

Intelligence Center

Platform Administrator



CONTENTS

1. The Intelligent Enterprise	Introducing the Intelligent Enterprise	8
	The Intelligence Center.....	9
	Your role: the Platform Administrator	10
	Your administration team	11
	Platform Administrator's experience and qualifications	11
	Exercise 1.1: Deploying the MicroStrategy platform.....	13
 2. Standardizing the Intelligent Platform		
	Coordinate with other Intelligence Center users	17
	Example: Communicate object migration schedule.....	17
	Create a documentation repository.....	18
	Intelligent Enterprise standards documentation	19
	Artifacts created for each environment	21
	Establish a training schedule	21
	Exercise 2.1 Discuss the value of communication in the Intelligent Enterprise	23
	Standardizing requirements gathering	23
	Establish naming conventions	26
 3. Platform Administration Planning		
	Platform architecture	29
	Exercise 3.1: Review of MicroStrategy analytics architecture.....	32
	Exercise 3.1 Solution: Review of MicroStrategy analytics	

architecture	33
MicroStrategy platform deployment options.....	34
Enterprise data center and public cloud deployments	35
MicroStrategy Cloud Platform deployment	36
MicroStrategy Cloud Environment: Fully managed option.....	37
Choosing the right cloud option	38
Platform architecture best practices	38
Exercise 3.2: Explore the cloud environment.....	42
Exercise 3.3: Access applications on the Linux machine.....	44
Exercise 3.4: Deploy MicroStrategy Web Universal.....	52
Enterprise platform management	55
Exercise 3.5: Create a self service application	57
Exercise 3.6: Connect and import data from an external data warehouse for your project users.....	60

4. Platform Configuration

Configuring connectivity.....	66
Best practices for creating data connectors	68
Exercise 4.1: Automate platform-warehouse connectivity	70
Implementing platform security.....	80
Managing user authentication	80
Managing user security.....	88
Exercise 4.2: Use connection mapping to secure data access	96
Platform security best practices.....	101
Configuring Distribution Services.....	109
Distribution Services best practices	110
Exercise 4.3: Configure Distribution Services error handling	115
Exercise 4.4: Configure Distribution Services components	118
Exercise 4.5: Configure a cache update subscription.....	124
Exercise 4.6: Automate subscription execution	128
Streamlining your analytics deployment	131
Exercise 4.7: User provisioning	134
Exercise 4.8: Establishing data security	142
Exercise 4.9: Verify content privileges and deployment	145
Exercise 4.10: Monitoring real-time telemetry data.....	149

5. Object Migrations

Managing project lifecycle	150
----------------------------------	-----

Implementing object migrations: Using project releases and change management systems	154
Best practices for object migrations.....	156
Exercise 5.1: Duplicate the MicroStrategy Tutorial project	159
Exercise 5.2: Create an Object Manager package for an Intelligent Cube	161

6. Platform Monitoring

Guide environment monitoring	168
Analyzing real-time data: Platform Analytics.....	171
Platform Analytics architecture and services	174
Exercise 6.1: Accessing Platform Analytics dossiers	176
Exercise 6.2: Enable statistics from Workstation for platform monitoring.....	178
Environment management	179
Exercise 6.3: Monitoring your environment	180
Monitoring upgrades: Testing for discrepancies	182
Testing environment discrepancies due to upgrades: Integrity Manager.....	183
Exercise 6.4: Test the integrity of reports across projects.....	184
Monitoring licenses: Using MicroStrategy License Manager	187
License Manager.....	187
Managing and verifying licenses	190
Managing Named User licenses.....	192
Updating processor capacity: Using Service Manager	195
License Manager best practices.....	197
Exercise 6.4: Perform a License Manager audit	198
Exercise 6.4 Solutions: Perform a License Manager audit	199
Exercise 6.5: Generate a License Manager report.....	201
Exercise 6.5 Solutions: Generate a License Manager report.....	203
Exercise 6.6: Configure the license check time.....	206
Monitoring product usage: Compliance Telemetry Dossier	207
Exercise 6.7: Use the Compliance Telemetry Dossier.....	208

7. Troubleshooting

Supporting the analytics environments: troubleshooting	214
Creating troubleshooting guidelines.....	215
Using predefined log files	216
Exercise 7.1: Analyze a log file	219
Exercise 7.2: Enable logging in Command Manager.....	224
Resolving issues: tools for troubleshooting.....	225

