously, can't already be in use.

    Use your text editor to change the instance name and the port numbers in the following configuration files, on or near the specified line numbers. You can use the `netstat -a -b` command in your Command Window to see what ports are already being used. Some of the values shown below have been truncated and ellipses added to conserve space. *The example values shown are just suggestions and may not work on your system.*

Discovery2\platform\settings\logiDataService.json:

Line #	Default Value	Example New Value
3	"instanceName": "NGP_1",	"instanceName": "NGP_2",
77	"uri": "tcp://localhost:61616?..."	"uri": "tcp://localhost:51616?..."
81	"uri": "stomp://localhost:61613"	"uri": "stomp://localhost:51613"
118	"url": "jdbc:h2:tcp://localhost:9092/LogiDB",	"url": "jdbc:h2:tcp://localhost:5092/LogiDB",
166	"port": 7654,	"port": 5654,
176	"port": 7654,	"port": 5654,

Discovery2\platform\settings\logiApplicationService.json:

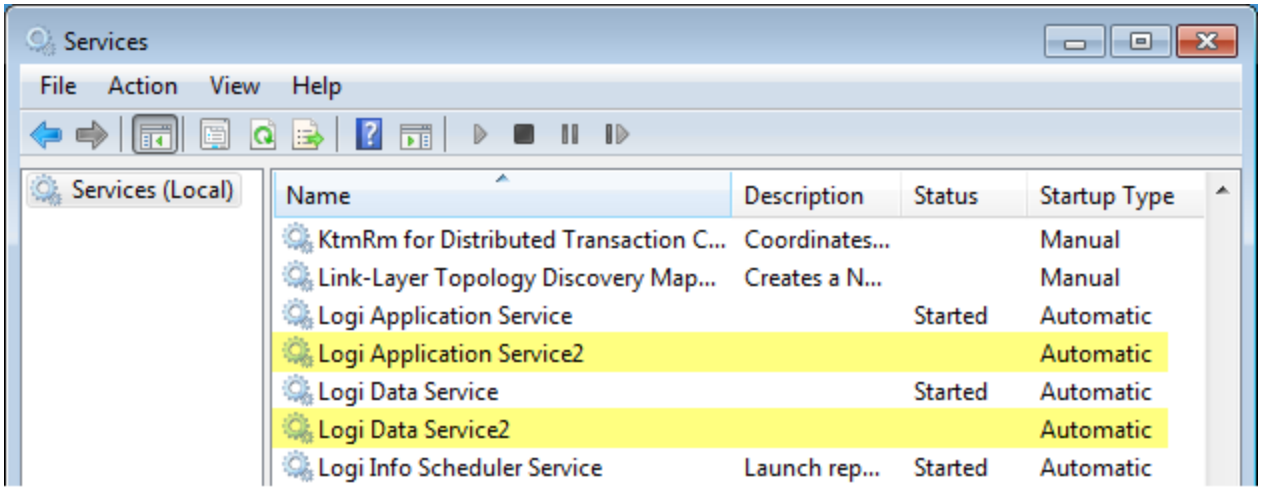
Line #	Default Value	Example New Value
13	"uri": "stomp://localhost:61613"	"uri": "stomp://localhost:51613"
26	"port": 8443,	"port": 5443,
40	"port": 3000,	"port": 5000,

Discovery2\platform\bin\clientSecret.js:

Line #	Default Value	Example New Value
40	"port": 3000,	"port": 5000,

The port values on or near Line #40 in the last two files must be the same, as shown above.

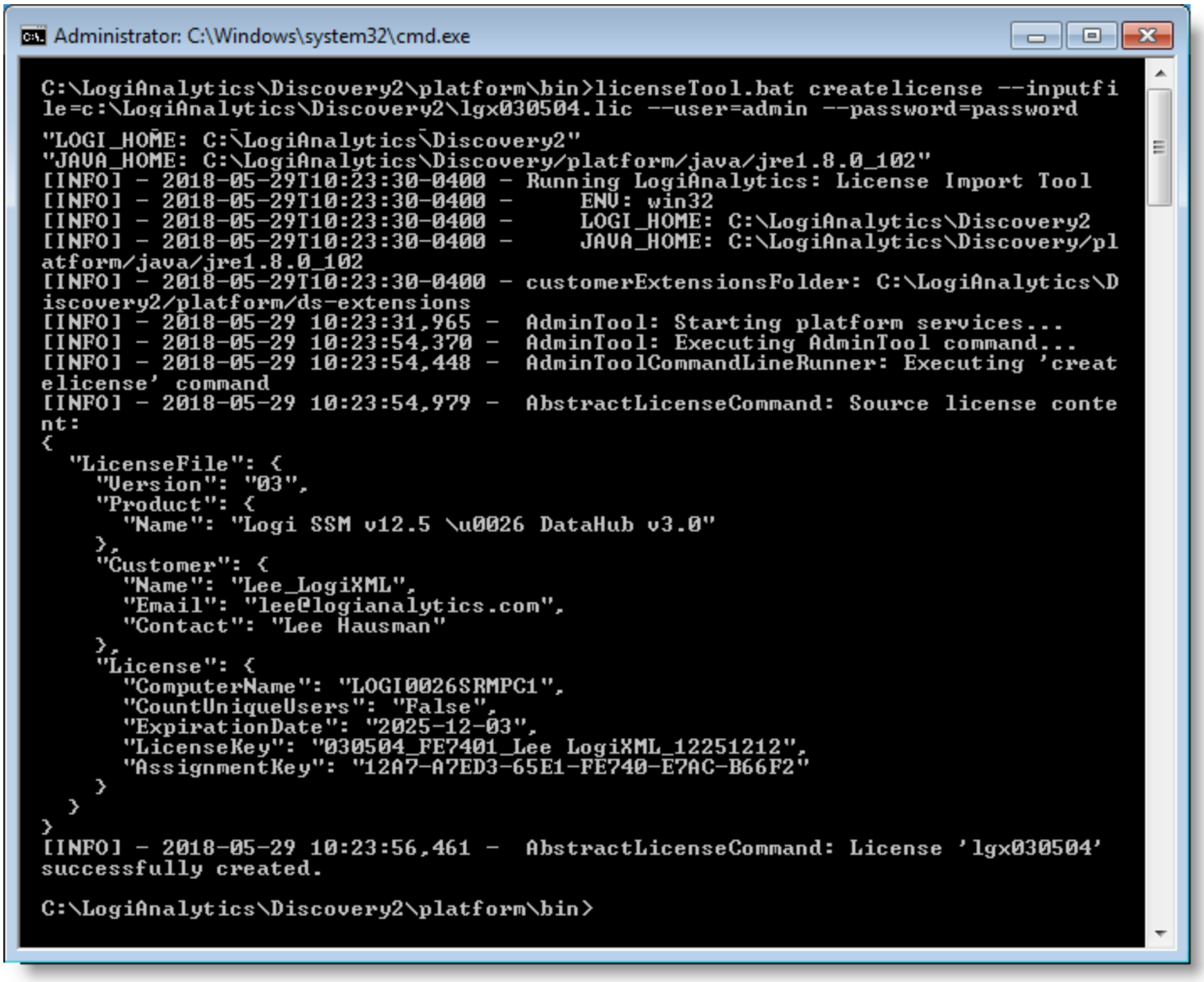
If you make any changes to these settings in the future, remember that you'll need to stop and restart the services for the changes to take effect.



12. Open Windows' Administrative Tools → Services management tool and start the **Logi Data Service2** and then the **Logi Application Service2** services, as shown above (the service names will match the Display Names you entered earlier in Steps #7 and #8, or if you're installing a single instance they'll be "Logi Applications Service" and "Logi Data Service"). Ensure that the services show a *Started* status and all is well, then *stop* both services. Leave the Services management tool open.

13. Next, you need to import the product license. For a single instance installation, download a Logi product license file, which is similar to `lgx030504.lic`, from DevNet and save it to `C:\LogiAnalytics\Discovery`.

    For a multi-instance installation, copy the Logi product license file, which is similar to `lgx030504.lic`, from your first DM installation (or download it from DevNet) and save it to `C:\LogiAnalytics\Discovery2`.



Stop the Logi services and import the license by running the following in your Command Window:

```
licenseTool.bat createlicense --inputfile=c:\LogiAnalytics\Discovery2\lgx030504.lic --user=admin --pass-
word=password
```

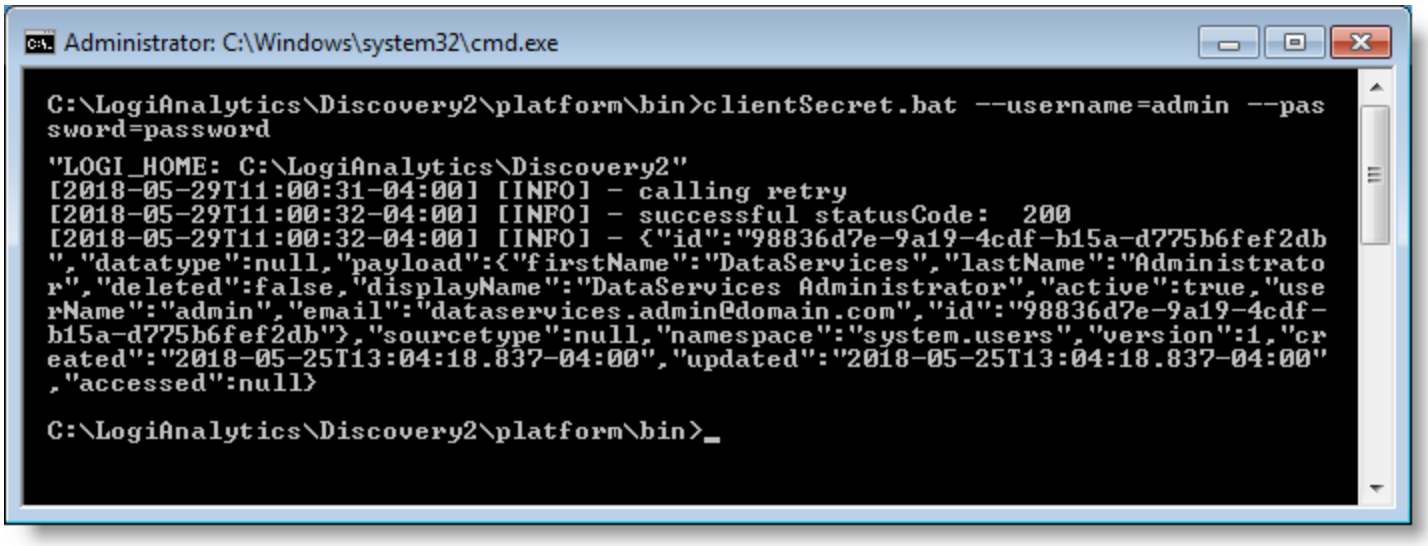
providing the correct path and license file name. The user and password values are literally "admin" and "password" - *do not* substitute your own values here. Leave the Command Window open.

14. After the license file has been imported, for a single installation, return to the Services management tool and restart the Logi Data Service and then the Logi Application Service services. Leave the Services management tool open again.

    For a multi-instance installation, return to the Services management tool and restart the Logi Data Service2 and then the Logi Application Service2 services. Leave the Services management tool open again.

15. In your Command Window, update the "Client Secret" by running:

```
clientSecret.bat --username=admin --password=password
```

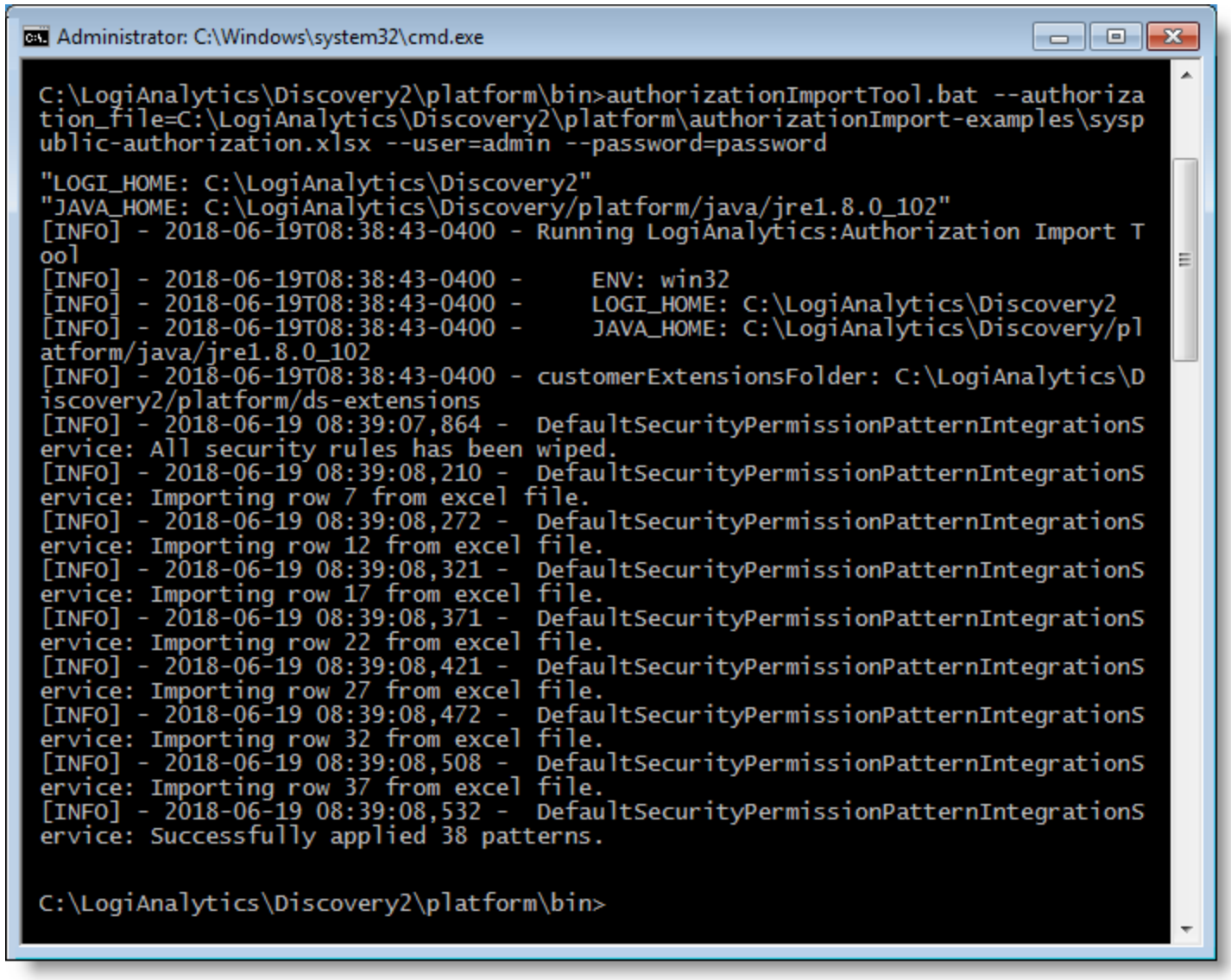


as shown above. Again, use the literal "admin" and "password" argument values.

16. By default, only an Admin user is allowed to login and create Dataviews, Widgets, and Dashboards. If that's satisfactory, skip ahead to the next step. Otherwise, to allow *all* other users to also login and create these objects, while still in the `\platform\bin` folder, run the following two commands from your Command Window.

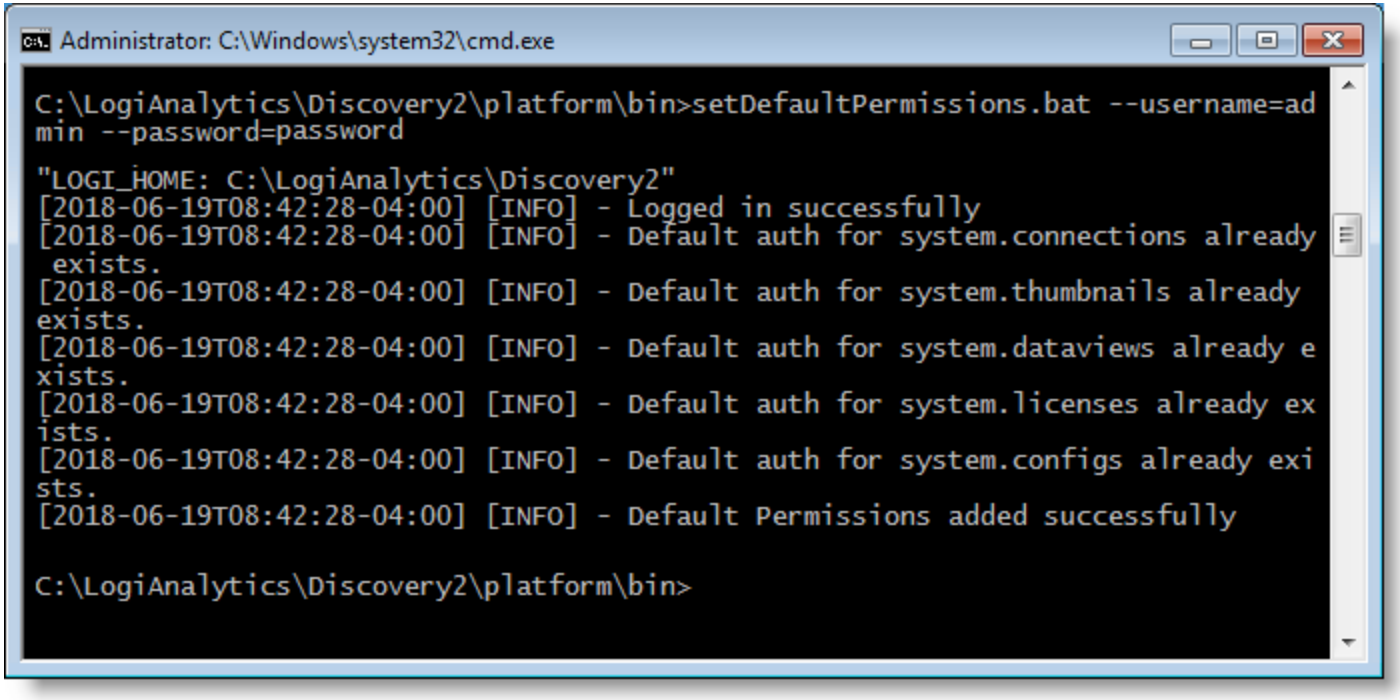
    The first command reads an Excel spreadsheet file containing the object authorizations. We've provided an example Excel file with the authorizations for "syspublic" or all users. Use the fully-qualified path and filename for that file as an argument in the first command:

    ```
    authorizationImportTool.bat --authorization_file=C:\LogiAnalytics\Discovery2\platform\authorizationImport-examples\syspublic-authorization.xlsx --user=admin --password=password
    ```



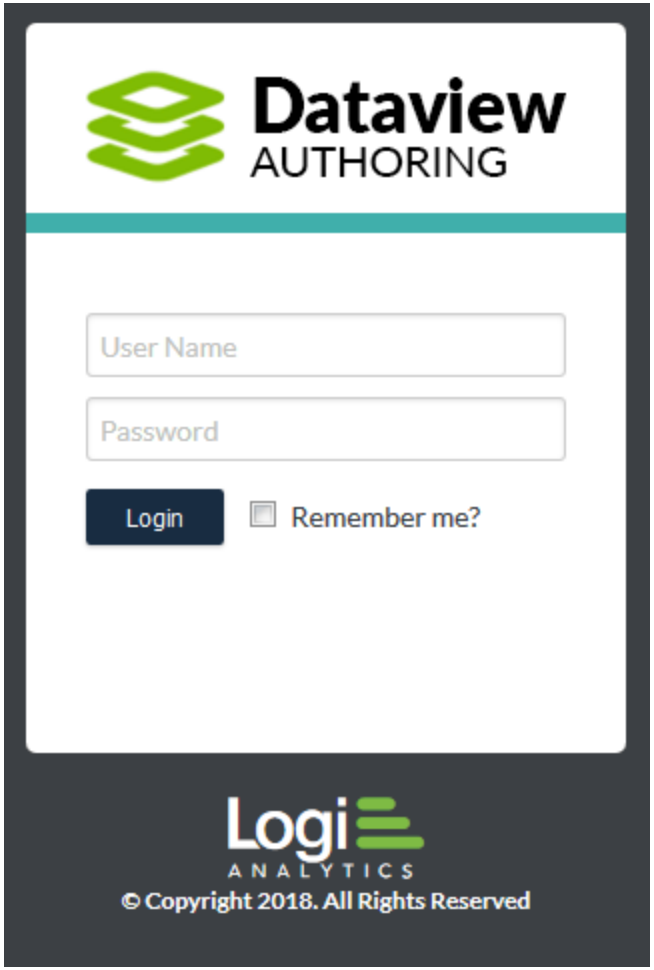
and the output is shown above. Then run the second command:

```
setDefaultPermissions.bat --username=admin --password=password
```



as shown above. You can close the Command Window now.

17. In the Services management tool, stop and restart the two Logi services again.



18. In your browser, navigate to localhost:5000/Datahub (using whatever port value you entered in Line #40 in Step #11 - 3000 is the single instance default) and then login as "admin" with "password".

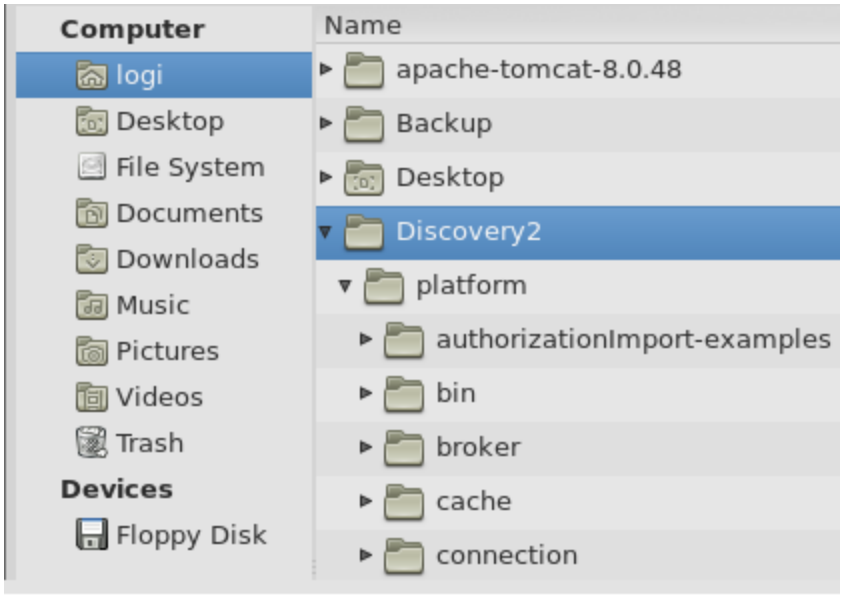
That completes the installation and configuration. Repeat the process for each new instance you add.

Manual Installations under Linux

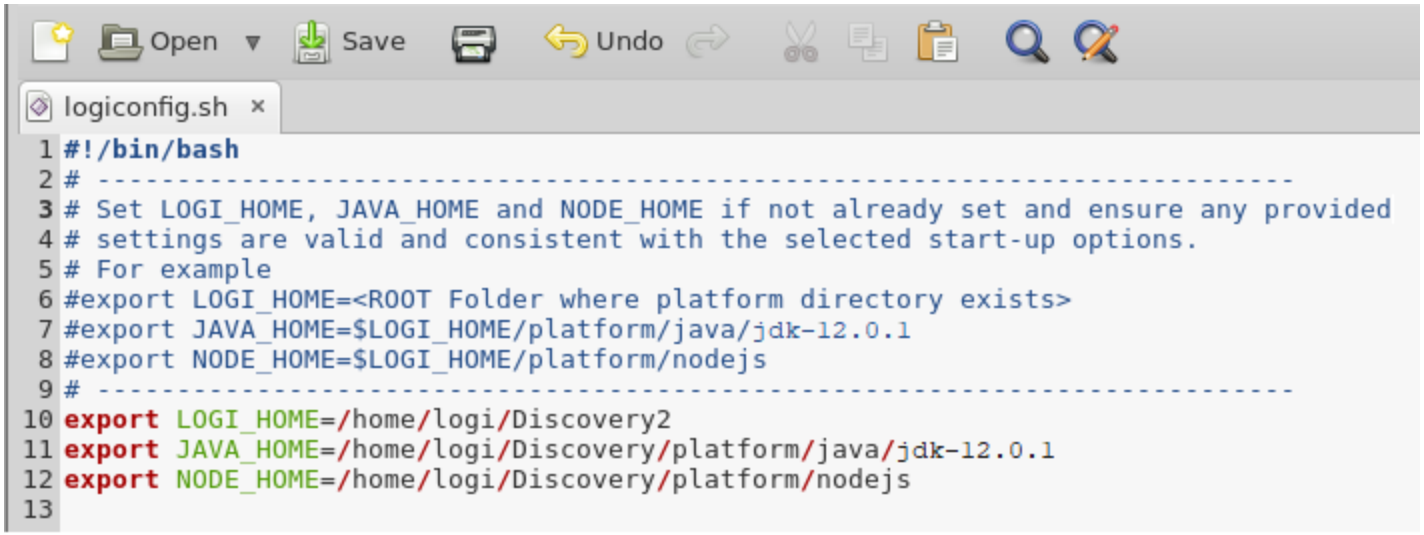
Many of the commands used here are the same as those shown in the "Manual Installations Under Windows" on page 266 topic, so screenshots of the resulting terminal or command line output will not be duplicated in this topic. The examples shown are based on Centos 7, you may need to adjust them for your specific Linux OS variant. File and folder names and command line arguments are, of course, case-sensitive.