Configuring what is logged: MicroStrategy Diagnostics and Performance Logging tool	225
Exercise 7.3: Enable report SQL tracing.....	226
Exercise 7.4: Review of frequently used diagnostics	230
Exercise 7.4 Solutions: Review of frequently used diagnostics	232
Exercise 7.5: Troubleshoot Diagnostics and Performance Logging tool issues.....	234
Exercise 7.5 Solutions: Troubleshoot Diagnostics and Performance Logging tool issues.....	235
Troubleshooting Intelligence Server issues: Analyzing DSSErrors.log file	237
Core dump and stack trace.....	239
Exercise 7.6: Log a stack trace	241
Troubleshooting MicroStrategy Web server issues: Diagnostics and Statistics features	243
Exercise 7.7: Set up diagnostics.....	244
Analyze system and server performance: Web statistics	245
Exercise 7.8: Set up web statistics	247
Viewing server logs.....	248
Troubleshooting data source issues: using DB Query tool	249
Exercise 7.9: Create a Freeform SQL report.....	250
Exercise 7.10: Troubleshoot an issue using the DB Query tool.....	255

8. Platform Optimizations

Optimizing environments: Reducing computational distance	263
Implementing for performance: Product placement considerations	264
Optimizing systems for performance: Server lifecycle management	267
System optimization: Server requirements	267
Server deployment and upgrades	268
Reviewing upgrade components	268
Optimizing web performance: Tuning MicroStrategy Web and application servers.....	269
Exercise 8.1: Increase Java heap size	270
Tuning Intelligence Server for performance	271
Performance influencers: Caching	272
Exercise 8.2: Configure subscription caching to History List.....	279
Performance influences: Intelligent Cubes.....	281
Exercise 8.3: Automate cube refresh using Command Manager.....	287
Performance influencers: Scheduling	288

	Performance influencers: History List	290
	Exercise 8.4: Automate deletion of History List messages	292
	Optimizing query performance: Database connections.....	292
	Exercise 8.5: Configure DB connection timeouts.....	299
	Exercise 8.6: Configure connections and job priorities	301
	Exercise 8.7: Set usage limits using governors	314
	Exercise 8.8: Troubleshoot a governing issue.....	316
	Exercise 8.8 Solutions: Troubleshoot a governing issue	317
A. Appendix: Platform administrator Checklist	Platform Administrator description.....	318
	Check list overview	319
	Assess.....	319
	Plan.....	319
	Create	320
	Publish.....	320
	Operate	320
	Optimize	321
	Assets and tooling	321
	Detailed check list	324
	Assess	324
	Environments.....	324
	Users and Projects.....	324
	Project Objects	325
	Subscriptions	325
	Plan.....	325
	Platform Environments Architecture	325
	Platform Data Architecture.....	326
	Platform Project Architecture	326
	Platform Security Architecture.....	327
	Platform Configuration Protocols	327
	Analytics Security Architecture	327
	Create	327
	Users and User Groups	327
	Configure Directory / Identity Integrations	328
	Platform Environments	328
	Distribution Services Subscriptions.....	328
	Cache and Cube Refresh Schedules	328
	Data Connectors within the Platform Environments	329
	Publish.....	329
	Platform Environments Synchronization	329
	Platform Analytics	330

Configuration Documentation.....	330
Updated Upgrade Procedures.....	330
Operational Procedures	330
Operate	331
Monitor Platform Report	331
Support Intelligence Center Architects.....	331
Handle Platform Environment Cases	331
Troubleshoot Platform Issues	332
Coordinate with Intelligence Center.....	332
Provision User Access	332
Upgrade environments.....	332
Optimize	333
Platform Project Performance	333
Platform Environment Performance	333
Platform Services Performance	333
Enterprise Data Set Performance	333
Enterprise Applications Performance	334
Enterprise Mobile Applications Performance	334
Definitions.....	334

THE INTELLIGENT ENTERPRISE

Introducing the Intelligent Enterprise

Your company, MartZon, is a fast-growing retailer that has decided to transform into an Intelligent Enterprise. As MicroStrategy users, MartZon wants to leverage its existing investments and successfully deliver powerful analytics and mobility solutions across the enterprise.

The Intelligent Enterprise is a data-driven organization that effectively designs and implements Business Intelligence (BI) solutions while promoting effective use of data across your enterprise. This fosters growth and development, with a focus on data governance and alignment of strategic business goals to technology investments.

Getting there requires the right tools and structure to balance traditionally counteractive forces—agility and governance, convenience and security, ease of use and enterprise functionality—all critical capabilities that the MicroStrategy platform is positioned to support with its unique intelligence architecture. A successful Intelligent Enterprise:

- Drives adoption and success of enterprise Business Intelligence.
- Coordinates BI implementations.

- Maintains sound data governance and a single version of the truth.
- Provides formal approach to documenting processes, creating content, and ongoing maintenance.
- Ensures that BI is aligned with enterprise strategy.

Along with quick and easy ad-hoc departmental solutions, MicroStrategy has the robust, proven ability to support high-scale deployments and establishing a single source of the truth. MicroStrategy's tools include data-governance features, administrative controls, and management capabilities with the enterprise platform software, all critical to the Intelligent Enterprise.

The Intelligence Center

The Intelligence Center is comprised of a team of expert architects and administrators who define, develop, and provide guidance across the enterprise. With the collective know-how to maximize BI investments, the Intelligence Center drives optimization of system architecture, upgrades, configuration, performance, scalability, and stability.

Intelligence Center

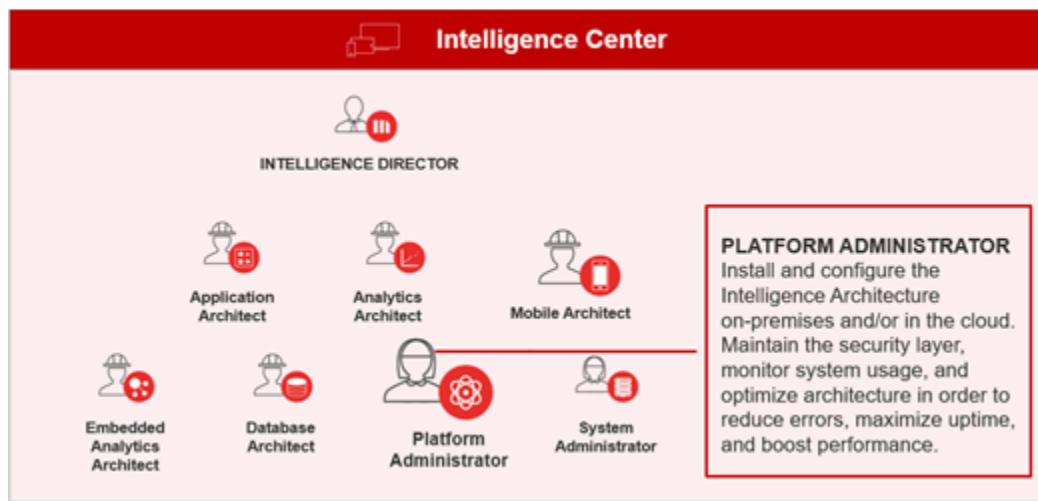
The diagram illustrates the Intelligence Center personas, organized into two main sections: PERSONAS and Intelligence Center Director (ICD).

PERSONAS

- Intelligence Center Director (ICD)**: Create Intelligence environments by deploying the Intelligence Architecture, supervising the Intelligence Center, and running Intelligence Programs to support enterprise and departmental analytics and mobility applications for all constituents.
- Application Architect (APA)**: Create, share, and maintain intelligence applications for the enterprise. Publish standardized application objects and promote departmental applications from self-service into the enterprise environment.
- Analytics Architect (ANA)**: Create, publish, and optimize a federated data layer as the enterprise's single version of the truth. Build and maintain the schema objects and abstraction layer on top of various, changing enterprise assets.
- Mobile Architect (MBA)**: Build, compile, deploy, and maintain mobile environments and applications. Optimize the user experience when accessing applications via mobile devices. Integrate with preferred VPN, SSO, and EMM protocols.
- Services Architect (SVA)**: Inject, extend, and embed analytics into portals, third-party, mobile, and white-labeled applications. Publish web services and data services for use by Developers in building departmental applications.
- Database Architect (DBA)**: Design and maintain database enterprise assets. Optimize database performance and utilization based on query type, usage patterns, and application design requirements.
- Platform Administrator (PLA)**: Install and configure the Intelligence Architecture on-premises and/or in the cloud. Maintain the security layer, monitor system usage, and optimize architecture in order to reduce errors, maximize uptime, and boost performance.
- System Administrator (SLA)**: Set up, maintain, monitor, and continuously support the infrastructure environment through deployment on AWS, Azure, Windows, or Linux, all while optimizing performance and controlling costs.

Your role: the Platform Administrator

With your extensive background in the analytics platform administration, your CEO has selected you to become MartZon's Platform Administrator. You are responsible for overseeing the planning, creating, publishing, operating, assessing, and optimizing of the enterprise intelligence platform environments.



The enterprise intelligence platform environment hosts the data, application, security, and related services necessary to deploy analytics and mobility applications throughout the enterprise.

As a Platform Administrator, some of your responsibilities include:

- **Project configuration**—Direct the configuration of MicroStrategy environments to ensure that they are tuned and optimized.
- **User provisioning**—Plan the strategy for user provisioning to ensure secure and effective management of users and groups.
- **Analytics security architecting**—Develop guidelines for safeguarding your analytics platform environments.
- **Distribution Services subscription configuration**—Develop guidelines for configuring Distribution Services to enable the scheduling and delivery of reports, dossiers, and documents to subscribers' email accounts, file locations, and printers. As part of the subscriptions, your team will define the subscription recipients, delivery method, and frequency.
- **Cube configuration**—Develop the cube configuration and refresh strategy for easy data maintenance.

- **Cache management**—Create the cache management strategy to ensure fast data access.
- **Project objects migration**—Devise a strategy for synchronizing all analytics platform environments using tools such as Object Manager and Project Merge Wizard.
- **Platform environment monitoring and alerting**—Devise guidelines for monitoring the environment for key parameters and using alerting tools to ensure that the environment is operating optimally.