To install the DM:

1. Ensure that you're logged into your computer as an administrator.
2. Download the Discovery .zip file using the link provided by Logi staff and save it to a convenient location.
3. Create a new folder for the new instance and give it a name that's relevant to you. Assuming you logged in as user "logi", the default for a single installation would be: `~/Discovery` and for a multi-instance installation we'll use `~/Discovery2` in these instructions.



4. Extract the downloaded .zip file contents into that new folder - it requires 600+ MB of space and will take several minutes. The resulting folder structure should be similar to that shown above.

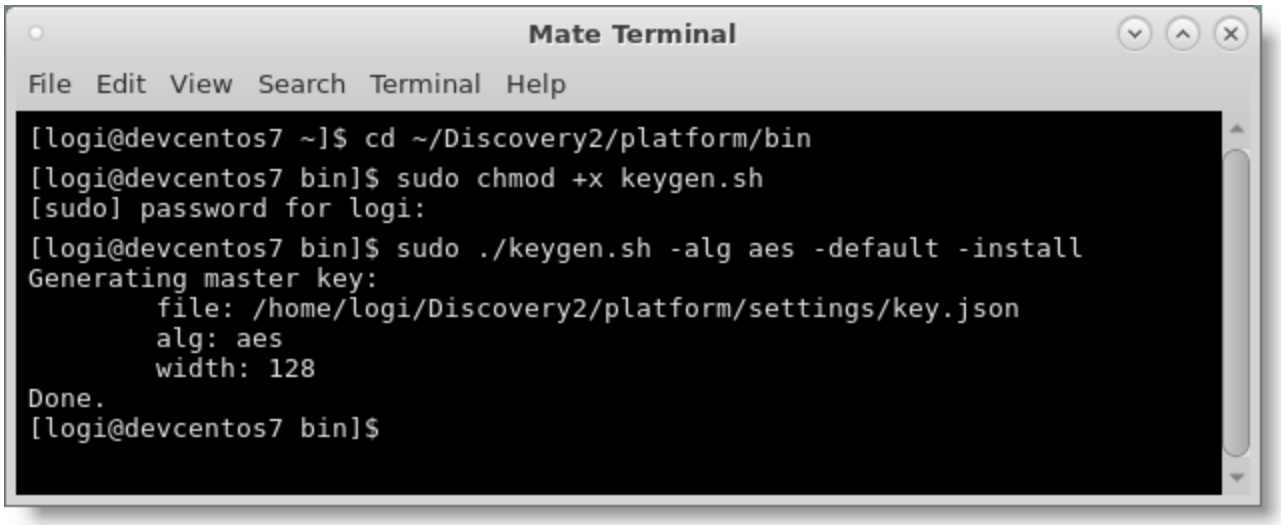


5. Once the extraction completes, use a text editor like **pluma** or **nano** to open the `~/Discovery.. or ~/Discovery2/platform/bin/logiconfig.sh` file and edit the three environment variable paths, as shown above. LOGI_HOME will be the `/Discovery` folder for a single installation or `/Discovery2` for a multi-instance installation.

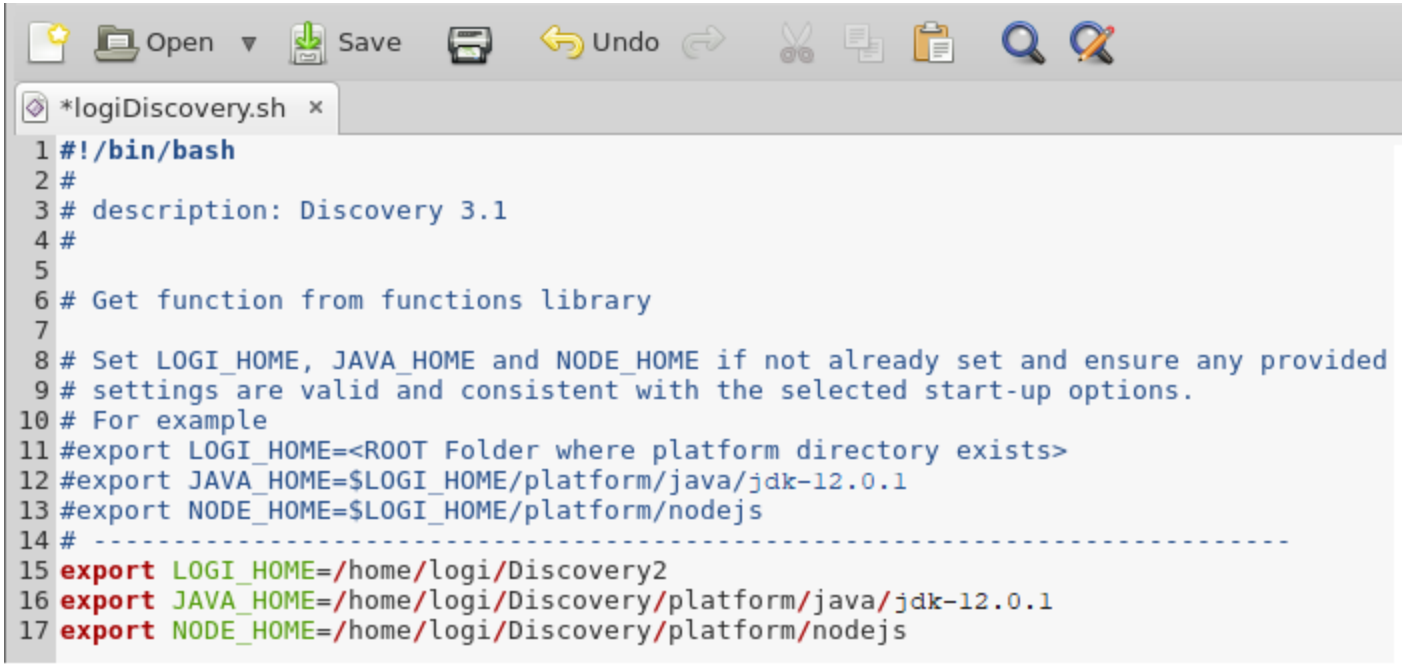
   If you want to use the JRE and Node.js installed for your first instance by the standard installer, edit their paths here to point to their folders, as shown above. Or, if you want separate instances of them, too, you can download them, install them, and edit the paths here appropriately (see the Requirements section in "Installing Discovery Module Instances Manually" on page 264for version information).

6. Generate Master Key data by opening a terminal or command line window, navigating to the `~/Discovery2/platform/bin` folder, and executing:

   ```
   sudo chmod +x keygen.sh
   sudo ./keygen.sh -alg aes -default -install
   ```



A successful execution will look like the image shown above. You'll use the command line window again later, so leave it open.



7. Open the `~/Discovery2/platform/bin/logiDiscovery.sh` file and edit the environment variable paths, as you did in Step #5.
8. Save and then copy the edited `logiDiscovery.sh` file to the `/etc/init.d` folder, using the appropriate source path:

   ```
   sudo cp ~/Discovery2/platform/bin/logiDiscovery.sh /etc/init.d/logiDiscovery.sh
   ```
9. An installation of the DM is configured to use a number of communication ports. If you're installing a single instance and would like to use the default settings, skip ahead to the next step. These ports must be unique for each DM instance and so you'll need to configure them. These ports must be above 5000 and, obviously, can't already be in use.

   Use the following commands and your text editor to change the instance name and the port numbers in the following configuration files, on or near the specified line numbers. You can use the `netstat -an`command in your command line window to see what ports are already being used. Some of the values shown below have been truncated and ellipses added to

conserve space. *The example values shown are just suggestions and may not work on your system.*

```
cd ~/Discovery2/platform/settings/
sudo chmod 755 logiDataService.json
sudo pluma logiDataService.json
```

Line #	Default Value	Example New Value
3	`"instanceName": "NGP_1",`	`"instanceName": "NGP_2",`
77	`"uri": "tcp://localhost:61616?..."`	`"uri": "tcp://localhost:51616?..."`
81	`"uri": "stomp://localhost:61613"`	`"uri": "stomp://localhost:51613"`
118	`"url": "jdbc:h2:tcp://localhost:9092/LogiDB",`	`"url": "jdbc:h2:tcp://localhost:5092/LogiDB",`
166	`"port": 7654,`	`"port": 5654,`
176	`"port": 7654,`	`"port": 5654,`

```
sudo chmod 755 logiApplicationService.json
sudo pluma logiApplicationService.json
```

Line #	Default Value	Example New Value
13	`"uri": "stomp://localhost:61613"`	`"uri": "stomp://localhost:51613"`
26	`"port": 8443,`	`"port": 5443,`
40	`"port": 3000,`	`"port": 5000,`

```
cd ../bin
sudo chmod 755 clientSecret.js
sudo pluma clientSecret.js
```

Line #	Default Value	Example New Value
40	`"port": 3000,`	`"port": 5000,`

The port values on Line #40 in the last two files must be the same. If you make changes to these settings in the future, you'll need to stop and restart the services for the changes to take effect.

10. Still in `~/Discovery.. or ~/Discovery2/platform/bin`, grant full permissions to the `logiApplicationService.sh` and `logiDataService.sh` files, using these commands:

```
sudo chmod 755 logiApplicationService.sh
sudo chmod 755 logiDataService.sh
```

11. Open and run the Logi services using the following commands:

```
cd /etc/init.d
sudo chmod +x logiDiscovery.sh
sudo ./logiDiscovery.sh start
```

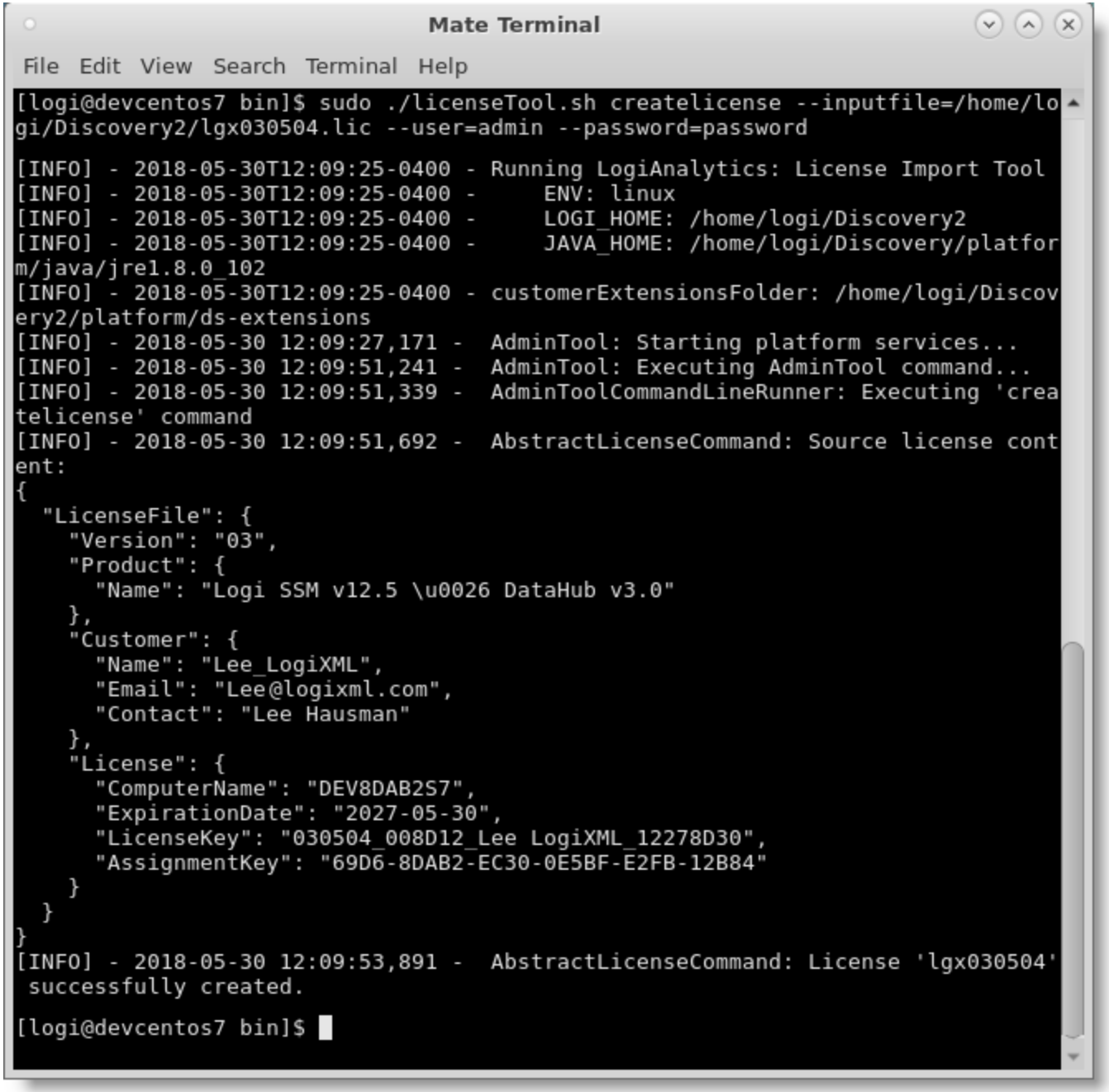
12. Next, you need to import the product license. For a single instance installation, download a Logi product license file, which is similar to `lgx030504.lic`, from DevNet and save it to `~/Discovery`.

    For a multi-instance installation, copy the Logi product license file, which is similar to `lgx030504.lic`, from your first

DMinstallation (or download it from DevNet) and save it to `~/Discovery2`.

Stop the services and import the license using the following commands, providing the correct path and license file name.:

```
sudo ./logiDiscovery.sh stop
cd ~/Discovery2/platform/bin
sudo chmod +x licenseTool.sh
sudo ./licenseTool.bat createlicense --inputfile=/home/logi/Discovery2/lgx030504.lic --user=admin --pass-
word=password
```


Mate Terminal
File Edit View Search Terminal Help
[logi@devcentos7 bin]$ sudo ./licenseTool.sh createlicense --inputfile=/home/logi/Discovery2/lgx030504.lic --user=admin --password=password
[INFO] - 2018-05-30T12:09:25-0400 - Running LogiAnalytics: License Import Tool
[INFO] - 2018-05-30T12:09:25-0400 - ENV: linux
[INFO] - 2018-05-30T12:09:25-0400 - LOGI_HOME: /home/logi/Discovery2
[INFO] - 2018-05-30T12:09:25-0400 - JAVA_HOME: /home/logi/Discovery/platform/java/jre1.8.0_102
[INFO] - 2018-05-30T12:09:25-0400 - customerExtensionsFolder: /home/logi/Discovery2/platform/ds-extensions
[INFO] - 2018-05-30 12:09:27,171 - AdminTool: Starting platform services...
[INFO] - 2018-05-30 12:09:51,241 - AdminTool: Executing AdminTool command...
[INFO] - 2018-05-30 12:09:51,339 - AdminToolCommandLineRunner: Executing 'createlicense' command
[INFO] - 2018-05-30 12:09:51,692 - AbstractLicenseCommand: Source license content:
{
"LicenseFile": {
"Version": "03",
"Product": {
"Name": "Logi SSM v12.5 \u0026 DataHub v3.0"
},
"Customer": {
"Name": "Lee_LogiXML",
"Email": "Lee@logixml.com",
"Contact": "Lee Hausman"
},
"License": {
"ComputerName": "DEV8DAB2S7",
"ExpirationDate": "2027-05-30",
"LicenseKey": "030504_008D12_Lee LogiXML_12278D30",
"AssignmentKey": "69D6-8DAB2-EC30-0E5BF-E2FB-12B84"
}
}
}
[INFO] - 2018-05-30 12:09:53,891 - AbstractLicenseCommand: License 'lgx030504' successfully created.
[logi@devcentos7 bin]$

The user and password values are literally "admin" and "password" - *do not* substitute your own values here.

13. Restart the Logi services, using the following commands:

```
cd /etc/init.d
sudo ./logiDiscovery.sh start
```

14. Update the "Client Secret", using the following commands and providing the correct path:

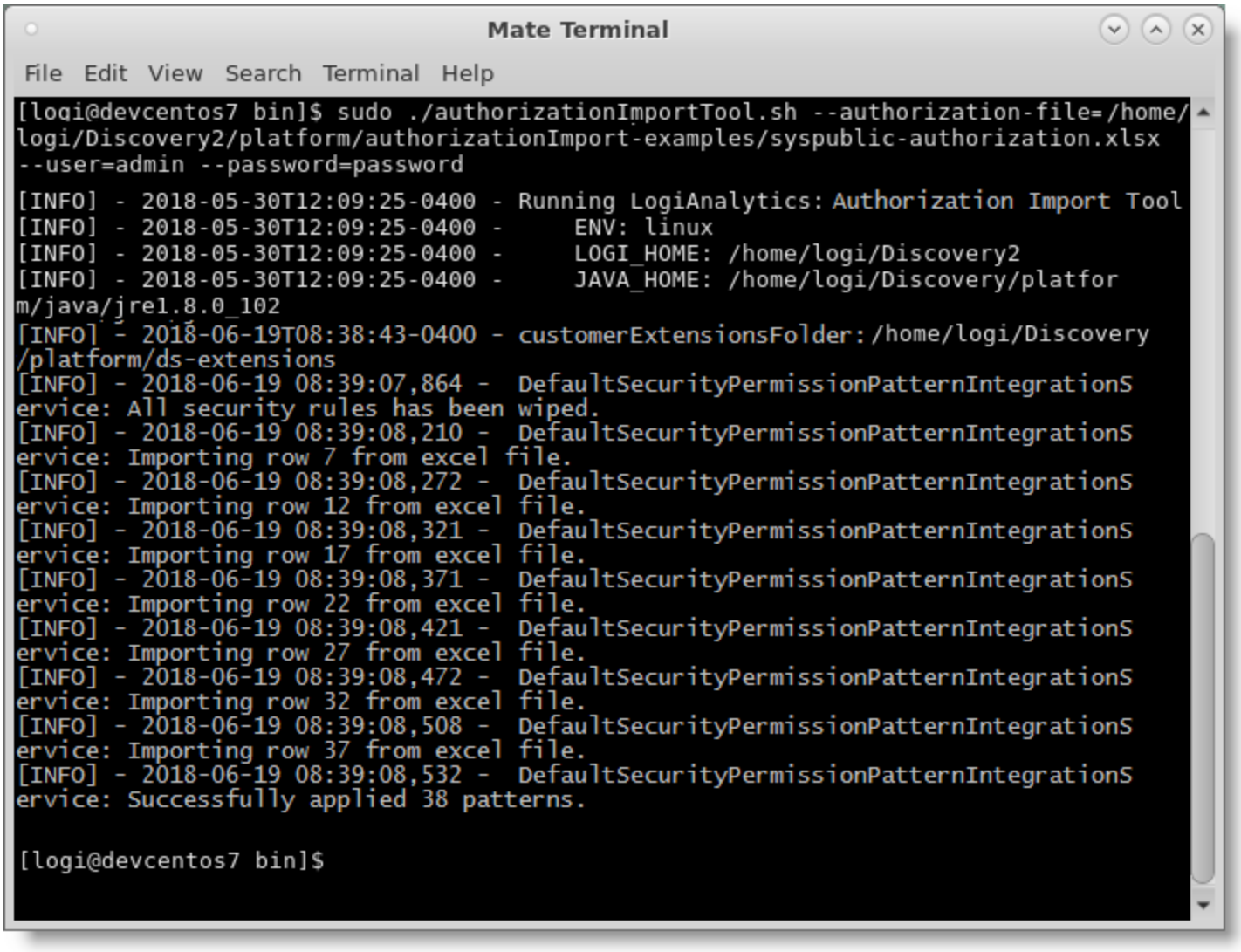
```
cd ~/Discovery2/platform/bin
sudo chmod +x clientSecret.sh
sudo ./clientSecret.sh --username=admin --password=password
```

as shown above. Again, use the literal "admin" and "password" argument values.

15. By default, only an Admin user is allowed to login and create Dataviews, Widgets, and Dashboards. If that's satisfactory, skip ahead to the next step. Otherwise, to allow *all* other users to also login and create these objects, while still in the `/platform/bin` folder, run the following two commands from your Command Window.

The first command reads an Excel spreadsheet file containing the object authorizations. We've provided an example Excel file with the authorizations for "syspublic" or all users. Use the correct, fully-qualified path and filename for that file as an argument in the command:

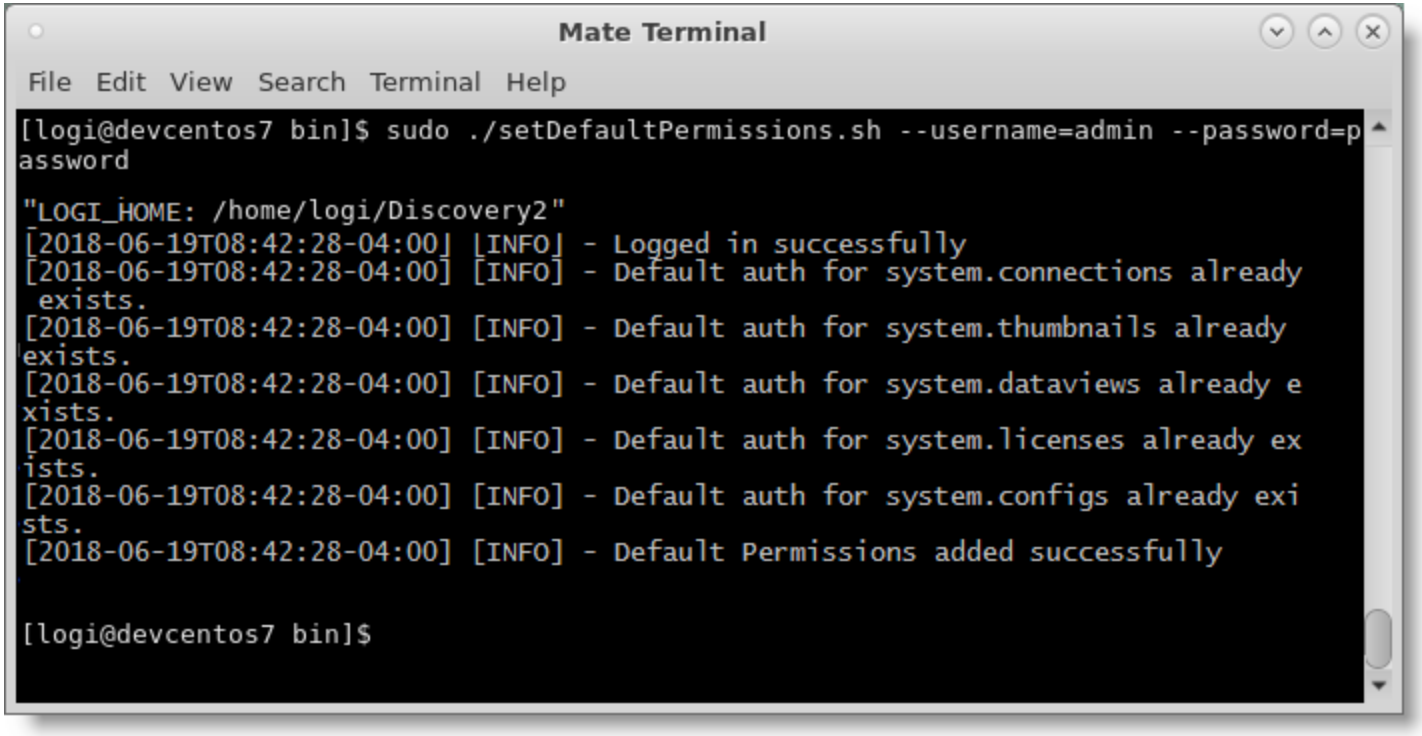
```
sudo chmod +x authorizationImportTool.sh
sudo ./authorizationImportTool.sh --authorization-file=/home/logi/Discovery2/platform/authorizationImport-
examples/syspublic-authorization.xlsx --user=admin --password=password
```



and the output is shown above. Then run the second command:

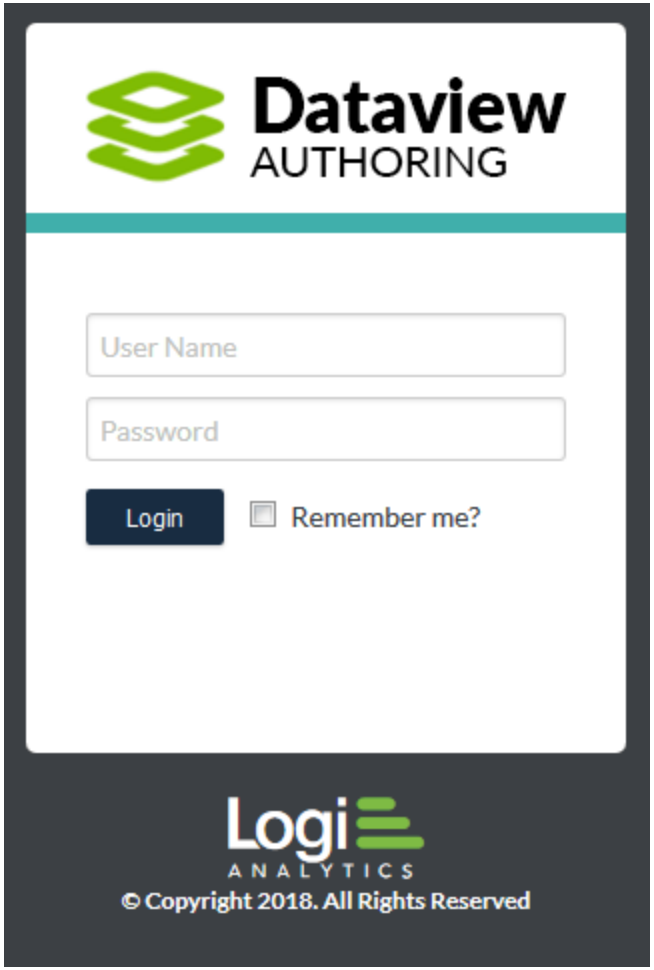
```
sudo chmod +x setDefaultPermissions.sh
sudo
```

```
./setDefaultPermissions.sh --username=admin --password=password
```



as shown above. You can close the Command Window now.

16. Stop and restart the Logi services again.



17. In your browser, navigate to localhost:5000/Datahub (using whatever port value you entered in Line #40 in Step #9, 3000 is the single instance default) and then login as "admin" with "password".

That completes the installation and configuration. Repeat the process for each new instance you add.

Installing Logi Studio

Welcome to Logi Analytics reporting products.

The following topics guide you through installing Logi Studio on a single computer for use with a Java-based web server:

- Installation Scenarios
- Product Licenses
- Installing Logi Studio
- Installing Logi Scheduler for Java
- Installing Add-on Modules
- Modifying or Repairing a Studio Installation

Administrator privileges on the target computer are **required** to complete the installation of Logi reporting tools. Logi Info for the Windows environment requires the .NET Framework 4.x. If not already in place, with your consent, appropriate versions of the .NET framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

Installation Scenarios

The installation file you received contains everything you need to create both .NET and Java web applications. The distinction between the two is made when you start to build an application. There are two major parts to the product in the installation file: **Studio** (the development tool) and the **Server Engine**.

Studio is a Windows application that's installed on a Windows platform and the **Server Engine** is a set of files that's added to each Logi application and is processed by the web server. When you build an application, Studio adds the engine files to the application folder.

A typical installation scenario for developers who wish to create Logi applications for Java is to install Studio on a Windows **development** machine, equipped with the Oracle JDK or OpenJDK 8, 11, 12, 13, or 14 (Info v12.6 SP2+ required for version 11, 12, 13, and 14) and one of the approved web servers. In this way, development and testing can be done on a single non-production machine.

 Oracle has changed its Java usage policies - see "Java Usage Policy" on page 342 for important information.

However, for deployment to production, only the **Server Engine** files (and your application) need to be copied to the production web server, which can be a Linux/UNIX or Windows platform. This is discussed in more detail below.

32- or 64-bit?

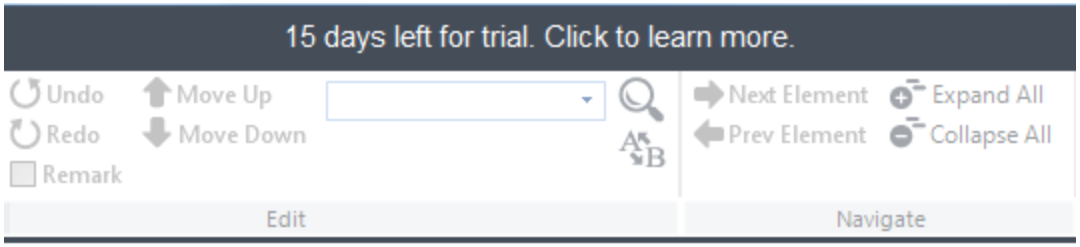
Logi Info Java applications will run on 32-bit and 64-bit Linux/UNIX platforms.

Java Server Configurations

 Important configuration information for each Java web server we support can be found in "Java Server Configurations" on page 346.

Product Licenses

Current Logi Info versions come with a built-in **15-day trial license**. You don't need to do anything but install the product and you can begin using it immediately. A clearly-visible display, shown below, in the Studio main menu counts down the days remaining in the trial period.



Clicking the counter display will take you to a web page that offers information about **purchasing** a Logi Info license.

After the trial period expires, Studio and any Logi reports you may have developed will *no longer run* without a real license.

Logi Analytics licenses are **server-based** rather than individual-user or concurrent-access licenses, so an unlimited number of end-users can access Logi reports through a single web server.

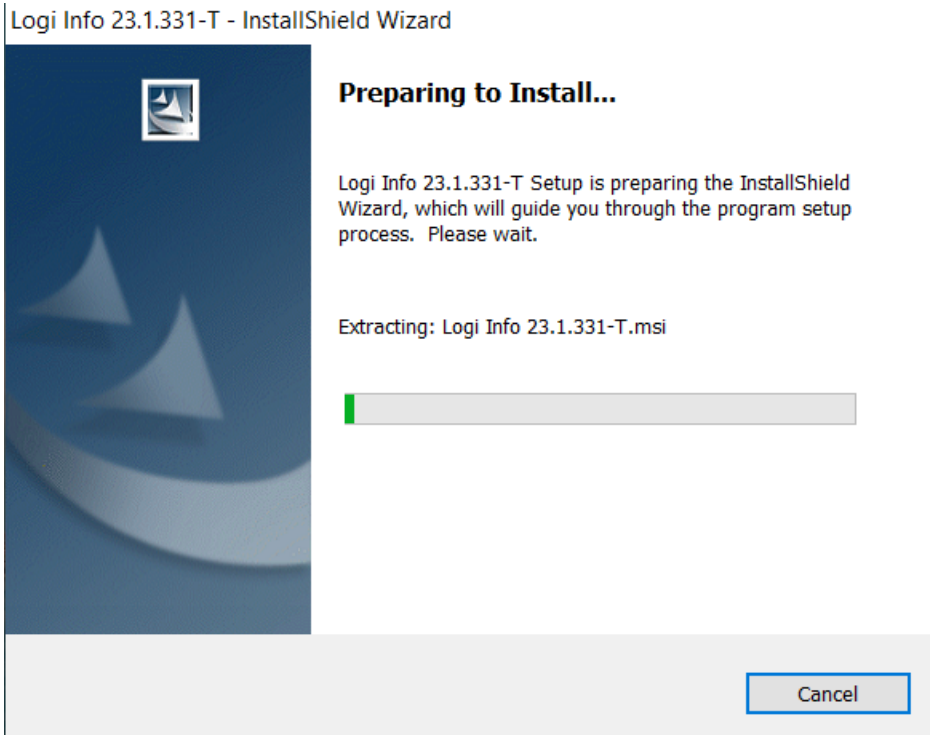
Our licensing scheme allows you to deploy our product on one development machine and on one production server. Additional separate licenses for Studio, for additional developers, are also available.

Licenses are keyed to you or your organization; they take the physical form of license files, which are assigned to a specific computer. DevNet includes a **License Management** page where you can manage your licenses, including reviewing them, assigning and un-assigning them to machines, and generating license files, at any time, without any interaction with our staff. For more detailed information about licenses, see "Product Licensing" on page 39.

Installing Logi Studio

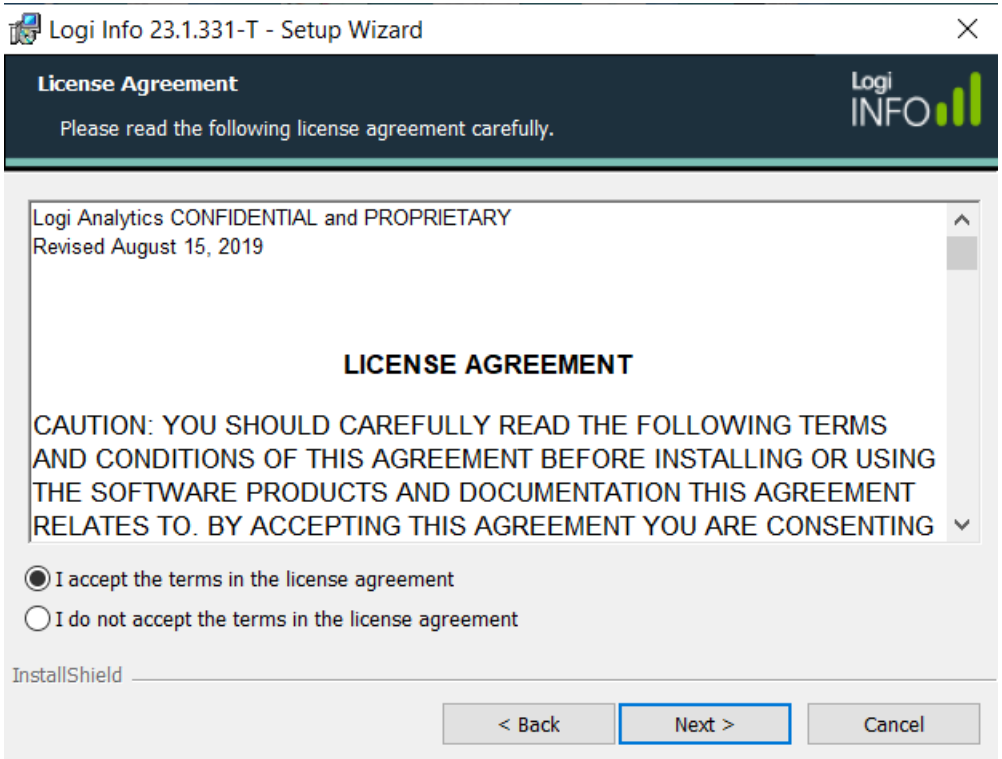
This topic demonstrates how to install Logi Studio v23.1 for use with a Java-based web server.

1. Double-click the Logi product **installation file** to launch InstallShield. Allow it to complete the installation preparation.



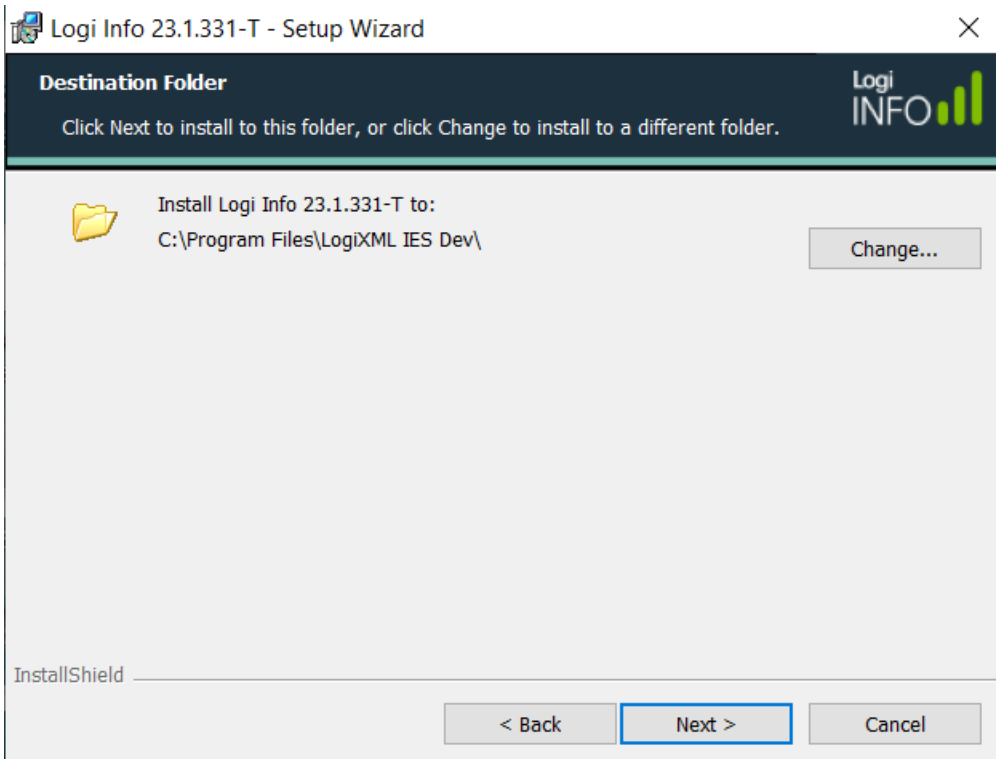
2. When the **Welcome Screen** appears, select **Next:**

3. **License Agreement**: Select the "I accept the terms..." radio button after reading the license agreement and select **Next** to continue:

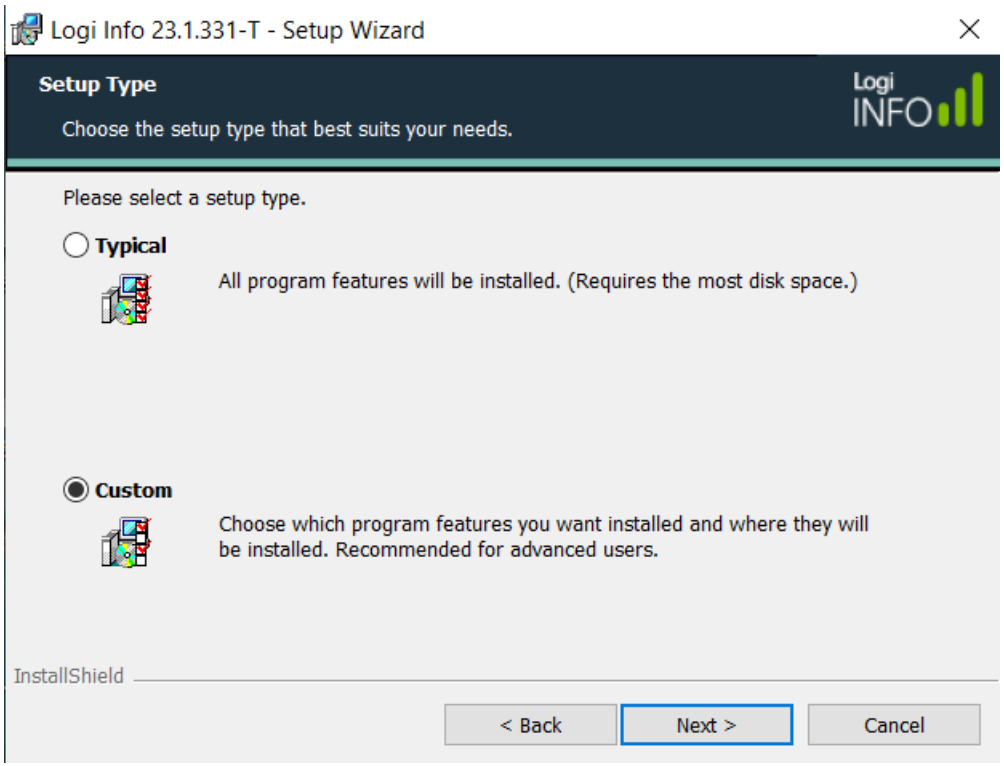


As usual, you can select **Back** at any time before the physical installation begins to go back to the previous screen

4. **Destination Folder:** *Optional* - select **Change** to specify an alternative installation location if you don't like the default location. Select **Next** to accept the installation location and continue:



5. **Setup Type:** Select the **Typical** or **Custom** radio button (see Custom information below) and click **Next** to continue:



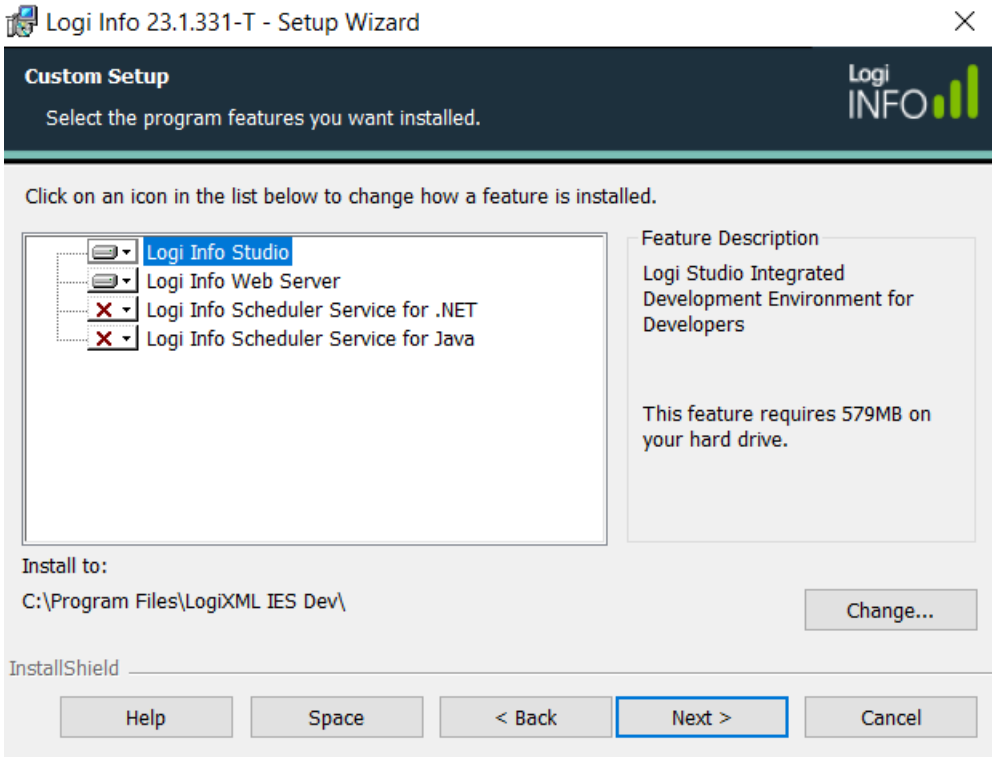
If you selected a "Typical" setup, **skip ahead** to Step 6. If you selected a "Custom" setup, the dialog box (shown below) appears. The following components are available during a **Custom** setup:

- **Studio** - The integrated development environment used by developers to create applications and report definitions. Select to remove Studio from the installation if you're only installing the Logi Server Engine.
- **Server** - The Logi Server Engine that processes XML data in report definitions and outputs HTML (includes Server Manager).

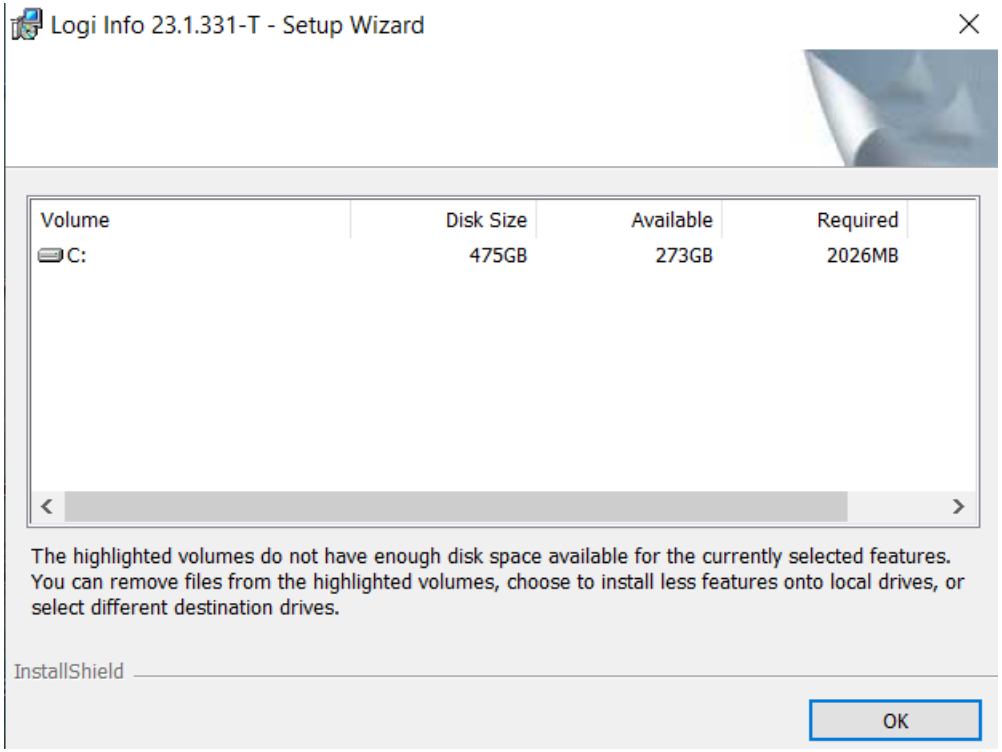
- **LogiXML Scheduler Service for .NET** - The Logi Windows Service that manages scheduled events; not required if you want to test scheduled report generation and distribution using Java facilities.

- **LogiXML Scheduler Service for Java** - The Logi Java daemon that manages scheduled events; required if you want scheduled report generation and distribution on Linux/UNIX-like systems.