Your administration team

Your main objective as MartZon's Platform Administrator is to develop standards that define clear duties and best practices for the members of your team who perform platform administration work. Through these standards, you guide administrators in configuring well-tuned and stable analytics environments.

The standards that you develop will ensure that consistent administration practices are implemented by your team to produce consistent results.

Platform Administrator's experience and qualifications

As the Platform Administrator in your organization, you should have the following experience and qualifications:

- Experience with administering MicroStrategy projects from end to end.
- Extensive knowledge of MicroStrategy products, plus experience with project configuration, the security model, object migrations, caching, user provisioning and security filters, Intelligent Cubes management, metadata creation, VLDB settings, use of administrative tools such as Command Manager and Integrity Manager, and performance tuning and platform optimizations.
- Performance tuning and troubleshooting experience.
- Ability to debug and fix technical issues.
- Knowledge of SQL and databases.
- Ability to interact with business users for requirements gathering and support.

In this course, we will revisit the platform administration topics you know, but with a holistic approach of your new Intelligent Enterprise. The course focuses on

the key competencies to help you succeed as the Platform Administrator and support your Intelligent Enterprise in providing a well-tuned and optimized analytics platform environment. You can gain this knowledge through a combination of experience and training, found in the following MicroStrategy classes, culminating in the hands-on certification:

- 2021.311 Administration for Enterprise Analytics
- 2021.312 Administration for MicroStrategy on Cloud
- 2021.314 Working with MicroStrategy on Linux
- 2021.612 Advanced Big Data Administration
- 2021.066 Platform Administrator: Configuration, Managing, and Optimizing
- 2021.036 Platform Administrator (PLA) Certification

Exercise 1.1: Deploying the MicroStrategy platform

MartZon has decided to use the MicroStrategy platform for enterprise analytics and mobility. As the Platform Administrator, you will come up with a list of the deployment steps if installing the platform on premises.

Based on your prior knowledge and experience, list as many steps as you can think of that are involved in deploying the MicroStrategy analytics platform on premises.

Exercise 1.1 Solutions: Deploying the MicroStrategy platform

MartZon has decided to use the MicroStrategy platform for enterprise analytics and mobility. As the Platform Administrator, you will come up with a list of the deployment steps if installing the platform on premises.

Based on your prior knowledge and experience, list as many steps as you can think of that are involved in deploying the MicroStrategy analytics platform on premises.

Some of the steps involved in deploying the MicroStrategy analytics platform are as follows:

- 1** Install and activate the MicroStrategy analytics platform components
- 2** Establish connectivity between various components
- 3** Configure MicroStrategy Intelligence Server, the Web server, Library server, and Developer and set up authentication between these components
- 4** Create and deploy objects, including schema objects (such as attributes and facts), configuration objects (such as users and security roles), and application objects (such as reports and filters)
- 5** Operate and monitor the environment
- 6** Assess and optimize the environment to ensure it is stable and well-tuned

STANDARDIZING THE INTELLIGENT PLATFORM

Communication between those who have stake in the success of the analytics platform is key in an Intelligent Enterprise. Both within the organization as a whole and the Intelligence Center, communication channels should be kept open to ensure that all members are working toward a common goal, mitigating the risk of conflicting priorities.

As the Platform Administrator, you are responsible for coordinating with other Intelligence Center users and creating repeatable standards that steer your team through the requirements gathering process to ensure consistent analytics environment administration at MartZon. These standards will help your team configure, operate, and maintain analytics environments that meet the expectations of business users.

In this chapter, we will review:

- Coordinating with fellow administrators, analysts, application designers, and architects in the Intelligence Center
- Creating a repository to store items such as:
 - The standards documentation that you create
 - The analytics environment artifacts that administrators on your team produce

- Establishing a training schedule to keep your team in tune with standard procedures and upcoming modifications to your analytics environments

Coordinate with other Intelligence Center users

Although each role in the Intelligence Center is responsible for a subset of the analytics solution, you must maintain the communication lines between yourself and other administrators and architects to develop and maintain a well-tuned environment. To do this, plan regular meetings with other Intelligence Center users to communicate proposed changes to the platform and to review and update the standards you have created for your own team.

Specifically, maintain communication with the following roles:

- **Intelligence Director**—Deploys the Intelligence architecture and supervises the Intelligence Center
- **Application Architect**—Creates, shares, and maintains analytics applications
- **Mobile Architect**—Builds, compiles, deploys, and maintains mobile environments and applications
- **Embedded Analytics Architect**—Injects, extends, and embeds analytics into portals, third-party, mobile, and white-labeled applications
- **Database Architect**—Designs and maintains database enterprise assets
- **Analytics Architect**—Designs the enterprise data sets, including data models, dimensional models, project schema, enterprise KPIs, data dictionary, cubes, datamarts, and managed objects.
- **System Administrator**—Sets up, maintains, monitors, and continuously supports the environment infrastructure

Can you think of any platform administration work that will require coordination with each of the above roles?

Example: Communicate object migration schedule

As new reports, attributes, and other objects get created in your organization, your platform administration team will need to migrate those objects across different environments (such as development, test, and production).

When you decide to schedule object migrations, you must share your intentions with other users in the Intelligence Center and discuss the possible impacts. For example, object migrations during certain time frames on a week day may adversely impact your users as the system may need to be taken offline for some time. However, by coordinating with other users, you can schedule the migrations during appropriate time frame (such as off-business hours or on weekends) to avoid such disruption.

Open communication with other users in the Intelligence Center helps you work together to provide a stable and optimized analytics platform for all users when they need it.

What else does your team need to communicate to others in the organization?

Create a documentation repository

In addition to the communication channels between yourself and other users in the Intelligence Center, you must construct a communication channel to share documentation with different teams and project stakeholders. To accomplish this, create a central documentation repository using a documentation management platform like Sharepoint that enables you to maintain documents, control access, and track changes. Providing access to documentation is a vital component of project tracking and continuity. The document repository should contain folders

for individual projects, as well as a dedicated folder for standards and guidelines, as in the following sample:

The screenshot shows a user interface for managing documentation. At the top, there are navigation icons for 'New' (plus sign), 'Upload' (up arrow), 'Sync' (refresh/circular arrow), 'Share' (two people icon), and 'More' (dropdown). Below the header is a search bar with the placeholder 'Find a file' and a magnifying glass icon. A dropdown menu labeled 'All Documents' is open, showing a list of items. The columns are 'Name' and 'Modified'. The items listed are: 'Development project' (modified 35 minutes ago), 'Test project' (modified 35 minutes ago), 'Production project' (modified 35 minutes ago), 'Security requirements' (modified 35 minutes ago), 'Governing settings' (modified 35 minutes ago), and 'Connectivity requirements' (modified 35 minutes ago).

Name	Modified
Development project	35 minutes ago
Test project	35 minutes ago
Production project	35 minutes ago
Security requirements	35 minutes ago
Governing settings	35 minutes ago
Connectivity requirements	35 minutes ago

The documentation repository maintains the following types of documentation.

Intelligent Enterprise standards documentation

As you develop standards and guidelines for the platform administration, share them with your team, as well as other administrators and architects in the Intelligence Center. Distributing your standards ensures that all Intelligence Center teams are properly coordinating efforts, and provides you with an opportunity to verify that your team practices align properly with other teams. The documentation that you create also enables your platform administration team to use consistent practices in installing products, and operating and maintaining platform environments.

For example, you might maintain standards and guidelines in the following sample documents, which can guide administrators in creating artifacts for each environment:

Standards Document	Description
Gathering Requirements	Standards that help administrators navigate the following requirements gathering tasks: <ul style="list-style-type: none">• Interviewing stakeholders• Soliciting requirements for the environment administration, including configuration of security and system settings• Understanding how progress is measured
Examining Data Sources	Guidelines that help administrators perform the following: <ul style="list-style-type: none">• Configuring connectivity• Establishing naming conventions• Configuring governing and other settings
Monitoring and Troubleshooting	A series of processes that can be employed by administrators to perform the following environment monitoring and troubleshooting tasks: <ul style="list-style-type: none">• Understanding analytics platform environment utilization• Investigating dataset execution and publishing• Identifying and troubleshooting connectivity, performance, and other system issues• Determine upgrade impacts
Optimizing Performance	Standards that help administrators complete the following environment performance optimization efforts: <ul style="list-style-type: none">• Determining cache, cubes, and History List management strategies• Optimizing SQL generation• Defining memory, timeout, and other governing settings

Maintain standards documentation in its own folder within your documentation repository such as the one shown in the following image:

MartZon Platform Administration Documentation ➔ Standards

New Upload Sync Share More

All Documents ...

Find a file



✓ Name Modified

Examining Data Sources * ... 11 minutes ago

Gathering Requirements * ... 11 minutes ago

Monitoring and Troubleshooting * ... 11 minutes ago

Optimizing Performance * ... 11 minutes ago

How can you ensure that your standards and guidelines remain relevant as future MicroStrategy updates are released?

Artifacts created for each environment

In addition to the standards documentation you create, the administrators on your team will produce artifacts for each environment in which they are involved. As part of your standards, you will guide administrators in producing artifacts for each environment such as:

- Connectivity requirements
- Authentication requirements
- User and group requirements
- Governing and other configuration settings

Within your documentation repository, create a distinct archive to house artifacts for each environment. For example, you might create distinct repositories for your development, test, and production environments. Sharing access to environment artifacts ensures that your team members and stakeholders in the organization can reference a single source to help them understand the trajectory of each environment, make decisions, and determine whether changes are required.

Establish a training schedule

MicroStrategy environments evolve over time, as will your guidelines for administering environments. The standards that you document must be delivered to your team through regular training sessions and team meetings. You

must also keep all team members abreast of changes made to environment security, connectivity, and so forth.