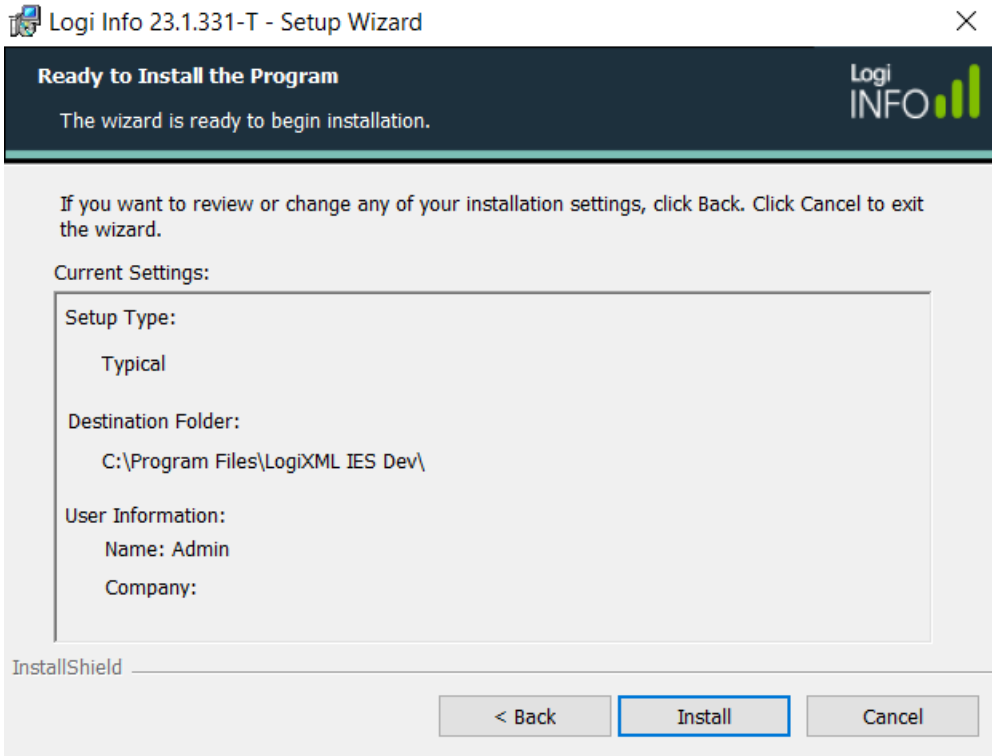
Click any of the components shown above and make a selection from the pop-up menu to include them in the installation. If desired, select **Space** to review the disk space requirements for the Custom setup you've selected. Select **Next** to proceed without the review:



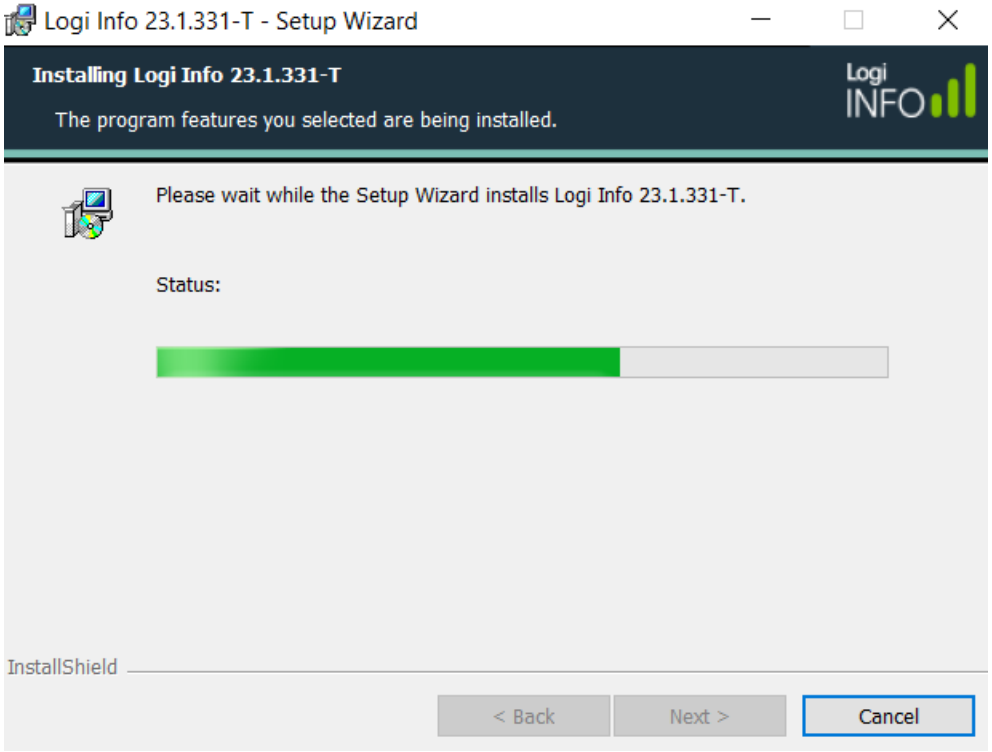
The **Disk Space Requirements** display will give you information about the available storage space and warn you if there is not enough space to complete the installation. You can repeatedly adjust the components in the Custom setup and see the effect on storage here, if necessary. Select **OK** to return to the previous dialog box.



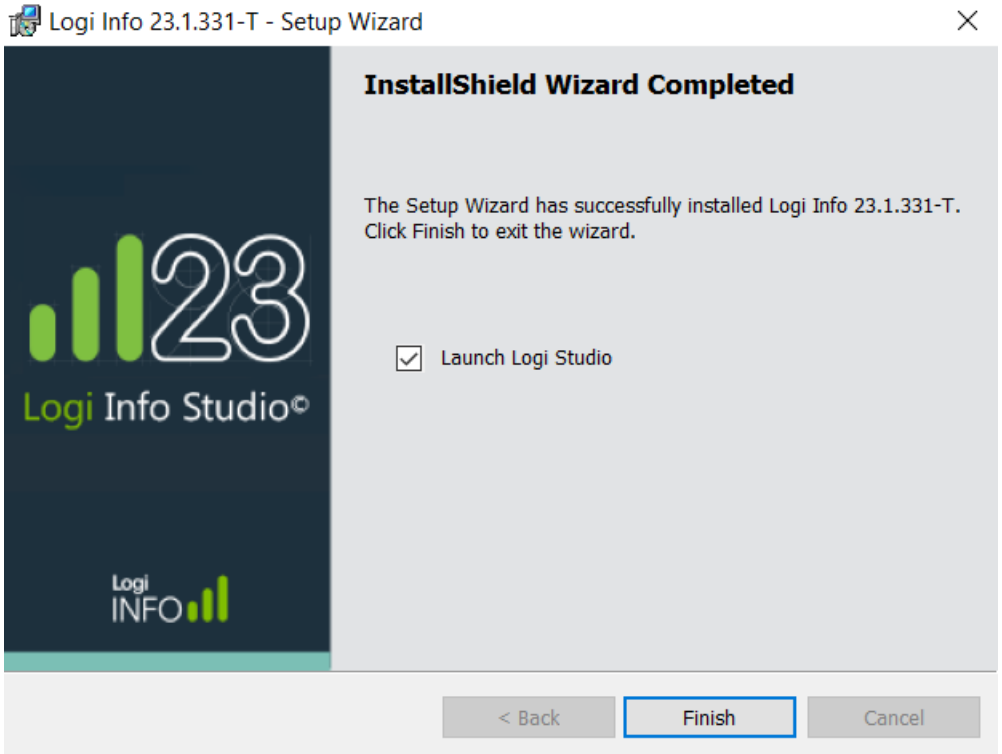
6. **Installation Summary**: Review the installation summary and select **Install**:



The physical installation will begin and you'll see several progress indicators for different tasks:



8. **Installation Complete**: Select **Finish** to exit the installer (if you have only installed the Logi Server Engine, there will be no "Launch Logi Studio" check box visible).



9. If you left the "Launch Logi Studio" check box checked, Studio will now launch and you should see a splash screen, like below:

You can also launch Studio by using Start Menu → All Programs → Logi Info or Report → Studio.

Should you need to, you can launch Server Manager using Start Menu → All Programs → Logi Info or Report → Server Manager or from Studio's Tools menu. More information about using Server Manager is available in Using Logi Studio.

Installation is complete and you may begin to use Studio and/or the Server Engine immediately.

Installing Logi Scheduler for Java

Logi Info includes the Logi Scheduler for Java, which runs on **Windows** or **Linux**. In simple terms, the Scheduler is a proprietary application that runs "in the background" on your server as either a Windows service or as a Linux/UNIX daemon.

Services of this type are generally configured to start up when their host computer is started and therefore run automatically with little user intervention. From a performance perspective, they pose little overhead as they consume very few resources and spend most of their time idling.

When the Scheduler "runs" an event from its database, it does so by calling a specified task in a process definition in a Logi Info application. That task can then run reports, send emails, export data, etc.

When you run the Logi Info installation and elect to install the Logi Scheduler for Java, it places all of its files in:

```
C:\Program Files\LogiXML IES Dev\LogiXML Scheduler Service Java
```

To install the scheduler on a production server, you need to copy the contents of this folder to it, in the directory of your choice.

Due to the nature of the Scheduler and its interactions with Logi reports, you can install the Scheduler on a different server than the web server, if desired.

For Windows

The Scheduler is started by running `<yourDirectory>\bin\Scheduler.bat`. If you wish to start Logi Scheduler for Java automatically in Windows, then use the Registry editor to add a String value in

```
HK_Local_Machine\Software\Microsoft\Windows\CurrentVersion\Run
```

named `LogiSchedulerJava` and set its data value to the path to `Scheduler.bat`, for example:

```
C:\Program Files\LogiXML IES Dev\LogiXML Scheduler Service Java\bin\Scheduler.bat
```

When running Scheduler for Java, the Windows User Account Control (UAC) system can interfere with normal operations. The Log, RunNowTasks, and Schedules directories require Write file access and may be blocked by UAC. Options to resolve this included disabling UAC and using UAC to set the file permissions of these directories to allow Write access.

For Linux/UNIX

The Scheduler is started by running `<yourDirectory>/bin/Scheduler.sh`.


The use of symbolic links in Linux for this application should be avoided as they may lead to unstable behavior. If you wish to start Logi Scheduler for Java automatically in Linux, add a line to `/etc/rc.d/rc.local`. The invocation should be similar to

```
/opt/rdScheduler/Scheduler.sh &
```


The ampersand (&) at the end is significant: it causes the application to load as a background process.

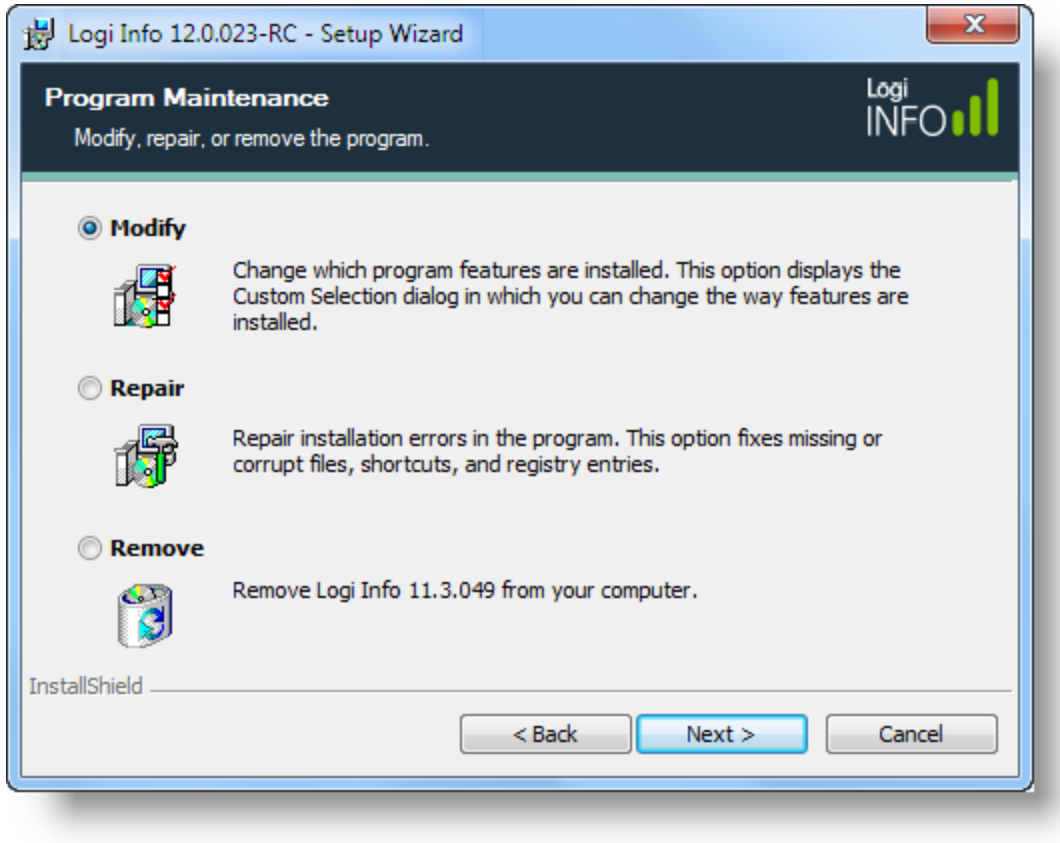
Logi Scheduler for Java logs events to the Linux Syslog. Normally these events do not require monitoring but, if issues do occur, the logs can be reviewed for diagnostic clues. The Linux Syslog utility does not permit remote access by default, so to enable this functionality, the `/etc/init.d/syslog`file needs to be modified. In it, the line invoking syslogd needs to be modified to include the `-r` option.

Installing Add-on Modules

Logi Info supports "Add-on Modules", optional software packages that enhance and extend the functionality of the product. They're installed separately, after Logi Info is installed. *Add-on Modules* describes them and includes links to their installation topics.

Modifying or Repairing an Installation

Suppose you suspect a .DLL file is missing or is corrupted and you want to fix it, or you need to otherwise change your Studio installation.



These kinds of situations can be addressed by either modifying or repairing the installation, which you should do by re-running the installation .exe file. *Do not* use Control Panel →Add-Remove Programs to do this; it will request an .msi file, which is not retained after the original installation.

IIS Express Web Server

The Logi Info product installation file includes the compact, .NET-based web server **IIS Express**. When you run its New Application wizard to create a new .NET Logi application, Logi Studio will offer to install this web server on your computer if it detects that no other web server has been installed.

The following topics discuss this web server in more detail:

- Installing IIS Express
- Managing Applications
- Removing IIS Express

About the IIS Express Web Server

The IIS Express web server is a compact version of Microsoft IIS. It's available for free from Microsoft and Logi Analytics has chosen to distribute it with Logi Info.

IIS Express is a light-weight web server that's ideal for development purposes. It runs as a Windows application (not a service) and is compatible with most modern flavors of Windows, including all editions of Windows 7, Server 2003 SP2, Server 2008, Server 2008 R2, Server 2012, and Server 2016. It requires .NET 4.x and does not require Administrator privileges to perform most tasks.

IIS Express is an ideal solution for developers or evaluators who do not want to, or cannot, install IIS on their development or evaluation computers. However, IIS Express is not suitable as a production web server as it has some performance and security limitations.

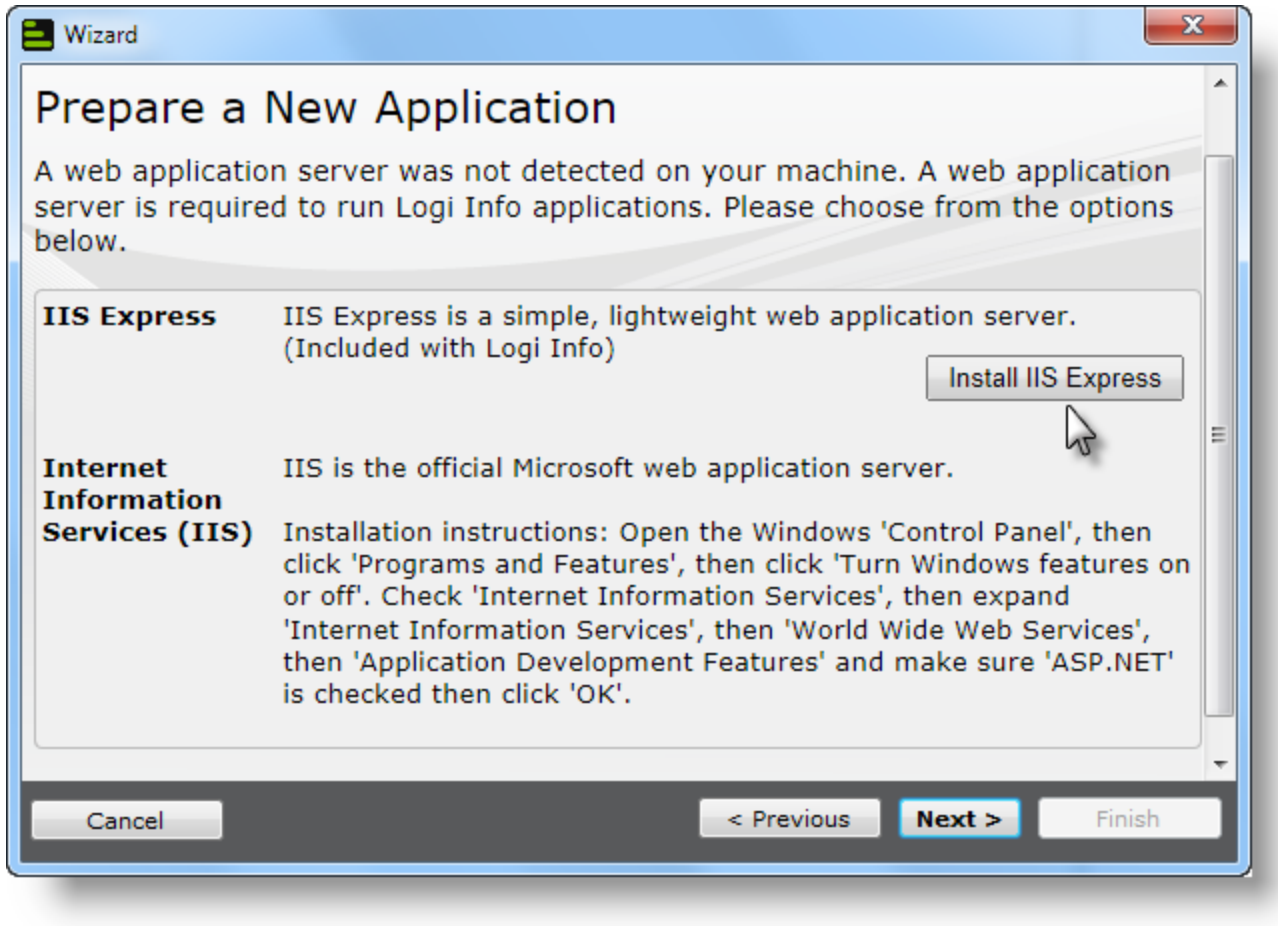
Complete information about IIS Expressis available from Microsoft.

Installing IIS Express

Please note that, although it's included with the product, IIS Express is *not* automatically installed when Logi Studio is installed. Developers who have no need of IIS Express (i.e. those who have IIS installed already) will never see it.

If you have regular IIS installed and decide to "turn it off" in Control Panel in order to see how IIS Express works, when you turn it back on later you may not have all of your IIS custom Application Pools and Virtual Directories restored! If that happens, they can be restored as part of a general System Restore Point restore operation (assuming you have a recent Restore Point saved). This is not something we recommend, in general, so if you must do it, do it on a dedicated test machine, not your development or production machine.

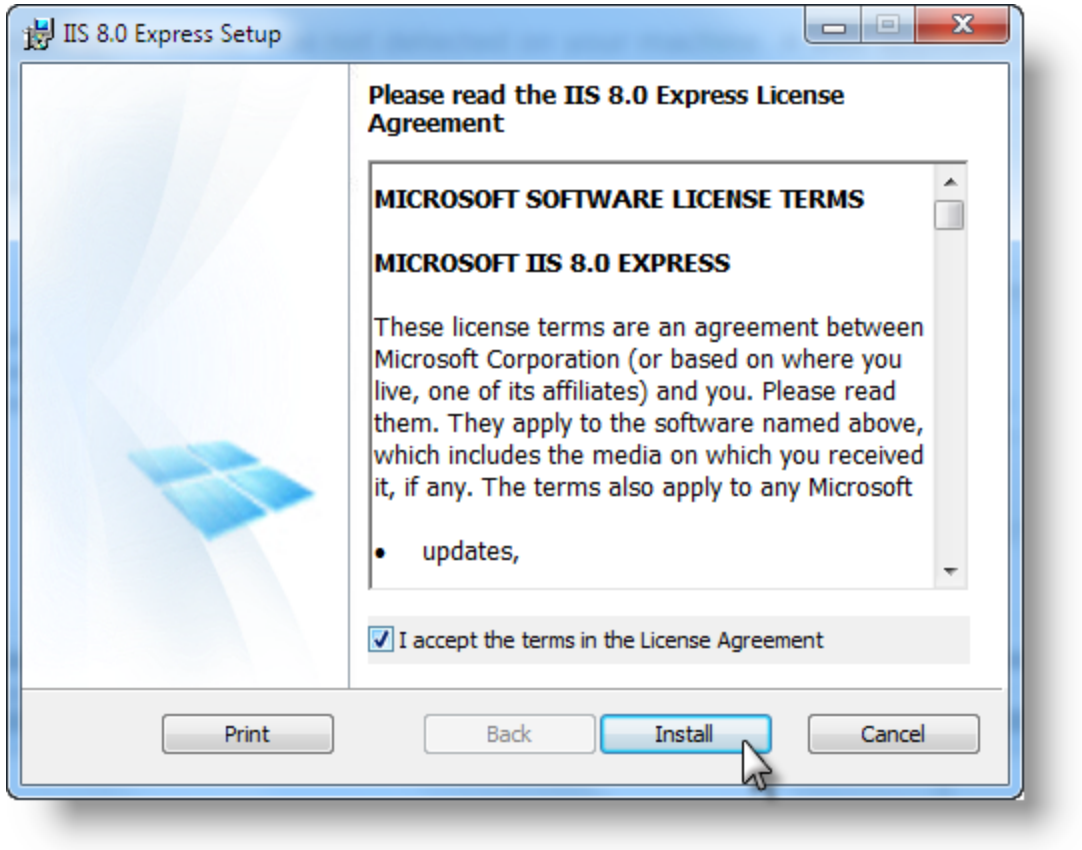
The first time Logi Studio is used to create a new application (by selecting File→New Application... from Studio's main menu), its New Application wizard will determine whether IIS is installed on the computer where Studio is installed.



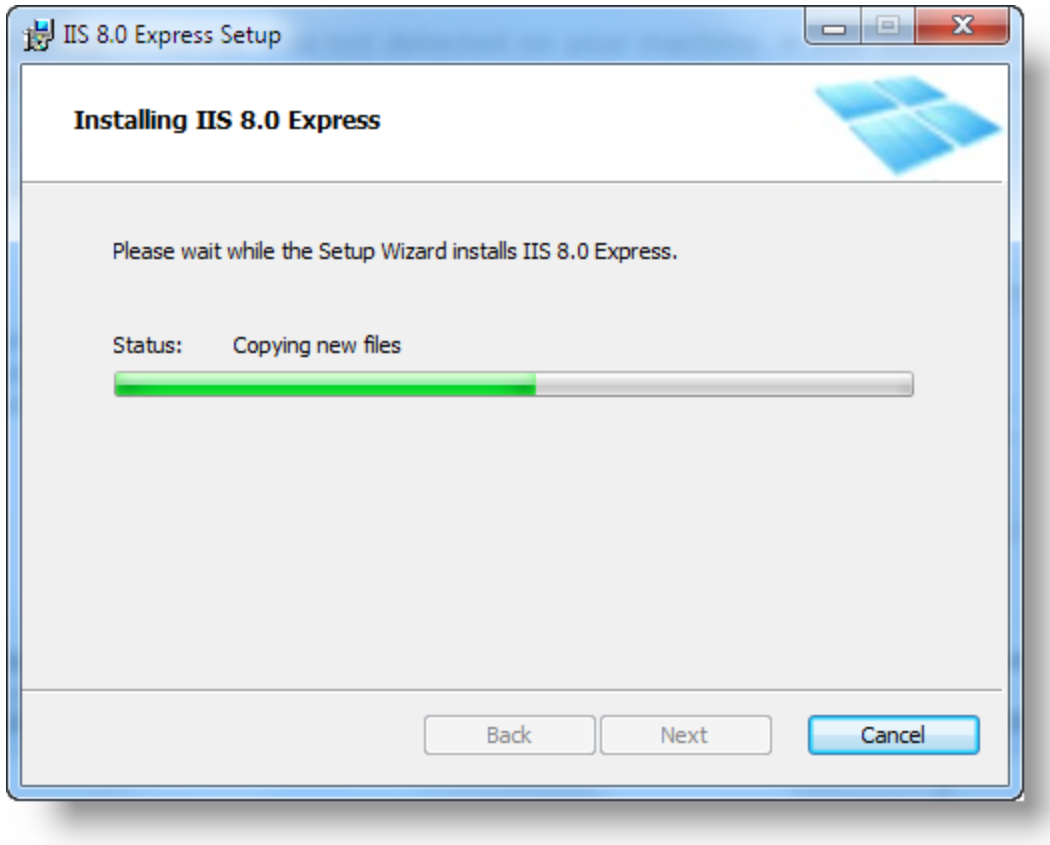
If Studio can't detect an IIS installation, the dialog box above will be displayed. The user can then decide whether to install IIS Express or IIS. Why would anyone decide to use IIS Express instead of IIS?

- IIS Express, as installed by Studio, is configured and ready to go immediately and does not need the additional configuration IIS requires.
- IIS Express can be installed and administered by users who do not have Admin privileges on their computer.
- IIS Express can be easily uninstalled when no longer needed.