Exercise 2.1 Discuss the value of communication in the Intelligent Enterprise

Your team recently received a dataset enhancement request from a data analyst in the Finance department who wants to add additional metrics and attributes to an existing Intelligent Cube.

Consider the following questions and write a few answers to each. You will discuss your answers with the class.

- 1 Which parties need to be involved in modifying the Intelligent Cube? For example, is platform administration team responsible for redesigning cubes or is that responsibility of the Analytics Architect team?
- 2 From administration perspective, how would you identify the impacts of updating the cube in question?
- 3 If the Analytics Architect team decides to make the requested updates, in which analytics environment should they make the change? How would you migrate the cube to another environment for testing?
- 4 When would you communicate the changes? Who in the organization would you alert?
- 5 Think of a project that was negatively impacted by a lack of communication between the Intelligence Center teams. Can you think of any project standards that could have been implemented to prevent the problem?

Standardizing requirements gathering

To ensure consistent analytics environment administration at MartZon, you must create standards that guide administrators through the requirements gathering process and help your team configure, operate, and maintain analytics environments that meet the expectations of business users.

For example, effective gathering of department users' application privileges and data access permissions, with appropriate documentation of stakeholders who interact with the analytics systems, is integral to maintaining system access and data security. When such information is properly documented and maintained, system access for new users in the department can be added and users who have left the organization can be removed from the analytics system with confidence. The user security requirements gathering standards that you create eliminate administration guesswork that can lead to problems for business users.

The standards that you create also help administrators produce the Business Requirements and other appropriate documentation for each project.

The following sections discuss the steps you can perform to gather requirements.

Interview experts and stakeholders

To understand business processes and gather requirements for the analytics environments, your team must interview application designers, architects, and other subject matter experts to extract knowledge from the people who are involved with the processes. Gathering requirements enables your team to configure an analytics environment that is fast and stable while also providing the required level of application and data access to business users in the organization.

Your goal is to develop standards and guidelines that help your team reliably conduct interviews and extract relevant information that will help them configure the analytics environment. For example, you might create standards to help platform administrators perform the following requirements gathering activities:

- **Assemble the right people**—Create a list of stakeholders and experts that you want your team to interview for each project. For example, any finance project may require input from the CFO, the accounting team, the sales team, and financial analysts who regularly interact with the MicroStrategy application and the data.
- **Prepare questions in advance**—Create a standard list of questions that you want platform administrators to ask in every interview. These questions should help administrators understand business processes and user goals. For example, a finance project would require questions such as “What data do users in different divisions in the Finance department are authorized to see” (to determine data security measures that would need to be implemented) and “What type of tasks different users perform” (to determine the level of access the users need to different features in the MicroStrategy application).
- **Ask follow-up questions**—Direct administrators to ask clarifying questions to understand how users are using the system, if they are satisfied with the system performance, and so forth. For example, a finance project may require a clarifying question like “In which folders are users allowed to save their reports?” or “Are there any reports, dossiers, or documents that are taking excessively long time to execute?”
- **Shadow data analysts**—In addition to interviewing, steer administrators to watch users to determine how they currently use their analytics platform environment. For example, if the finance department uses an existing platform to process financial disclosure data, administrators should watch the analysts to clarify the process and understand how it can be improved.

- **Anticipate future changes**—Set the expectation that administrators will conduct follow-up interviews after new objects or other changes have been made to their environments. For example, the finance department should be interviewed after a new Intelligent Cube containing the financial data has been migrated to the production environment to determine whether it impacts the response times, and identify opportunities for improvement.

Your platform administration team should implement your interviewing standards to work with applicable business units, understand requirements, and extract relevant information for the analytics environment configuration. This process helps administrators create a MicroStrategy environment that meets the needs of stakeholders. For example, a thorough understanding of the financial disclosure process, its inherent deadlines and dependencies, and user expectations can help administrators create an environment that is fast and stable while ensuring that users are only able to see the data they are authorized to see.

What are some general questions that your team should ask business users during the requirements gathering stage?

Identify the data-related details

As part of your requirements gathering guidelines, direct the platform administration team to leverage business requirements to identify data sources that users use. This helps your team identify the security mechanisms you will need to put in place. In addition, it can also help you in configuring timeout and other system settings.

As the Platform Administrator, you should create standards that guide your team in extracting the desired level of detail from the stakeholders. This information can then be used in configuring the analytics platform environment properly.

For example, your team might speak with the HR department to understand the type of data different users in the division can see. From this conversation, the platform administration team should identify appropriate privileges and permissions that different users in the department need. This information can also help them in creating security filters to ensure that each user is only able to access the data that he is authorized to see.

How can you help your team identify information that is relevant to analytics platform configuration, operation, maintenance, optimization, and troubleshooting?

Establish naming conventions

Naming convention are guidelines that you must establish to direct administrators in configuring environments, users and groups, DSNs, and other objects. These standards must be established in the early phases of analytics environment creation to ensure that all platform administrators on your team create consistent and intuitive names that are logical and predictable to all developers, application designers, architects, analysts, and other stakeholders.

For example, if a naming standard does not exist, the same user may be created with the following names—jdoe, JohnD, john_doe, or johndoe. Without a standard naming convention, it would be difficult to determine if these objects are redundant, or if they are different. This can lead to inconsistent object updates and confusion.

Examine your corporate and business unit guidelines to identify existing naming conventions, and then develop a strategy that can be implemented by administrators to reliably name components that will be created for each analytics platform environment. For example, you might establish the following naming convention of creating users using the `firstname_lastname` format.

As part of your requirements-gathering standards, consider your corporate naming conventions and establish naming conventions for administrators to implement when configuring analytics platform environment components.

PLATFORM ADMINISTRATION PLANNING

Cloud computing adoption continues to increase as organizations look to modernize their IT infrastructure, replacing their on-premises model for workload location. As MartZon transitions to an Intelligent Enterprise, you play a pivotal role in choosing a MicroStrategy cloud deployment option that is best suited for your organization. You collaborate with the Intelligence Director, System Administrator, and other architects in your team to devise an effective deployment strategy. As part of this task, you set the objectives for achieving your organizational goals, reducing capital expenditure, operational cost, and overhead to drive business profitability and efficiency.

In this chapter, we review:

- The planning for the platform architecture, including MicroStrategy analytics platform deployment options
- Evaluating MicroStrategy deployment options best suited for your organization
- Platform architecture best practices

Platform architecture

As the Platform Administrator, you develop guidelines for the deployment, management, and operation of the various components and products that makeup your analytics environment architecture. The types of guidelines you develop depend on the specific MicroStrategy Platform type and deployment strategy you choose to host your BI environment.

MicroStrategy Platform is available in two varieties:

- MicroStrategy Enterprise Platform
- MicroStrategy Cloud Platform

MicroStrategy Enterprise Platform

MicroStrategy Enterprise Platform is a complete version of the MicroStrategy Platform that you download and install in your own on-premises data center, an Enterprise Data Center, or a public cloud infrastructure. It is optimally configured for on-premises deployments. This platform type is:

- Available separately for both Windows and Linux
- Benefits:
 - **Flexibility:** Broadest possible range of architectural flexibility
 - **Freedom:** Freedom to deploy MicroStrategy into existing on-premises environments and leverage prior capital investments
 - **Control:** Locate your data and servers together where industry regulations require specific security and governance
 - **Safekeeping:** Safekeeping of Intellectual property

MicroStrategy Cloud Platform

MicroStrategy Cloud Platform is a complete version of the MicroStrategy Platform that you can provision into your own AWS or Azure regions in minutes. Developed using dozens of AWS or Azure services, the platform is optimally configured for the cloud. This platform type is:

- Available separately for both AWS and Azure
- Licensed either Perpetually or on a Term basis
- Benefits:

- **Optimized pre-configured architecture:** All necessary analytics, software, infrastructure, and hardware components combine to deliver unmatched application performance, scalability, stability, and high availability.
- **Flexible IT assets:** Rapidly spin-up and discard environments for cost-effective prototyping and testing.
 - Automate environment backups to enable rapid cloning and easy recovery.
 - Schedule environment scaling to meet demand and control costs.
- **Isolated environment:** Each MicroStrategy Cloud Environment is a dedicated architecture deployed within a separate Virtual Private Cloud providing maximum security, performance, and flexibility.