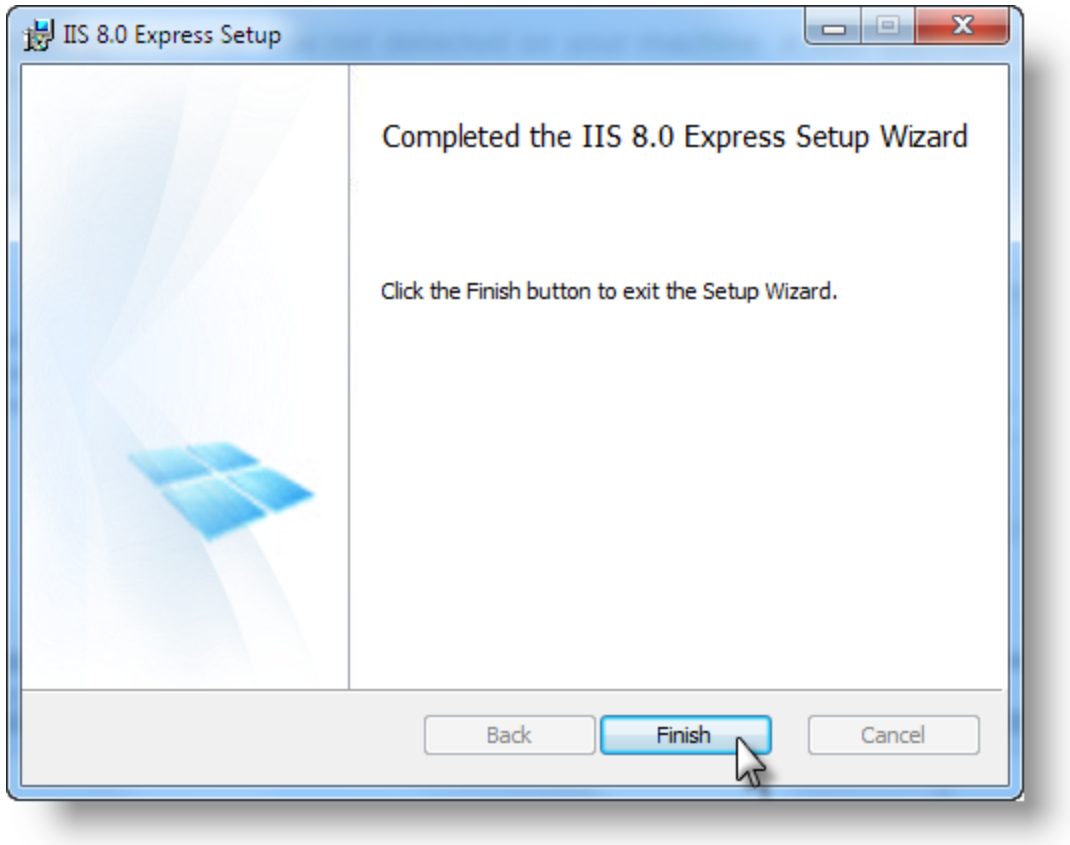
If you decide to install IIS Express, click **Install IIS Express**, as shown above, and a separate installer is launched:



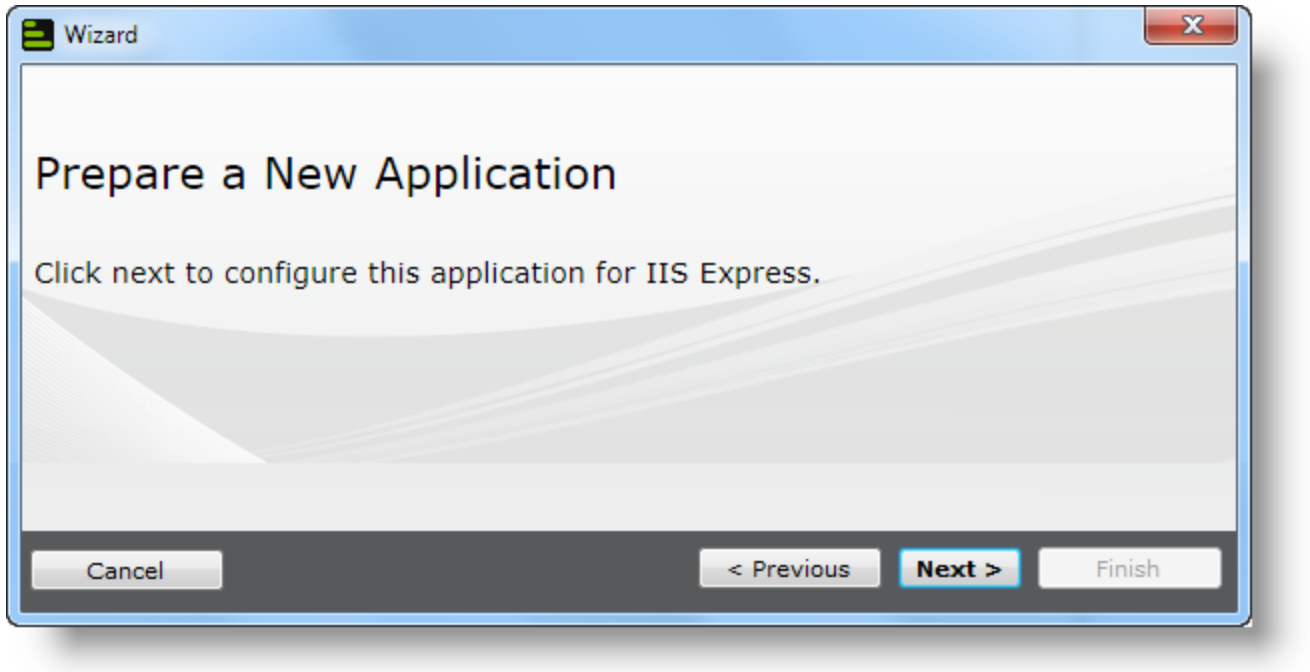
As shown above, you'll need to accept the License Agreement and click **Install**.



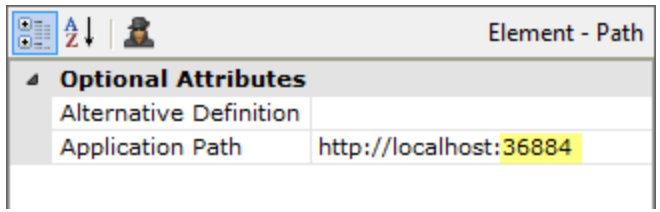
Once the installation starts, progress will be displayed in the dialog box shown above.



When the installation is complete, click **Finish** to return to Studio's New Application wizard.



Once IIS Express is installed, Studio becomes aware that it's dealing with IIS Express and not IIS, and it adjusts its prompts, wizards, and activities to be compatible with IIS Express, as shown above.



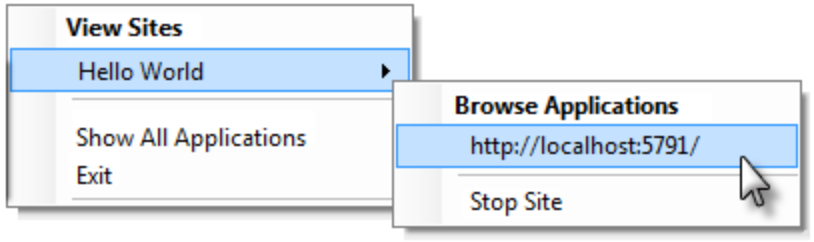
Once the wizard finishes, the only difference you can see in your report will be in its _Settings definition. The **Path** element's **Application Path** attribute will have a an address value similar to the one shown above. The high-lighted part is a port number, which will be different for each Logi app you create. Your Logi applications will be displayed just as they would if you had regular

IIS installed: you can preview them in Studio and browse them by using the Studio browse button and link or by entering their URL in an independent browser window.

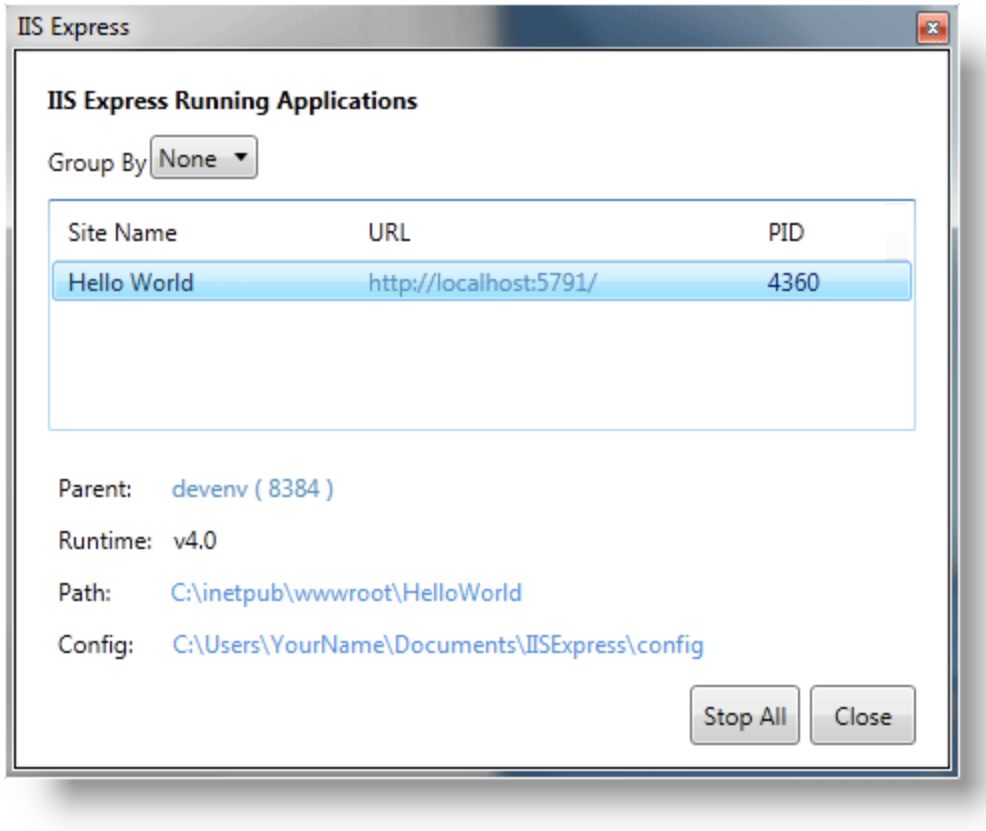
Managing Applications

You can see what applications are running on IIS Express using a tool that's available in the System Tray:

The IIS Express icon is shown above. If you don't see the icon in your System Tray, click the Hidden Icons button. Right-click this icon to see the context menus shown below:



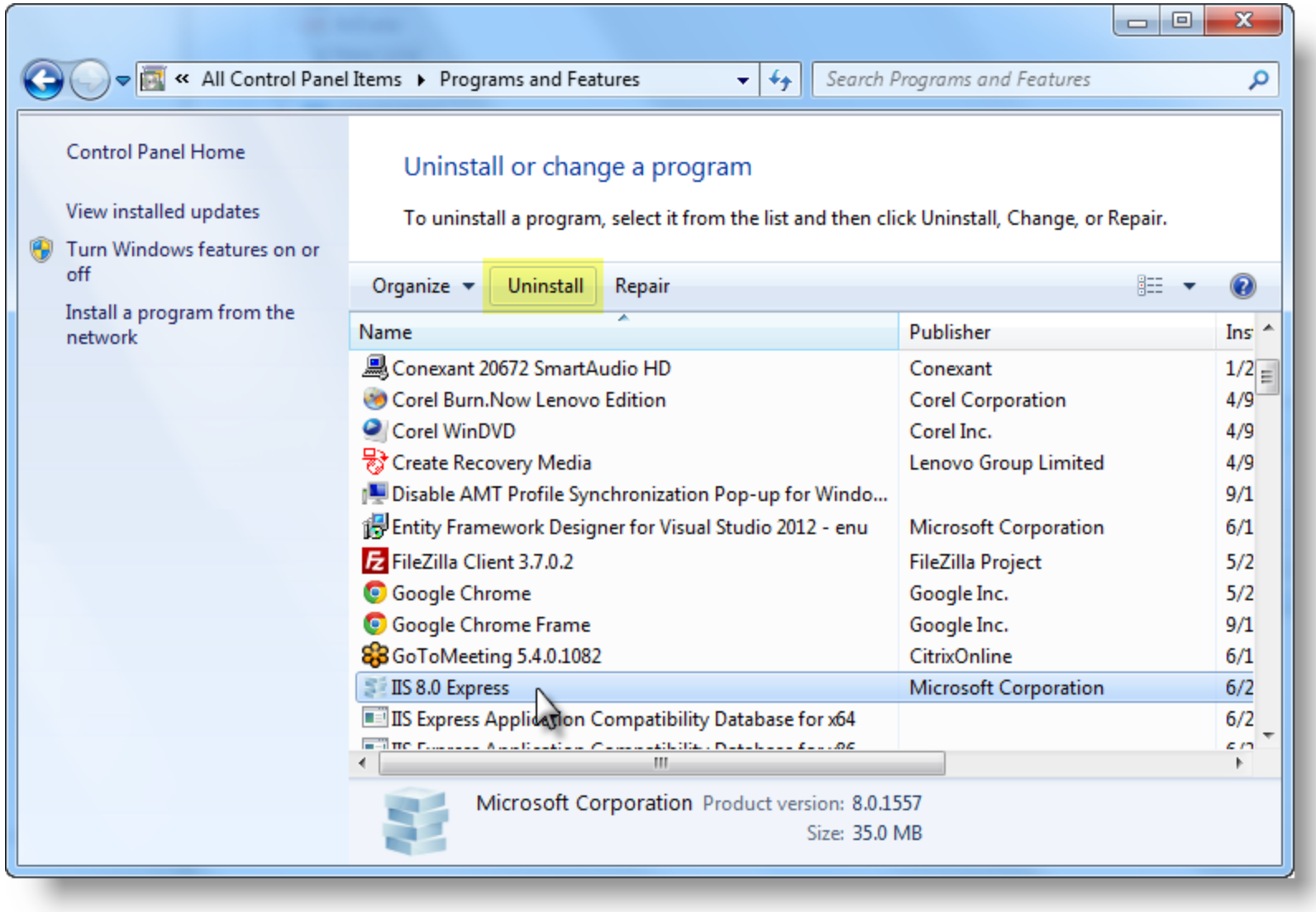
These menus allow you to view applications (web sites) that are running and stop them. This is a great way to re-discover the port number used for each application, for use in a URL, if you forget it.



If you click Show All Applications in the context menu, the dialog box shown above is displayed. It lists the Logi applications running in IIS Express and their details.

Removing IIS Express

IIS Express can be removed from your computer using the standard Windows Control Panel **Programs and Features** applet:



If you subsequently wish to use your Logi applications with IIS, be sure to check that your Application Path, in the _Settings definition, is correct after registering the application with IIS through Studio.

Command Line Install

This topic discusses command-line installation, which lets you run the Logi Info installation from a Windows Command Prompt window.

About Command-Line Installation

Command-Line installation allows you to install Logi Info without using the regular, interactive installer interface. This can be useful if you use scripts to prepare servers, or do installations in an automated fashion across a network, or work in an environment where the Windows desktop is unavailable for security reasons. This is also known as a "silent installation" because no user responses are required.

When installing Logi Info using this type of installation, it's assumed that the pre-installation software prerequisites have been met (.NET Framework 4.x and IIS web server installed or enabled) and that you accept the License Agreement. The default installation folder will be: `C:\Program Files\LogiXML IES Dev`. Review the regular installation documentation for details about the software prerequisites, if necessary.

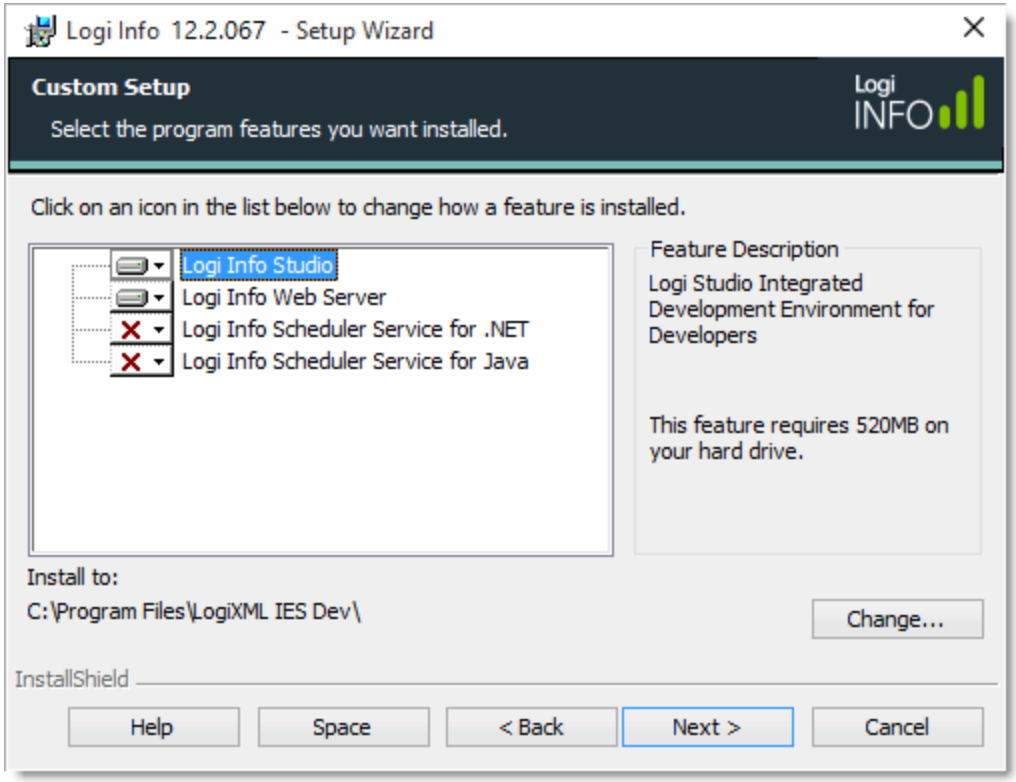
In order to run a command-line installation, the installing entity (you, or maybe a script) *must* be able to open a "Command Prompt" window, or run a PowerShell session, as an Administrator.

For examples of command-line installation, see Command Line Installation Examples.

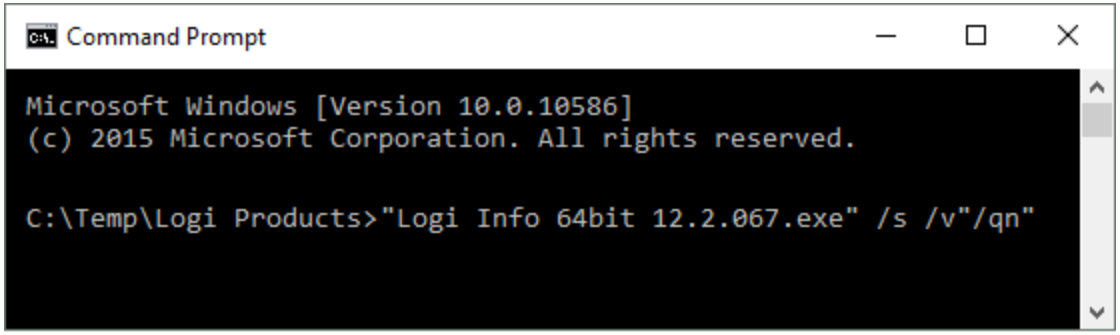
Command Line Installation Examples

The following examples will show you the commands for five common installation operations. The examples assume the commands are executing in the folder with the Logi Info installation file - they do not include any navigation commands you may need to get to that folder.

For orientation purposes, the image below shows you the four program features that appear in the regular installation interface:



The following examples will show you the commands for five common installation operations. The examples assume the commands are executing in the folder with the Logi Info installation file - they do not include any navigation commands you may need to get to that folder.



An example Command Prompt window is shown above, with an installation command.

- Notice the quotation marks - they are very important!
- We *highly* recommend that you test your commands on a non-production machine first to ensure that you get the desired results.

Logi Info Studio + Logi Info Web Server

To install Logi Info Studio and the Web Server engine on the same computer, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn"
```

Don't forget to include *all four* double-quotation marks.

To install Logi Info Studio and the Web Server engine on the same computer but into a specific drive and folder, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn INSTALLDIR=\"D:\TargetFolder\Folder\""
```

The INSTALLDIR argument can be added to any of the installation commands.

 It uses *escaped* quotation marks (\") inside the surrounding quotation marks.

Logi Info Studio only

To install just Logi Info Studio, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn ADDLOCAL=AlwaysInstall,Studio"
```

Logi Info Web Server Engine only

To install just the Logi Info Web Server engine, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn ADDLOCAL=AlwaysInstall,Server"
```

Logi Info Scheduler for .NET

To install just the Logi Info Scheduler Service for .NET, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn ADDLOCAL=AlwaysInstall,SchedulerNet"
```

Logi Info Scheduler for Java

To install just the Logi Info Scheduler Service for Java, use this command:

```
"Logi Info 64bit 12.2.067.exe" /s /v"/qn ADDLOCAL=AlwaysInstall,SchedulerJava"
```

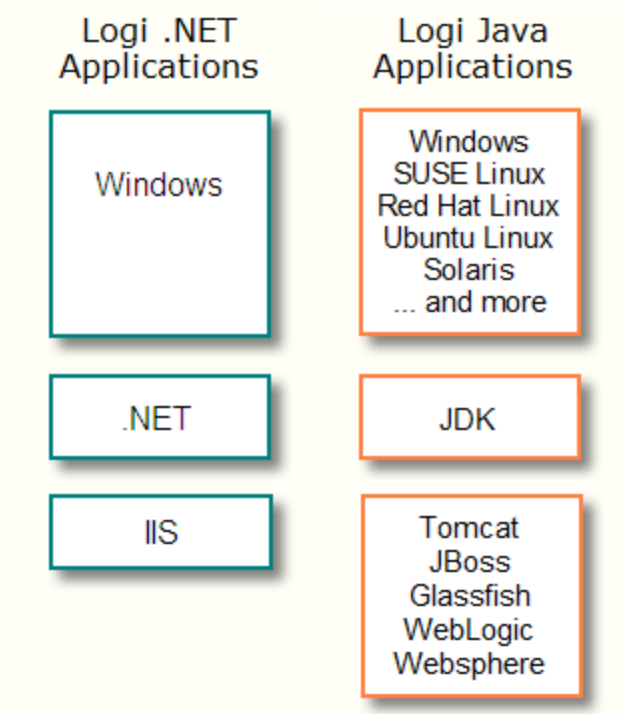
About Logi Apps and Java

Logi Info v23.1 produces web applications that use either the .NET Framework or Java libraries. You can run Logi Java applications on web servers under Linux and other UNIX variants, as well as under Windows. This topic describes some of the development options you have for .NET and Java libraries and provides links to related topics.

For information about installing Logi Studio, see "Installing Logi Studio" on page 295.
For information about configuring Java servers for use with Logi applications, see "Java Server Configurations" on page 346.

Logi Info consists of two parts: **LogiStudio** and the **LogiServer Engine**. When you select a Java application type during development, Studio generates applications that use JDK libraries rather than the .NET framework. The Server Engine includes special components that allow you to deploy Logi applications as Java applications on Windows or Linux/UNIX platforms.

Development Options



Your process for creating Logi applications is identical regardless of which application ty
some specialized elements to support the Java environment, such as **Connection.JDB**

The **formats** of definition, support, and other files that make up a Logi application **are**
aged reporting products.

Logi Studio, though Java-aware, is a Windows application. Developers creating Logi a
develop their Logi applications, regardless of the final deployment environment. Your d
and testing their application entirely on a Windows platform before moving it to a produ

Deployment Options

You can deploy Logi applications on servers running Windows or Linux/UNIX. You can n
ers and enable it to use the JDK by adding **special folders** to the application folder (ch
the application path and connection attributes, may also be necessary).

The following topics provide additional information about Logi apps and Java:

- General Requirements
- Product Licensing
- Differences from Applications for .NET
- Sharing Session Variables with Other Apps
- Cross-site Scripting (XSS) Protection

Logi Apps General Requirements

There are specific hardware and software requirements that you must match to use Logi apps. This topic describes the requirements.

Please review the following requirements carefully to ensure that your hardware and software are suitable.

Hardware

We do not have specific hardware configurations to recommend for use with the product. Logi Studio can only be installed on 64-bit Windows platforms. Logi Info is only available in a 64-bit version. Logi Info v12+ can be installed on a single computer under Windows Server 7 (supporting ending 2023), 8, 10, 2012 r2, 2016, and 2019 for use with the IIS web server. Logi Java applications must also be run 64-bit Linux/UNIX platforms.

Software

Review "System Requirements - Logi Info" on page 17 to ensure that your web server and other software are supported.

Oracle JDK or OpenJDK

 Oracle has changed its Java usage policies - please read our "Java Usage Policy" on page 342 for important information.

The official Oracle Java Development Kit or OpenJDK 8, 11, 12, 13, or 14 (Info v12.6 SP2+ required for 11+) is required for Logi Java applications, as it includes the Server JVM and will provide faster performance. You must install the JDK on the web server, the JRE is not sufficient.

Database Connectivity and Tools

Connectivity to databases such as **Oracle**, **MySQL**, and **MS SQL Server** is supported via native connections. Connections using **JDBC** are also available. Connections to a variety of non-database datasources are also supported, including XML, CSV, Excel data files, and web services.

We provide specific MySQL and MS SQL Server database drivers with the Logi Server Engine, so developers do not need to separately download and install them. However, these may not be the correct version for all circumstances and developers may need to update or replace them.

Our SQL Server JDBC driver works with JDK or OpenJDK 8, (Info v12.6 SP2+) 11, 12, 13, or 14 and MS SQL Server 2005+. The previous version of the driver's `sqljdbc4.jar` file, however, is still provided, as `sqljdbc4.old`, for users who work with MS SQL Server 2000.

If you are working with **Oracle**, and wish to use the **Database Browser** and **Query Builder** tools within Studio, you may need to install and configure the **Oracle Client (OLEDB)** on the Windows platform where Logi Studio is installed.

Logi Info and SSRM now support Oracle 19c databases.

Required Platform Knowledge

If you or your developers are creating Logi Java applications for Linux/UNIX servers, you should ensure that you have a **good working understanding** of the OS, the JDK, the web server, and the admin/management tools on their server. The detailed knowledge required to successfully administer the servers can be **extensive** and **complex** and is well beyond the scope of Logi documentation.

Logi Info Product Licensing

Logi Info v23.1 comes with a built-in, **15-day trial license**. You need do nothing but install the product and you can begin using it. A clearly-visible display in Studio's toolbar counts down the days remaining in the trial period.

Select the counter display to see a web page that offers information about **purchasing** a regular Logi license.

After the trial period expires, Studio and any Logi reports you may have developed will *no longer run* without a valid license.

You *may not* use our products for redistribution with, or embed them in, other products without an **OEM license**; contact our Sales group for more information if you need an OEM license.

For more detailed information about licenses, see "Product Licensing" on page 39.

Differences of Java from .NET Logi Applications

Logi Java applications provide all of the features and functionality found in Logi .NET applications. This topic describes some of the differences between .NET and Java implementations.

Naturally, Logi Java apps do not support Windows-specific technologies such as OLEDB. Instead, Logi Info supports Analogous Java technologies, such as **JDBC**. However, for the best results, you should try to use *native* connections, such as Connection.MySQL, before using *generic* connections like Connection.JDBC.

JavaScript is the only scripting language you can use when building Java applications. However, you can use a built-in "VBScript emulator" for intrinsic functions, such as IIF(), which are modeled on VBScript. They're translated into JavaScript behind-the-scenes. Currently, Logi applications for Java *do not* have support for the following elements*:*

- Any OLAP elements (XOLAP elements, however, *are* supported)
- DataLayer.Web Scraper

For information about installing Logi Studio, see "Installing Logi Studio" on page 295.
For information about deploying Logi Java applications, see "Java Server Configurations" on page 346.

Sharing Session Variables with Other Apps

On a web server, Logi applications for Java and regular (non-Logi) Java applications maintain their session variables in different ways. And, typically, you access Logi application session variables using @Session tokens, while Java application session variables are accessed via JSP or by using the `javax.servlet.http.HttpSession` object. This topic describes these attributes.

If a Logi application for Java is to be integrated with a Java application, then how can their session variables be shared? The **Java Session Copying** element, available in the _Settings definition, provides a selective method for session variable copying between the two applications.

 In order to use this feature, you must not use session variable names that contain *spaces* in your Logi application, and there's a small performance penalty for this process, so it's best to minimize the number of variables copied. It may be necessary to restart your web server if you edit the lists in the attributes described below.

The element has four attributes, consisting of Regular Expressions, which control which session variables are copied. As it can be difficult to write Regular Expressions which both include *and* exclude strings, we've provided two attributes ("include") which are processed first, followed by two other attributes ("exclude") which are processed second, making it easier to accomplish both types of operations.

Attribute	Description
Copy From Java Exclude	Specifies, as a comma-separated list of Regular Expressions, the Java application session variables that will *not* be copied to the Logi application session variable space. This attribute is processed *after* Copy From Java Include, removing variables from that list.
Copy From Java	Specifies, as a comma-separated list of Regular Expressions, the Java application session variables that *will* be copied to the Logi application session variable space.

Attribute	Description
Include	
Copy To Java Exclude	Specifies, as a comma-separated list of Regular Expressions, the Logi application session variables that will *not* be copied to the Java application session variable space. This attribute is processed *after* Copy To Java Include, removing variables from that list.
Copy To Java Include	Specifies, as a comma-separated list of Regular Expressions, the Logi application session variables that *will* be copied to the Java application session variable space.

Cross-site Scripting (XSS) Protection

Cross-site scripting (XSS) is a common form of security vulnerability affecting web applications. Attackers use XSS to inject client-side scripting into web pages via the URL, tricking the browser into trusting scripts run from malicious hosts. This topic describes how you can protect your Logi applications from XSS attacks.

These scripts usually access user and session information stored in cookies, and allow the hackers to forge trusted user behavior. Most modern browsers include a defense against this, the X-XSS-Protection header, which is enabled by default. However, this may not satisfy penetration testing and applications may be determined to be vulnerable. Beginning with early versions of Info v12, Logi applications can be configured to enable XSS Filtering at the web server, as follows:

Once copied, session variables are identified, in the Java app, using its session object's attribute ID and, in the Logi app, as the name used with a @ Session token. For example: Java JSP:session.getAttribute("userEmail")
Logi app:@Session.userEmail~
If circumstances demand it, it's possible to return to an earlier, "copy all" behavior by manually adding the following XML to your _Settings definition source code, as a child of the <Setting> tag:

```
<JavaSessionCopying CopyToJavaInclude="^" CopyToJavaExclude="DebugFile,-bUsesSort$,-tra$,-xmlDef$,
 -Xsl$,-rdDef$" CopyFromJavaInclude="^" CopyFromJavaExclude="^rd,^dt" />
```

1. Edit the <yourLogiApp>\WEB-INF\web.xml file and uncomment the code section that begins with:

   <filter>
   <filter-name>XssFilter</filter-name> Save the file.

2. Edit the `<yourLogiApp>\WEB-INF\XssFilter.properties` file and provide appropriate filtering directives.

   As installed, the default `XssFilter.properties` file includes six directive examples. Delete them and add:

filter1=,)<>!(@,GET
filter2=,)<>!(@,POST The effect of the examples above is to throw an error and not execute any page requested with GET or POST using any URL that includes any of these characters: ()<>!@ . The format of a filter directive consists of three parts:

filter#= *urlPattern*, *forbiddenCharacters*, *requestType*where: *urlPattern* = the URL, for example "rdReport=Default"; leave this blank to apply the filter to *all* URLs
forbiddenCharacters = the characters in the URL that will cause an error; leave this blank to allow *all* characters
requestType = HTTP request type: either GET or POST Here's another example:

filter18=rdReport=yourCustomReport,,POST This example filters URLs requested using POST and containing "rdReport=yourCustomReport" and it allows *all* characters. Prior to **v12.5 SP2**, filters could require a large number of directives to cover all the possibilities, and these might have conflicted with URLs generated by super-elements such as the Analysis Grid. After that release, this is handled internally and one or two directives like the first two shown above are sufficient to provide broad, generic protection in most cases.

Java Usage Policy

Oracle recently changed their terms and conditions for support of Java.

If your product or applications use Java, please read the following important information:

- Oracle Policy Changes
- Impact on Logi Products

Oracle Policy Changes

Starting in January 2019, Oracle discontinued its free, long-term support of the **Oracle Java Development Kit** (JDK) for enterprises. All Oracle JDK releases now require a Support Agreement with Oracle. Going forward, there are two new types of support available:

- The Long Term Support (LTS) release includes support and security updates for *several years*. The current LTS releases are Oracle JDK 8 and Oracle JDK 11.
- The Non-LTS release includes support and security updates for *six months*. Each Non-LTS release will be superseded by the next one, i.e. there will be no further support or security updates provided for previous releases. For example, the latest non-LTS release was Oracle JDK 12, released in March 2019, and it will be superseded by the next non-LTS release, in September 2019.

Organizations may choose instead to *not* use the Oracle JDK and to use the free **OpenJDK** (also maintained by Oracle) instead. OpenJDK details include:

- OpenJDK release frequency matches the Oracle JDK non-LTS release frequency, every six months, with each superseding the previous release.
- OpenJDK release numbers correspond to Oracle JDK release numbers, e.g. OpenJDK 11 corresponds to Oracle JDK 11.
- OpenJDK and Oracle JDK releases will include similar functionality, with Oracle JDK expected to have some additional proprietary features that won't impact Logi products.
- While there is no cost for using OpenJDK, it's released under a GPL 2 license, with ClassPath Exception, which may matter to some Logi customers.

Impact on Logi Products

The impact of Oracle's new Java release and maintenance policy varies by Logi product, as follows.

Logi Info and Logi SSRM

Logi Info Java applications currently work with Oracle JDK 8 and OpenJDK 8. Logi Info 12.6 SP2+ applications now work with Oracle Java 11, 12, 13, and 14 and OpenJDK 11, 12, 13, and 14.

Logi Info and SSRM Version	OpenJDK Version
v12.5-v12.6 SP1	Open JDK 8
v12.6 SP2-v12.6 SP6	Open JDK 11
v12.7-v12.7 SP1	Open JDK 12 and 13
v12.7 SP2-v12.7 SP5	Open JDK 13
v12.7 SP5- v12.7 SP7	Open JDK 14

Going forward, Logi Info will be certified on the latest OpenJDK releases, within three weeks of those releases becoming available. In the event that changes are required in Logi Info itself as a result of a JDK upgrade, Logi will provide them as a new Service Pack for the latest Logi Info major release.

Logi customers who choose to stay on one of Oracle's LTS releases (e.g. Oracle JDK 8, 11, or 12) can continue to do so. If issues are found by those customers as a result of an update to the Oracle LTS release, or due to a new Logi Info release, Logi will attempt to resolve those issues. Customers should generally expect resolution via Service Packs in the two latest major releases.

Discovery Module and Logi Platform Services

 We will no longer bundle Java into our product installers. Customers are expected to get, install, and maintain the latest OpenJDK versions on their own.

The Logi Discovery Module (DM) v3.2+ and Logi Platform Services (LPS) v3.4+ will work with the latest version of OpenJDK. Future releases will typically be supported within three weeks of release of a new OpenJDK version.

If you have questions regarding Oracle's policy changes, please visit Oracle's support roadmap page. For questions regarding impacts on Logi products and applications, please contact Logi Customer Support.

Java Server Configurations

Logi products work with a variety of Java-based web servers, which often require specific configurations to work correctly and for optimum performance.

The following topics provide the details of these configurations for each server, including:

- Setting Java Options
- Apache-Tomcat 5.5 / 6 / 7 / 8
- JBoss 7.1
- JBoss 5.1
- JBoss 4
- Glassfish 4.1
- Glassfish 3.0 / 3.1/ 4.0
- Glassfish 2.1
- WebLogic 12c
- WebLogic 10
- Websphere 7 / 8.5
- WildFly 8 / 9 / 10

Please note the following important information:


Oracle has changed its Java usage policies - see "Java Usage Policy" on page 342 for important information.

- The Oracle JDK or OpenJDK 8, 11, 12, 13, or 14 is *required* for non-Apple platforms, or as noted in individual configuration topics linked above. JDK 7 is *required* for Apple platforms. The JRE is not sufficient.
- It's *possible* to use Logi products on systems that use symbolic links or the Filesystem Hierarchy Standard (FHS), but it makes configuration much more complicated and *we don't recommend it*.

- When Logi Studio creates a new application, it's deployed in exploded format within a single "Logi application folder", not as a WAR file. WAR deployments by Studio are not supported at this time.
- We no longer distribute the `ojdbc6.jar`. Info Java includes hardcoded references to certain Oracle datatypes, which are referenced in the Oracle JDBC driver. To allow Info Java to compile we included `LogiSTUB.jar`. This jar replaces `ojdbc6.jar`, but only to allow Info to run; it does not allow Oracle data to actually be retrieved. To retrieve Oracle data, download `ojdbc8.jar` and delete `LogiSTUB.jar`.

JDK/OpenJDK 11 is supported starting with this version but its use requires these configuration steps:

1. In JVM options, set `--illegal-access=permit`
2. In the server's server.xml file, set `relaxedQueryChars="[]"`
3. Copy *<LogiAppFolder>*`/rdTemplate/Java11/jaxb-api-2.4.0-b180830.0359.jar` to *<LogiAppFolder>*`/WEB-INF/lib`.

JDK/OpenJDK 11 and higher are supported starting with Info v12.7 but its use requires these configuration steps:

1. In JVM options, set `--illegal-access=permit` and `-Dnashorn.args="--no-deprecation-warning"`
2. In the server's server.xml file, set `relaxedQueryChars="[]{}|"` in the connector elements.
3. Copy `<LogiAppFolder>/rdTemplate/Java11/jaxb-api-2.4.0-b180830.0359.jar` to `<LogiAppFolder>/WEB-INF/lib`
4. Copy `<LogiAppFolder>/rdTemplate/Java11/javax.activation-1.2.0.jar` to `<LogiAppFolder>/WEB-INF/lib`

 PhantomJS in Linux, as we distribute, is complete. There is no requirement to install Qt, WebKit, or any other libraries. However, it still relies on Fontconfig (the package Fontconfig or Libfontconfig, depending on the distribution). Normally, this is present; however, if you run PhantomjJS and receive a 127 error you need to complete another step. Navigate to the application `root/WEB-INF/lib` directory and enter "`ldd phantomjs`". All the referenced libraries appear. If a version of Fontconfig does not

appear then it must be installed. FontConfig installation instructions for various distros vary. The Ubuntu instructions are included here:

```
sudo apt-get update -y

sudo apt-get install -y fontconfig-config
```

For more information, see https://phantomjs.org/download and https://www.freedesktop.org/wiki/Software/fontconfig/.

Setting Java Options

In general, the following JVM flags should be added to the Java options settings that are used to tune/monitor the JVM.

The **Performance Settings** are *strongly recommended* for the Oracle JDK, whereas the **JConsole** and **Tracing Settings** are optional but recommended, at least during initial implementation.

Typical Performance Settings

```
-server (JDK)
-Xmx4096m (JDK)
-XX:MaxPermSize=256m (JDK)
-XX:-DisableExplicitGC (JDK)
-XX:CompileThreshold=8000 (JDK)
-Djava.awt.headless=true (JDK)
-Djava.net.preferIPv4Stack=true (needed if sending email and using Java 1.7+)
 -d64 (JDK - for 64-bit JVMs only)
```

 The Java heap space setting, `-Xmx4096m`, shown above and in individual sections below is sufficient for a *minimal* development system, *not* for a production system. A setting more appropriate for a 64-bit production server would be something like `-Xmx8192m`. Consult your production server administrator.

64-bit JVMs require the `-d64` flag in the Performance Settings to force the JVM to use 64-bit addressing.

JConsole Settings (optional, for monitoring)

```
-Dcom.sun.management.jmxremote
 -Djava.rmi.server.hostname=<your host ip addr>
 -Dcom.sun.management.jmxremote.port=1099
 -Dcom.sun.management.jmxremote.authenticate=false
 -Dcom.sun.management.jmxremote.ssl=false
 -Dnashorn.args="--no-deprecation-warning" (for JDK 11/12 only, to reduce console noise)
```

The JDK includes the Java Monitoring and Management Console (**JConsole**) tool. It uses the extensive instrumentation of the Java virtual machine to provide information on the performance and resource consumption of applications running on the Java platform using Java Management Extension (JMX) technology. For more information about JConsole, see this article.

Tracing Settings (optional, for testing)

```
 -verbose:gc
 -Xprof
 -XX:-CITime
 -XX:-PrintCompilation
-Xloggc:$CATALINA_HOME/logs/gc.log
 -XX:+PrintGCDetails
 -XX:+PrintGCTimeStamps
```

The $CATALINA_HOME environment variable in the example refers to the folder into which Apache-Tomcat has been installed; for other web servers substitute the appropriate variable.

About the Performance Settings

What do the Performance Settings recommended above actually do? Here's an explanation:

Setting	Description
-server	This option instructs the launcher to run the Java JVM in Server Mode. The JVM can optimize a number of things for server environments, improving performance.
-Xmx	This option sets the maximum amount of memory that can be allocated to the JVM heap, improving performance. Minimum requirement is 4096 MB (4GB).
-XX:MaxPermSize	This option sets the maximum amount of memory that can be used for the permanent generation, or "PermGen", Java's fixed block of memory for loading class files. Failure to set this to at least 128m will cause errors to occur and a setting of 256m is recommended.
-XX:-DisableExplicitGC	This option disables calls to the function System.gc(), which is often run explicitly by many classes. When it runs, it triggers a full garbage collection process, which consumes a lot of execution time and results in inefficient heap usage. When it's disabled, the JVM still performs garbage collection nonetheless, whenever necessary.
-XX:CompileThreshold	This option sets the number of method invocations/branches before compiling. The JVM usually waits for a method to be executed a certain number of times before it's compiled. Not compiling every method helps startup time and reduces RAM footprint. This option allows you to control that

Setting	Description
	threshold.
-Djava.awt.headless	This option, when set to *true*, prevents graphics rendering code from assuming that a graphics console exists, avoiding any chance of encountering an obscure Java bug related to X-servers and graphics rendering.

"Hot Deployment" Caution

Be aware that Tomcat, JBOSS, WebLogic, and other servers that offer "Hot" or "Auto" deployment may produce errors when using Logi Studio wizards. If this deployment feature is enabled, the changes the wizard makes may be deployed before the wizard completes all of its steps, causing an error.

Typically, "Hello World"-type Logi apps will work fine in this environment but use of Studio wizards to build more complex apps will not.

To avoid this problem, either disable the hot or auto deployment feature (technique varies by server) or increase the "scan interval" used by the feature substantially.

Apache-Tomcat 5.5 / 6 / 7 / 8

This topic demonstrates how to configure Apache-Tomcat 5.5 / 6 / 7 / 8.

1. Logi application folders must be stored beneath the `webapps` folder.

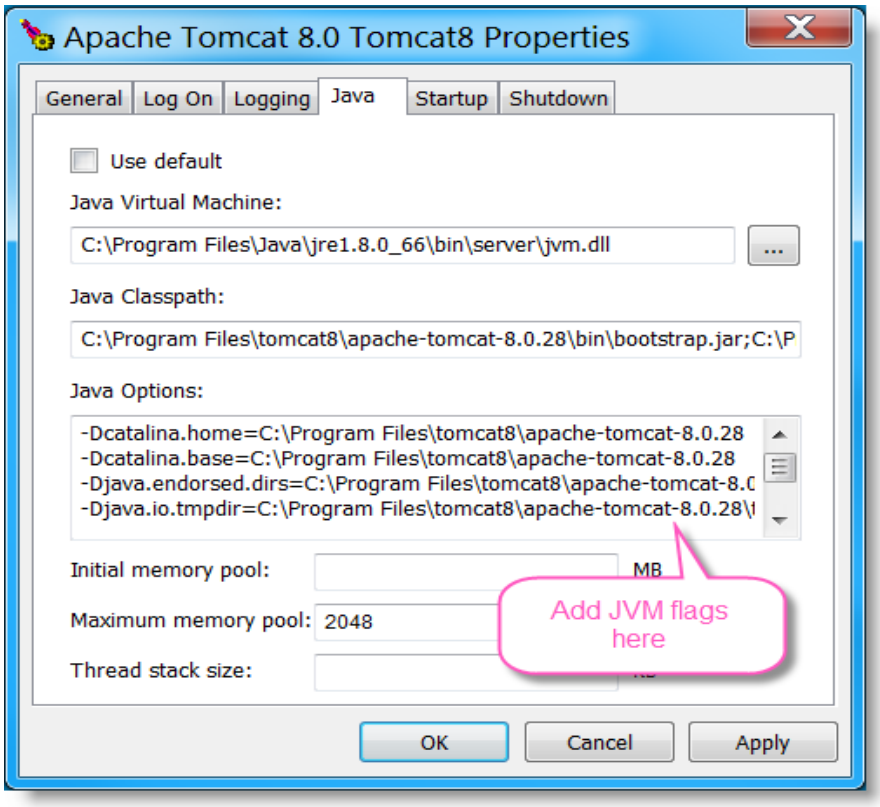
2. Use the environment variable $CATALINA_OPTS to pass in the JVM flags described above to the Tomcat start-up script. The $CATALINA_HOME environment variable in the example refers to the folder into which Apache Tomcat has been installed.

Under Windows, when Tomcat is installed as an *application*, the JVM flags can be set by creating the file `$CATALINA_HOME\bin\-setenv.bat` and placing a line of text similar to this in it:

```
set "CATALINA_OPTS = -server -d64 -Xmx4096m -XX:MaxPermSize=256m -XX:CompileThreshold=8000
-XX:-DisableExplicitGC -Djava.awt.headless=true"
```

The `-d64` flag assumes 64-bit software and the double-quotes enclose *everything* except the `set` command word.

Under Windows, when Tomcat is installed as a *service*, the JVM flags can be set by using the administrative tool installed with Tomcat (version number varies): `$CATALINA_HOME\bin\tomcat8w.exe`:



In the administrative tool, select the Java tab and add the JVM flags to the Java Options, as shown above.

Queries That Use Square Brackets with Tomcat 8.5+

Tomcat 8.5+will reject queries that include square brackets [ ] unless the following highlighted code is added to `$CATALINA_HOME\conf\server.xml`:

```
<Connector port="8080" protocol="HTTP/1.1"
 connectionTimeout="20000"
relaxedQueryChars="[]"
redirectPort="8443" />
```

EC Crypto Issue with Tomcat 8.5 and JDK 1.8.0.181 for SQL Server 2008

If you're using Apache-Tomcat 8.5 and JDK 1.8.0.181 under Windows and attempting to connect to SQL Server 2008, you'll encounter a security error. To remedy this issue:

1. Use File Explorer to find and edit `C:\Program Files\Java\jre1.8.0_181\lib\security\java.security`
2. On or about line #650, look for this property:

   ```
   jdk.tls.disabledAlgorithms=SSLv3, RC4, DES, MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC
   ```
3. Delete the value `3DES_EDE_CBC` and the preceeding comma. Save the file.

Special Connection Pooling for Oracle in Connection-Intensive Environments

Logi applications may quickly open and close dozens of database connections and, in some connection-intensive environments, this may cause server resource problems. The following configuration changes for Tomcat implement connection pooling and have been tested with our Connection.JDBC and Connection.Oracle elements:

For server-wide connection pooling, change these files:

1. Add this to the file `$CATALINA_HOME/conf/web.xml`:

```
<resource-ref>
<description> DB Connection Pooling</description>
<res-ref-name> jdbc/Oraclecp</res-ref-name>
<res-type> javax.sql.DataSource</res-type>
<res-auth> Container</res-auth>
</resource-ref>
```

2. Create a new file `$CATALINA_HOME/conf/context.xml`, containing the following (note variation for Apache 8.0):

```
<Context>
 <Resource name="jdbc/Oraclecp" auth="Container"
type="javax.sql.DataSource" removeAbandoned="true"
logAbandoned="false"
maxActive="20" maxIdle="3" maxWait="10000"
removeAbandonedTimeout="30"
username= "<yourUsername>"
password="<yourPassword>"
factory="org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory" (Apache 5.5 - 7.x)
factory="org.apache.tomcat.dbcp.dbcp2.BasicDataSourceFactory" (Apache 8.x)
driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@<yourServerName>:1521/<yourSID>"/>
</Context>
```

3. Copy the "ojdbc8.jar" file from *<LogiAppFolder>*`/WEB-INF/lib` to `$CATALINA_HOME/common/lib`

4. Manually edit your Connection element attributes in _Settings.lgx (using the Source tab in Studio's Workspace) to be:

```
Connection String = jndi-datasource-name=jdbc/Oraclecp
```

These links provide additional information about connection pooling:

https://people.apache.org/~fhanik/jdbc-pool/jdbc-pool.html
https://tomcat.apache.org/tomcat-8.5-doc/jndi-datasource-examples-howto.html

JBoss 7.1

The following configuration details apply for JBoss 7. *We do not recommend using JBoss 6.*

1. Auto-deploy is used with this server so either manually or through Logi Studio's New Application wizard, give the root folder for your Logi application a .war file extension. Ensure that this folder is placed beneath whatever folder you've configured as your auto-deploy folder, for example: `jboss/server/default/deploy/myLogiApp.war`

 See the Hello World! Tutorial for some examples of working with the New Application wizard.

2. Set the Java options by editing `$JBOSS_HOME/bin/run.conf` so that it includes:

```
-Xmx4096m
-XX:CompileThreshold=8000
 -Djava.awt.headless=true
```

3. Edit `<LogiAppFolder>/WEB-INF/web.xml` and add the following:

```
<context-param>
 <param-name>org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL</param-name>
 <param-value>true</param-value>
</context-param>
```

4. If your JBoss server isn't capable of dynamic deployment, or you're not sure about the server settings, restart your JBoss server, using `bin/standalone.sh`.

5. Use a specific port and parameter when browsing the Logi applications. For example:

```
http://localhost:port#/yourLogiApp/rdpage.aspx?rdReport=Default
```

and be sure to use the correct case-sensitive spelling of the Logi report definition file name, e.g. `Default` vs `default`.

JBoss 5.1

The following configuration details apply for JBoss 5.1:

1. Auto-deploy is used with this server so either manually or through Logi Studio's New Application wizard, give the root folder for your Logi application a .war file extension. Ensure that this folder is placed beneath whatever folder you've configured as your auto-deploy folder, for example: `jboss/server/default/deploy/myLogiApp.war`

 See the Hello World! Tutorial for some examples of working with the New Application wizard.

2. Set the Java options by editing `$JBOSS_HOME/bin/run.conf`so that it includes:

```
-Xmx4096m
-XX:CompileThreshold=8000
 -Djava.awt.headless=true
```

2. Edit *<LogiAppFolder>*`/WEB-INF/web.xml`:

Add the following:

```
<context-param>
 <param-name>org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL</param-name>
 <param-value>true</param-value>
</context-param>
```

Replace the following:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee web-app_3_0.xsd" version="3.0" id="WebApp_ID">
```

with:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://-java.sun.com/dtd/web-app_2_3.dtd"[]>
<web-app id="WebApp_ID">
```

Delete the following:

```
<cookie-config>
 <http-only>true</http-only>
 </cookie-config>
```

3. Delete the following .jar files from the *<LogiAppFolder>*/WEB-INF/lib folder prior to deploying:

   ```
   xml-apis-1.3.04.jar
   xml-apis-ext-1.304.jar
   xercesImpl-2.7.1.jar
   xalan-2.7.1.jar
   ```

4. Download the following files and save them to *<LogiAppFolder>*/WEB-INF/lib:

```
pd4ml.jar
ss_css2.jar
xalan-2.7.0.jar
```

JBoss 4

The following configuration details apply for JBoss 4:

1. Delete these files:
   <LogiAppFolder>/WEB-INF/lib/log4j-1.2.8.jar
   <LogiAppFolder>/WEB-INF/classes/log4j.properties

2. Auto-deploy is used with this server so either manually or through Logi Studio's New Application wizard, give the root folder for your Logi application a .war file extension. Ensure that this folder is placed beneath whatever folder you've configured as your auto-deployfolder, for example: `jboss/server/default/deploy/myLogiApp.war`

 See the Hello World! Tutorial for some examples of working with the New Application wizard.

3. The startup batch file "run.bat" may include the following line: `set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m`. This must be prevented from overriding our required Java Options environment settings (see "Setting Java Options" on page 349). Users can either modify the batch file or the environment variable.

4. JBoss ships with a xerces library that conflicts with Logi's implementation of it, so remove the following .jar files from the *<LogiAppFolder>*/WEB-INF/lib folder prior to deploying:

   xml-apis-1.3.04.jar
   xml-apis-ext-1.304.jar
   xercesImpl-2.7.1.jar

Glassfish 4.1

This topic demonstrates how to configure Glassfish 4.1.

1. Deploy your Logi application folder into `glassfish4/glassfish/domains/domain1/autodeploy`.
2. Use the Glassfish Admin Console at `http://<yourServer>:4848/common/index.jsf` to configure the Java options.
3. In addition to the recommended options listed in "Setting Java Options" on page 349, set the `-XX:MaxPermSize` setting to a minimum of `300m`.
4. Click this link glassfish-web.xml to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as `<LogiAppFolder>/WEB-INF/glassfish-web.xml`.
5. Click this link faces-config.xml to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as `<LogiAppFolder>/WEB-INF/faces-config.xml`.
6. You may see warnnig in the Glassfish log about JDBC4 drivers. If desired, these warning can be suppressed by moving the .jar files identified in the log entries to `glassfish4/glassfish/domains/domain1/lib/ext`.

Glassfish 3.0 / 3.1 / 4.0

This topic demonstrates how to configure Glassfish 3.0 / 3.1 / 4.0.

1. (Glassfish 3.x only) Manually add a .war extension to the Logi application folder. Deploy your application manually using the Glassfish Admin Console.

2. In addition to the recommended options listed in "Setting Java Options" on page 349, set the `-XX:MaxPermSize` setting to a minimum of `300m`.

3. Click this link sun-web.xml to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as `<LogiAppFolder>/WEB-INF/sun-web.xml`.

4. Click this link faces-config.xml to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as`<LogiAppFolder>/WEB-INF/faces-config.xml`.

Glassfish 2.1

This topic demonstrates how to configure Glassfish 2.1.

1. Manually add a .war extension to the Logi application folder. Deploy your application manually using the Glassfish Admin Console.

2. Click this link sun-web.xml to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as `<LogiAppFolder>/WEB-INF/sun-web.xml`.

WebLogic 12c

This topic demonstrates how to configure WebLogic 12c:

1. Auto-deploy is used with this server. Logi application folders must be manually given a .war file extension.

2. The JVM settings described above can be changed by editing them in

   `$WL_HOME/user_projects/domains/<Logi_Domain>/bin/setDomainEnv.sh (Linux/UNIX)`

   `$WL_HOME/user_projects/domains/<Logi_Domain>/bin/setDomainEnv.cmd (Windows)`

3. Click this link `weblogic.xml` to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as *<LogiAppFolder>*`/WEB-INF/weblogic.xml`.

4. If using requests that return a large amount of data, performance can be improved by configuring the "TCP Chunk" parameters. You must pass `-D weblogic.Chunksize=65535` for Logi to work properly under Weblogic. See the following information: *Tune the Chunk Parameters*

WebLogic 10

This topic demonstrates how to configure WebLogic 10:

1. Auto-deploy is used with this server. Logi application folders must be manually given a .war file extension.

2. The JVM settings described above can be changed by editing them in

   `$WL_HOME/user_projects/domains/<Logi_Domain>/bin/setDomainEnv.sh (Linux/UNIX)`

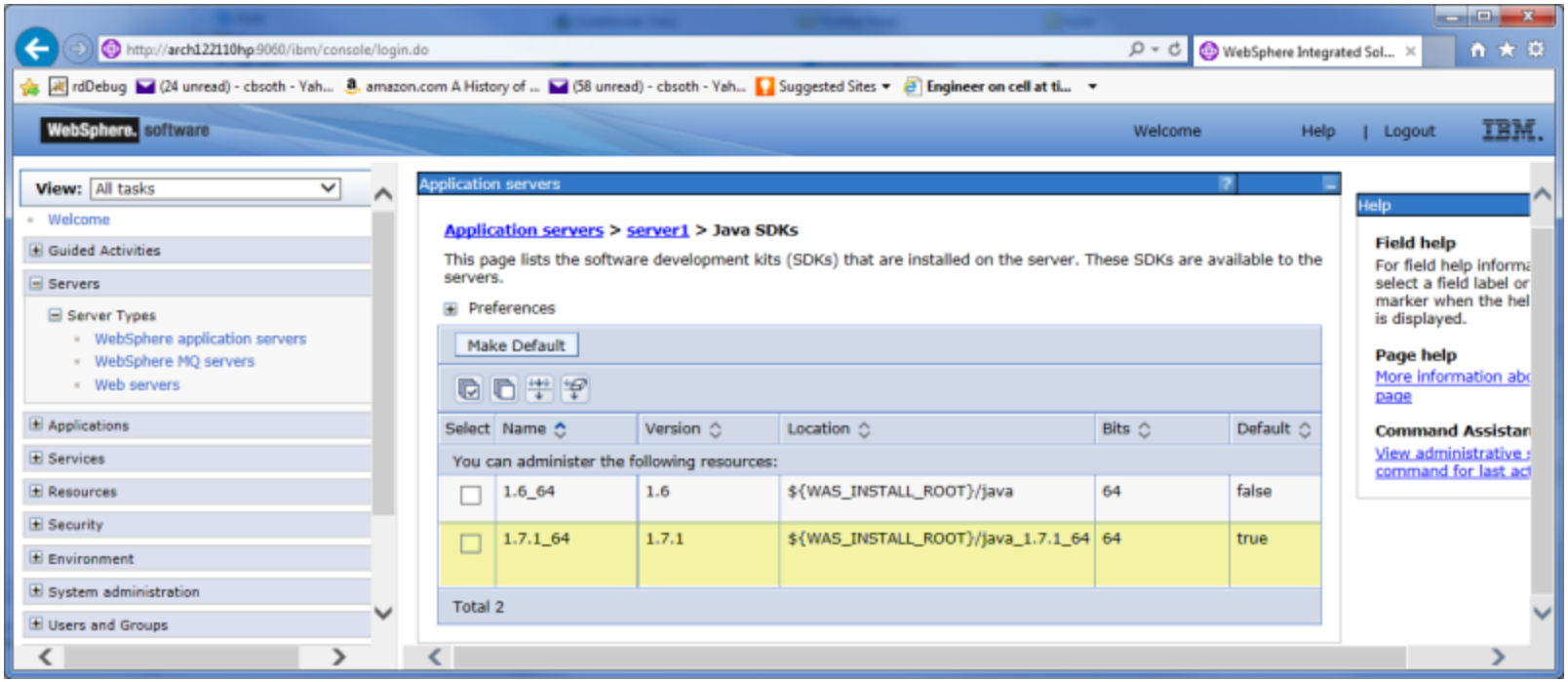
   `$WL_HOME/user_projects/domains/<Logi_Domain>/bin/setDomainEnv.cmd (Windows)`

3. WebLogic 10 - 10.2: Click this link `weblogic10.xml` to open it in you browser. WebLogic 10.3: Click this link `weblogic103.xml` to open it in your browser. Right-click and select "View page source" (or similar) to see a plain text version of the code. Save this code as *`<LogiAppFolder>`*`/WEB-INF/weblogic.xml`.

4. If using requests that return a large amount of data, performance can be improved by configuring the "TCP Chunk" parameters. You must pass `-D weblogic.Chunksize=65535` for Logi to work properly under Weblogic. See the following information: *Tune the Chunk Parameters*

Websphere 7 / 8.5

For best performance, Websphere 8.5 users should ensure that they're using SDK 7. Information about the SDK can be found here and it can be downloaded using the IBM Installation Manager from here.



WildFly 8 / 9 / 10

JBoss has been rebranded as WildFly. It *requires* JDK 1.8+ or OpenJDK 8+.

For Widfly 8 and 9, use Step 1 below to deploy and use JBoss 7: Steps 2-5 to configure.

The following configuration details apply for WildFly 10:

1. Auto-deploy is used with this server so either manually or through Logi Studio's New Application wizard, give the root folder for your Logi application a `.war` file extension. Ensure that this folder is placed inwhatever folder you've configured as your auto-deploy folder, for example: `wildfly-10.0.0.Final/standalone/deployments/myLogiApp.war`. In addition, create an empty marker (text) file named `myLogiApp.war.dodeploy` in the same folder.

2. Set the Java options by editing `$JBOSS_HOME/bin/run.conf` so that it includes:

```
-Xmx4096m
-XX:CompileThreshold=8000
 -Djava.awt.headless=true
```

And run `$JBOSS_HOME/bin/standalone.conf.bat`.

3. Edit `<LogiAppFolder>/WEB-INF/web.xml` and add the following:

```
<context-param>
 <param-name>org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL</param-name>
```

```
 <param-value>true</param-value>
</context-param>
```

4. Edit the `deployment-scanner` attribute in `$JBOSS_HOME/standalone/configuration/standalone.xml` to be:

```
 <deployment-scanner scan-interval="5000" relative-to="jboss.server.base.dir" path="deployments"
 auto-deploy-zipped="true" auto-deploy-exploded="false"/>
```

5. If your WildFly server isn't capable of dynamic deployment, or you're not sure about the server settings, restart your WildFly server, using `bin/standalone.sh`.

6. Use a specific port and parameter when browsing the Logi applications. For example:

```
http://localhost:port#/yourLogiApp/rdpage.aspx?rdReport=Default
```

and be sure to use the correct case-sensitive spelling of the Logi report definition file name, e.g. `Default` vs `default`.

Logi Product Upgrades

Each new release of Logi Analytics products provides feature enhancements and improvements. Customers are encouraged to take advantage of these improvements by *upgrading* to the latest releases when they become available.

The following topics provide guidance for the process of upgrading, modifying, and uninstalling your Logi product installations, and managing the related versions of your Logi applications:

- General Requirements
- Upgrade Impacts
- Upgrading the Products
- Changing .NET Application Versions
- Changing Java Application Versions
- Creating Application Pools in IIS 7+
- Creating Application Pools in IIS 6

Information about which Logi Info and add-on module versions we recommend using together is available in Release Pairings. For more information about Logi Info releases, see Release Notes.

The installation tool, InstallShield, *does not remove* any files or folders created or modified after the initial installation, including Sample Applications provided with Logi products that you may have modified.

General Requirements

This topic covers the general requirements for upgrading Logi products:

- Logi products for the Windows environment and Logi Studio require the .NET Framework 4.x. If not already in place, with your consent, appropriate versions of the .NET Framework are installed when Logi products are installed. They are also available for free from the Microsoft Download Center.

  Microsoft ended support for .NET Framework 4.0 and 4.5 in January 2016. We recommend use of .NET 4.6.
- Separate installations of different versions of Logi products and applications can co-exist as long as they are installed in different folders.
- IIS users need to implement separate Application Pools to isolate Logi applications using different .NET Framework versions. Application Pools are a standard feature of IIS 7+ and can be implemented in IIS 6 using the instructions provided later in this topic.

- Logi v12 Java applications require the Oracle JDK or OpenJDK 8 or (Info v12.6 SP2+) 11, 12, 13, 14

   Oracle has changed its Java usage policies. See "Java Usage Policy" on page 342 for important information.

Upgrade Impacts

 Logi v12 products include several changes you should know about *before* upgrading:

- Logi v12 products include stylistic and appearance improvements in charts, super elements, themes, etc. which are noticeably different. You may care to do a test upgrade before committing to a comprehensive upgrade.
- The built-in Themes shipped with Logi v12.1+ have been changed internally to allow customization with the Theme Editor tool. Custom Themes you may have created based on earlier versions of the built-in Themes *will continue to work* correctly in v12.1+ but may not be editable in the Theme Editor. To make them so, you will have to recreate your customizations starting with one of the new built-in Themes.
- You *must* use Application Pools that use .NET 4.x. If such an application pool doesn't exist, Studio will generate a warning when you upgrade an existing Logi application. When you create a new application, Studio will create a new application pool, named "Logi Info .Net v4.x", and will assign the application to it.
- Logi Info includes a free, 15-day trial license but, when it expires, you will need to purchase a regular license. If you're a customer in good standing, Logi Customer Service will assign that license to you in advance, but you will need to go to DevNet and download the license file.

Upgrading the Products

Upgrading to a new version involves two steps: installing the new release of Logi Info, and upgrading the version of any existing Logi applications. You can do the former without doing the latter; applications are not automatically upgraded. Original and upgraded applications can co-exist, so upgrading can be phased in as desired.

Getting an Upgrade

How do you get an upgrade? Customers who have **purchased a maintenance plan** for Logi Info, or are using it under an OEM agreement, may be able to download new product releases directly from our DevNet web site. See Support → Product Download.

If the download links are not available to you, contact Customer Service for alternate instructions for downloading upgrades. Customer Service can also answer questions you may have about purchasing or renewing a maintenance plan.

To upgrade your Logi product, run the downloaded installation program and install the new version into the same location as the previous version, or to a different location if you wish to have both versions installed. No un-installation of the original version is required, and existing definitions or data *will not* be overwritten. The upgrade installation will take 2-3 minutes to complete.

Evaluations

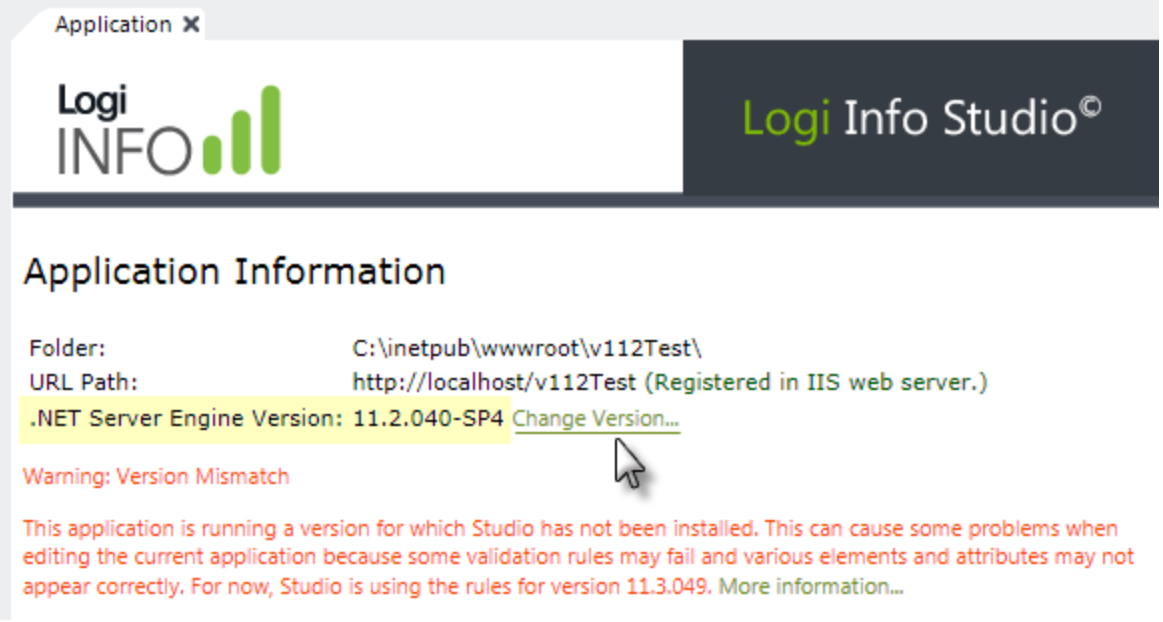
If you wish to install an upgrade for **evaluation** purposes, we recommend that you install it into a *different folder* than any previous version, such as `C:\Program Files\LogiXML IES Dev Test`. This will preserve the previous version of Studio. If you install an update into the same folder as a previous version, Studio will be updated and you will not be able to roll it back without uninstalling and re-installing the previous version entirely.

Backup Scheduler Data

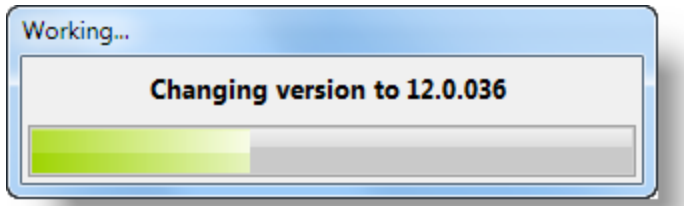
If you have previously installed and used the **Logi Scheduler** service and are not using the optional storage configuration to store your scheduled tasks on a networked database server, we recommend that you make a safety copy of your existing schedule data, before upgrading: `(Windows .NET) C:\Program Files\LogiXML IES Dev\LogiXML Scheduler Service\Schedules.vdb3` `(Windows Java) C:\Program Files\LogiXML IES Dev\LogiXML Scheduler Service Java\Schedules\*.*` `(Linux/UNIX) <installFolder>/Schedules/*.*` and then proceed with the upgrade installation. Your existing data file *should* be upgraded in place without any difficulty.

Changing .NET Application Versions

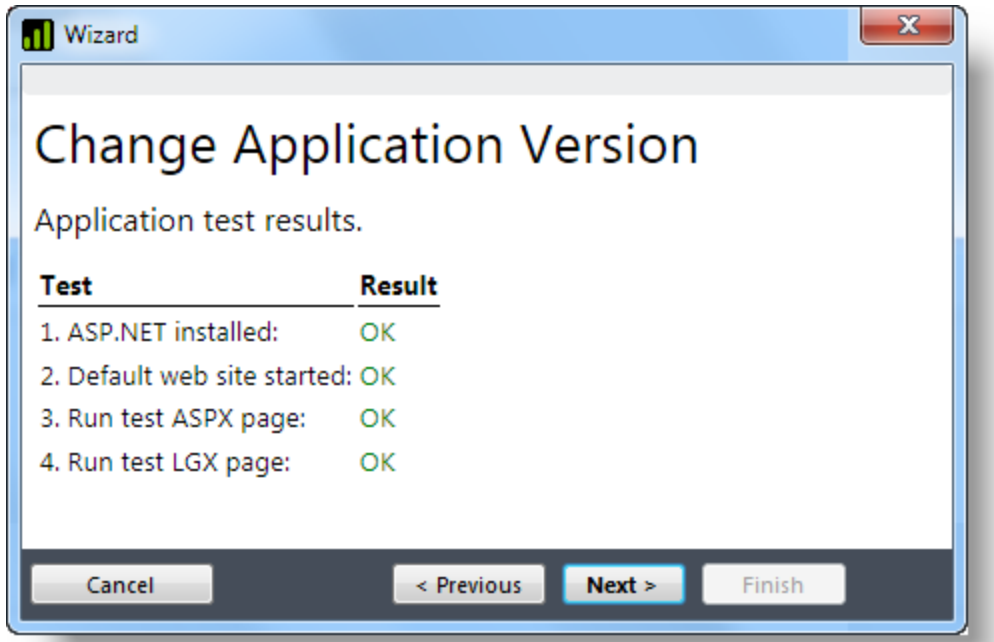
In order to use all of the features in the latest Logi Info release, after you install it you will need to change the application version on all of your Logi applications. There are two methods for changing an application version:



1. Using Logi Studio - When you open an application, Studio will display a warning in its Application tab, as shown above, if the application's version does not match the version of the Logi product being used. Click the **Change Version...** link to replace your Logi application's version-specific binary files with the files for a version that you select from a list of installed versions. *This will not affect any definitions or support files*.



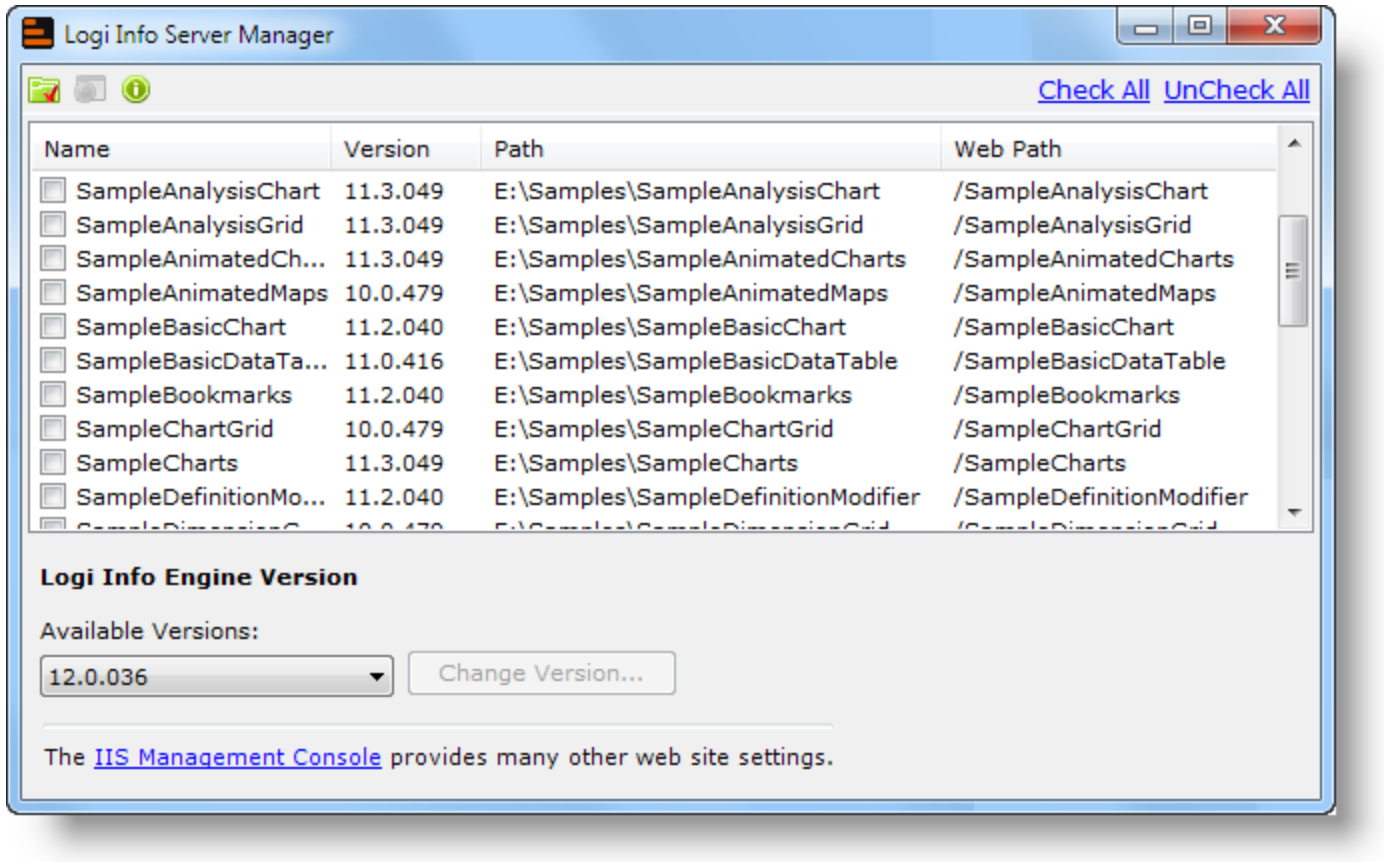
A progress indicator, like the one shown above, will be displayed.



Changing an application version will also trigger some web server diagnostics to run after the upgrade and you'll see some information about that process. Once the tests end, you'll the results, as shown above.

The failure of any test indicates that something is wrong with your web server environment's ability to run the upgraded application. This could be caused by a number of things, such as failing to install the new Logi version as an Administrator, or having incorrect file permissions on the application folder, or failing to have the correct version of .NET installed. Contact Logi Support for assistance.

2. Using Server Manager - You can also change application versions using Server Manager, which is installed with Studio and is available via Studio's Tools menu or the Start Menu. It's a tool that allows you to examine all of the Logi applications on the web server and manage them individually or in a batch.



Server Manager, shown above, only works with .NET applications, using the IIS or Cassini web servers; if neither of these servers is installed, Server Manager will prompt you to install them. The Server Manager dialog box displays a table showing each application installed on the local web server and a list of all the Logi product versions installed on the machine. To change an application's version, check one or more applications, select the version desired, and click **Change Version**. The process usually takes less than a minute for each application selected. Versions can be upgraded or downgraded.

Changing Java Application Versions

In order to use all of the features in the latest Logi Info release, after you install it you will need to change the application version on all of your Logi applications.

The process for upgrading a Logi Java application on your development machine is the same as it is for .NET apps: use the **Change Version** link in Studio as discussed in "Changing .NET Application Versions" on page 377.

The Server Manager tool discussed in "Changing .NET Application Versions" on page 377 is intended for use *only* with .NET applications, and expects to work with the IIS or Cassini web servers. You cannot use Server Manager for Java applications.

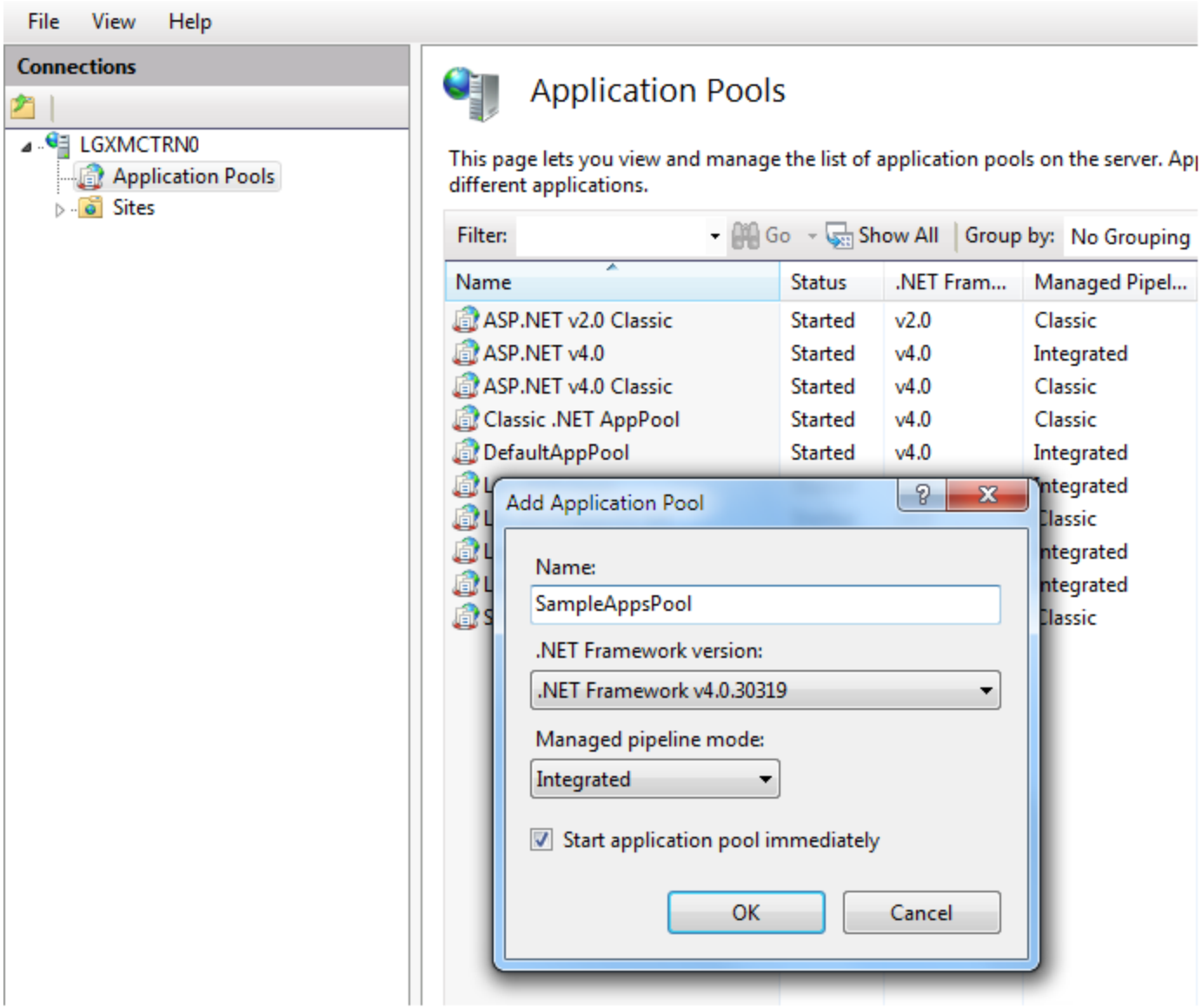
Changing your Java application version on a Linux/Unix production server (which won't have Logi Studio or Server Manager installed as they are Windows apps) is a little more complicated. You could just copy the application files and folders from the development machine to the production machine but that will likely result in an undesirable proliferation of .jar files over time. Instead, we recommend the following:

1. Change the version of your Logi app on your development machine using Studio or Server Manager.
2. On the production server, create a new folder which will become your new Logi app folder.
3. Copy *everything* from the app folder on your development machine (which is the new version) to this new folder.
4. Copy any files specific to the production app, such as _Settings.lgx, which might have different connection settings, from the original Logi app folder on the production server to the new app folder on the production server.
5. Rename the old and new app folders, or adjust any references so the new app folder is now the recognized location for your Logi app.
6. You may need to change the General element's Application Path attribute in _Settings.lgx if the path or folder name have changed.
7. Test your application.
8. Archive or delete the old app folder on the production server.

Creating Application Pools in IIS 7+

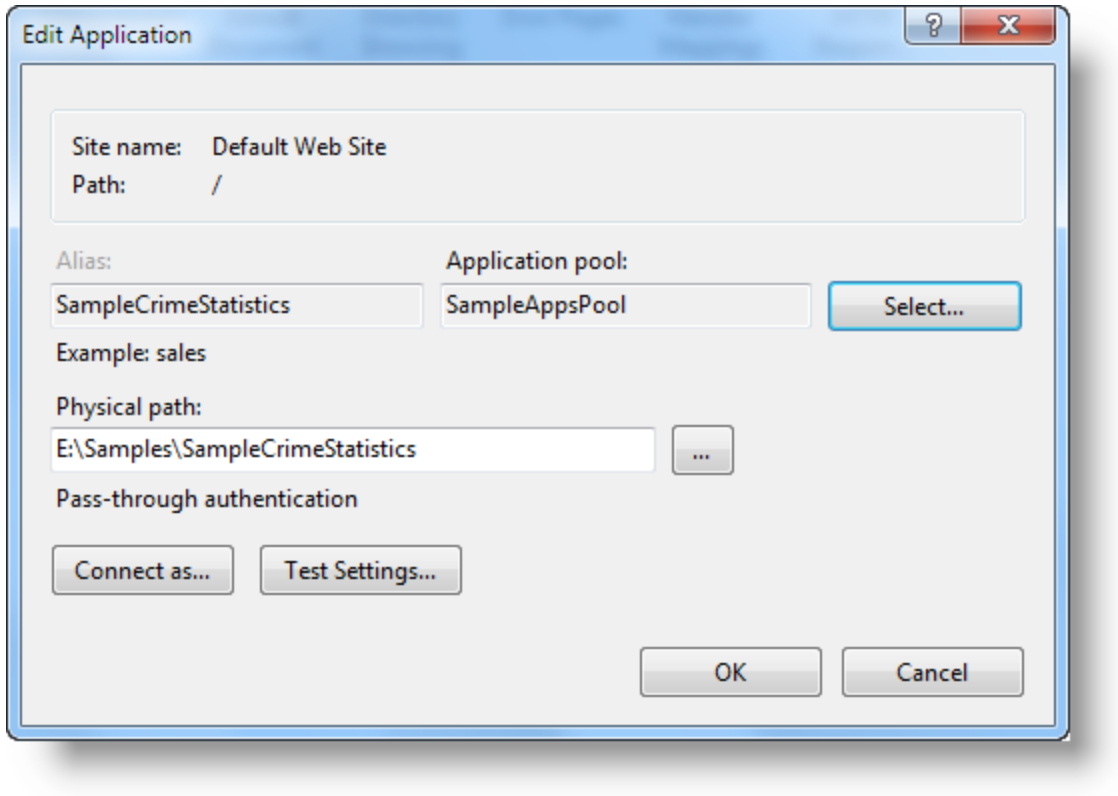
The **Application Pool** is a method of isolating different .NET applications within the web server, in order to provide improved reliability. The default application pool, "DefaultAppPool" is created at installation and all virtual directory applications, when created, are assigned to this default pool.

However, applications using different .NET versions can't be run at the same time in the same pool. If they are, error messages will appear for whichever application is started last. The recommended practice is to create a new application pool and assign all applications that use the same .NET version to it.



To create a new application pool:

1. Open the IIS Manager (see All Programs → Administrative Tools)
2. Select and right-click the **Applications Pools** item in the list on the left. Select **Add Application Pool...** from the pop-up menu.
3. Enter the new name of your choice as the Application Pool ID and leave the default pool settings selected. Click **OK**
4. Right-click the Application Pools item in the list on the left and click **Refresh** to see your new pool.



To assign an application to the new Application Pool:

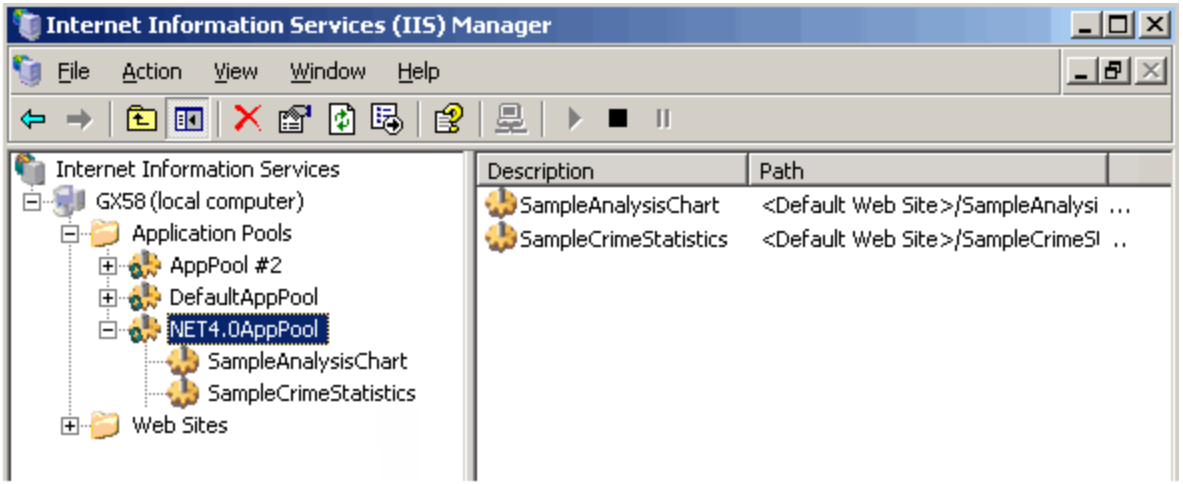
1. Expand the **Sites** item in the list on the left, then expand the **Default Web Site** item below it.
2. Find and select your application in the list.
3. In the right-hand Actions Panel, click **Basic Settings...**
4. A dialog box like the one shown above will appear. Click **Select...** and choose your new Application Pool from the list.
5. Click **OK**.

Your application is now assigned to your new Application Pool.

Creating Application Pools in IIS 6

Microsoft IIS 6 introduced **Application Pools** as a method of isolating different applications within the server, in order to provide improved reliability. The "Default Application Pool" is created by default at installation and all virtual directory applications, when created, are assigned to this default pool.

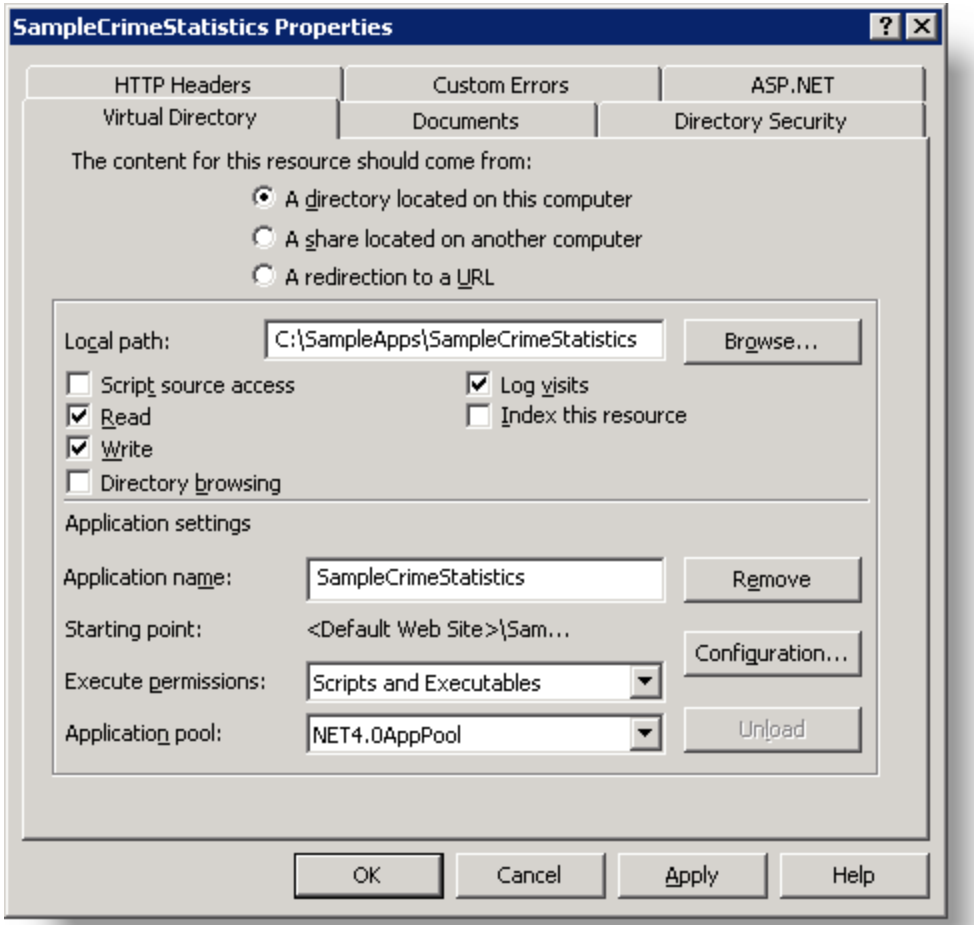
However, applications using different .NET versions can't be run at the same time in the same pool. If they are, error messages will appear for whichever application is started last. The recommended practice is to create a new application pool and assign all applications that use the same .NET version to it.



To create a new application pool:

1. Open the IIS Manager (see All Programs → Administrative Tools)
2. Select and right-click the **Applications Pools** item in the list on the left. Select **New Application Pool...** from the pop-up menu.
3. Enter the new name of your choice as the Application Pool ID and leave the default pool settings selected. Click **OK**

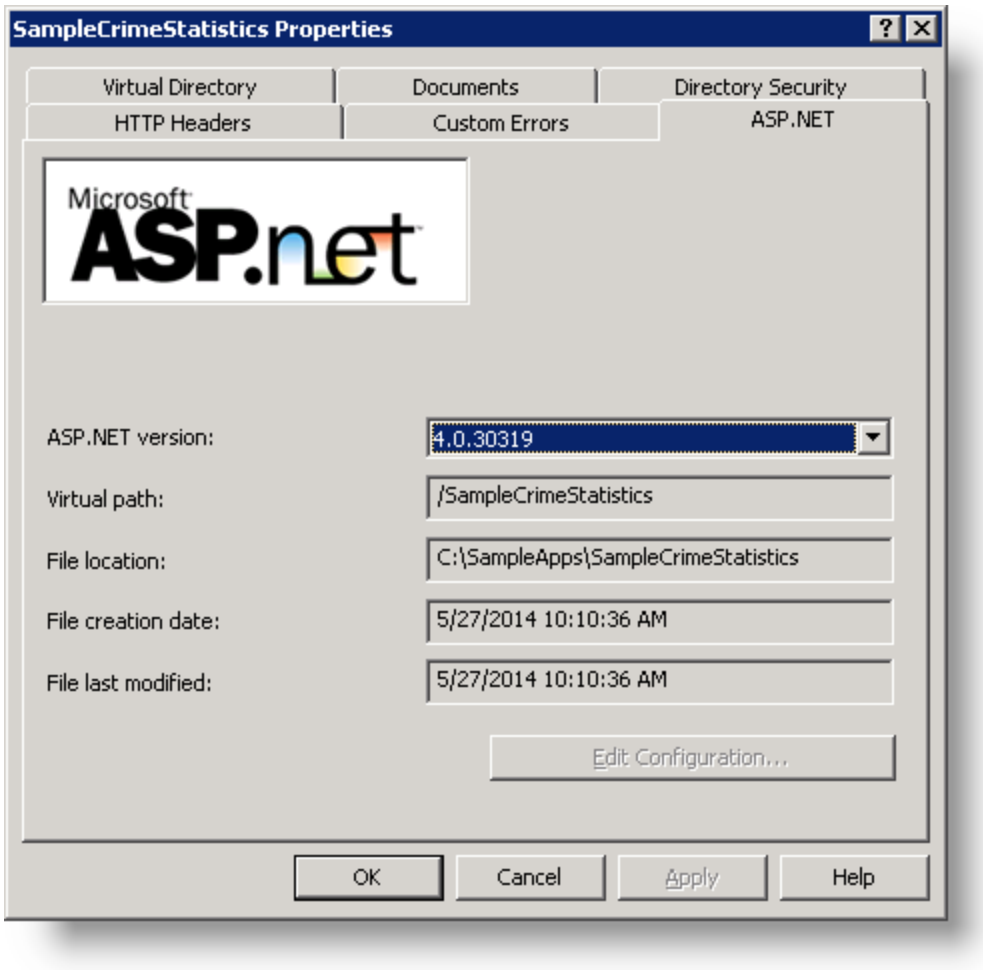
4. Expand the Application Pools item in the list on the left to see your new pool.



To assign an application to the new Application Pool:

1. Expand the **Web Sites** item in the list on the left.
2. Select and right-click the virtual directory representing your application. Select **Properties** from the popup menu.
3. Select the new application pool you just created from the list of Application Pools in the Properties dialog box.
4. Click **OK**

If you have any doubt about which version of .NET your application is set to use:



Click the **ASP.NET** tab on the Properties dialog box and inspect the ASP.NET version selection.

Glossary

A

API

API, short for Application Program Interface, is a set of routines, protocols, and tools for building software applications. In business intelligence, APIs may be used to enable end-users to directly update source systems.

Authentication

Authentication is the verification of a user's identity.

Authorization

After a user's identity has been authenticated, authorization grants or denies access to reports, columns, and records to selected users or user-groups.

B

Big Data

Refers to both the ever-growing volumes of data in use today and also to services that are specifically engineered to provide and manipulate very large data volumes.

Business Analytics

Business analytics, or business intelligence (BI), gives customers the ability to rapidly create scalable, interactive data analysis applications, and self-service capabilities users can access from anywhere and on any device.

C

Columnar Data Store

Columnar data store is a type of big data repository containing structured data in columns and rows. The main benefits are that the data can be highly compressed and is easily searchable.

CRM

A Customer Relationship Management (CRM) system is a database-based system that records a company’s daily customer-related transactions. CRMs can help customer representatives to provide better service, close more deals, and increase revenue.

CSS

Cascading Style Sheets (CSS) is a technology that allows the presentation aspects of web pages to be separated from the page content. It can be used to add "styling" (e.g. apply fonts, colors, alignment, spacing, and more) to web pages.

D

Data Discovery

Data discovery is the capability to analyze data on-the-fly and uncover insights from it.

Data Enrichment

Data enrichment is a method of preparing data to make it ready for analysis and exploitation, and can include formatting, adding calculations, joining with other data, and more.

DevNet

The Logi Developer Network website.

Drill Down

Drill Down is a capability that allows the user to get a view of the underlying or supporting data used in an analysis.

Drill Through

Drill Through is similar to Drill Down but takes it one step further by applying analysis to the underlying or supporting data.

E

Elemental Development

A development approach used in Logi Info that lets developers build feature-rich applications by using reusable, pre-built elements, rather than by writing low-level code.

F

Forecasting

A technique involving data mining and analysis leading to predictions about what will happen in the future.

G

Geo Mapping

The combination of geographic and other data to produce map visualizations, such as Google or Leaflet maps.

H

Heatmap

A Heatmap chart, sometimes called a "tree map", which uses a unique arrangement of rectangles to represent data and relationships, using color and size.

I

Interpolation

The process of evaluating a literal value match containing one or more placeholders, yielding a result in which the placeholders are replaced with their corresponding values.

J

JavaScript

JavaScript is a programming language supported by the majority of modern web browsers and used by many websites.

JDBC

Java Database Connectivity (JDBC) is an API used to access relational databases. Open Database Connectivity (ODBC) is a similar API designed for use with Java.

JSON

JavaScript Object Notation (JSON) is a lightweight data-interchange format that's easy for humans to read and write, and easy for computers to parse and generate.

K

KPI

Key Performance Indicators (KPIs) are visual indicators, in the form of color-coded shapes, which are tied to a pre-defined, critical threshold.

L

LDAP

The Lightweight Directory Access Protocol (LDAP) is an Internet protocol applications use to look up information from a server and is frequently used for containing user login information.

M

My Term

My definition

N

NoSQL

"Not only SQL" (NoSQL) is an alternative to traditional relational databases, and doesn't rely on tables and a pre-determined schema. NoSQL databases are especially useful for working with large sets of distributed data.

O

ODBC

Open Database Connectivity (ODBC) is an API used to access relational databases. Java Database Connectivity (JDBC) is a similar API designed for use with Java.

OLAP

Online Analytical Processing (OLAP) is the process of analyzing data stored in multi-dimensional "cubes".

R

REST

Representational State Transfer (REST) is a type of API used to provide interoperability between computer systems on the Internet.

S

SSM

The Self-Service Module (SSM) is a package that includes Logi Info + SSRM + Discovery or Logi Platform Services.

SSRM

The Self-Service Reporting Module (SSRM) is a Logi Info add-on module that adds special elements to Info and includes the InfoGo application.

W

Write-Back

The ability to update data sources, typically by adding, editing, or deleting data.