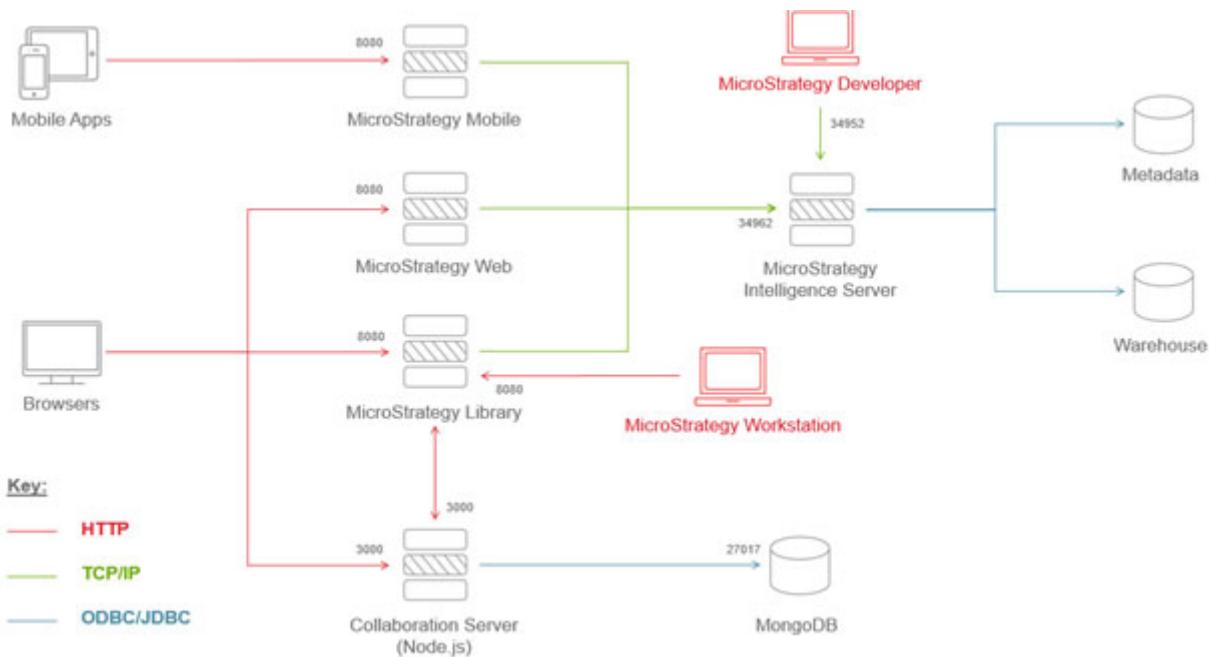
Configuring your environment

If you are deploying MicroStrategy Enterprise Platform, you are responsible for the installation of MicroStrategy Intelligence Server, Web, Mobile, Library, and other components in your environment. For example, in production environments, the Intelligence Server should be installed on a separate, dedicated machine, as Intelligence Server handles a variety of resource-intensive tasks such as caching and cube generation. Similarly, MicroStrategy Web should be installed on a separate, dedicated machine as most of the users in the production environments are web users, submitting requests via their browsers and placing a heavy load on the web server.

The production environments are typically based on a four-tier architecture as analysts and application designers submit requests using a browser or the MicroStrategy Mobile app. The four-tier architecture consists of various MicroStrategy platform components, including Intelligence Server and MicroStrategy Web or MicroStrategy Mobile. The Intelligence Server connects to the metadata and data warehouse using the open database connectivity (ODBC) protocol. Developer connects to Intelligence Server through Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP is a communication protocol used to connect to and communicate with other computers on the Internet or the network.

MicroStrategy Web or MicroStrategy Mobile servers communicate with the metadata and data warehouse through the Intelligence Server using TCP/IP.

The following image shows a four-tier MicroStrategy platform environment:



Exercise 3.1: Review of MicroStrategy analytics architecture

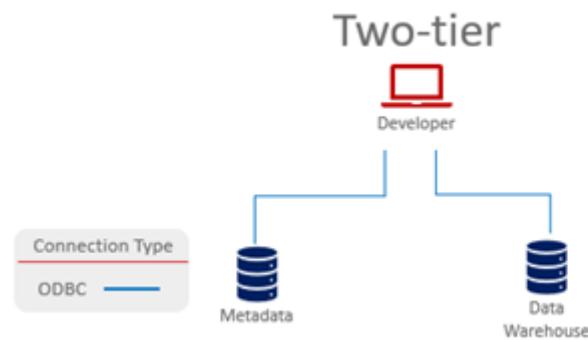
As you learned, a four-tier environment is used in production environments, with the users using web browsers or MicroStrategy Mobile apps to interact with the MicroStrategy analytics platform.

Briefly explain the architectural components and use cases for:

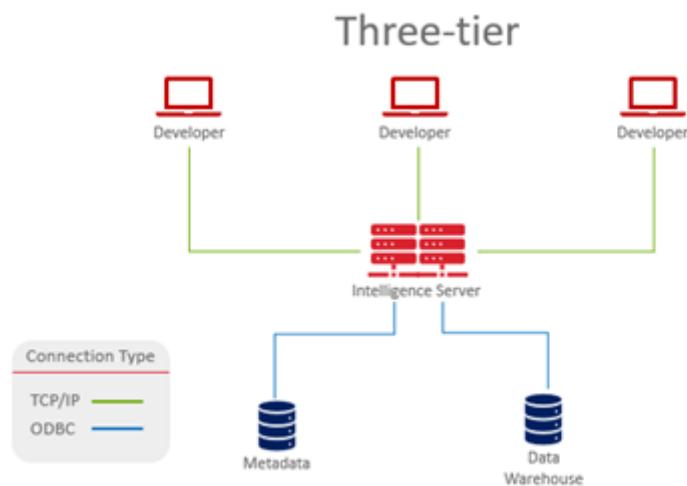
- Two-tier environment
- Three-tier environment

Exercise 3.1 Solution: Review of MicroStrategy analytics architecture

Two-tier environment—The two-tier environment or direct connection setup consists of MicroStrategy Developer connected directly to the metadata and data warehouse. The two-tier environment can only be used in the Windows environment and is primarily used for troubleshooting purposes. For example, if you enable logging in MicroStrategy Diagnostics and Configuration tool for troubleshooting, some of the components require you to restart Intelligence Server. However, by troubleshooting in a two-tier environment where feasible, you do not need to restart Intelligence Server.



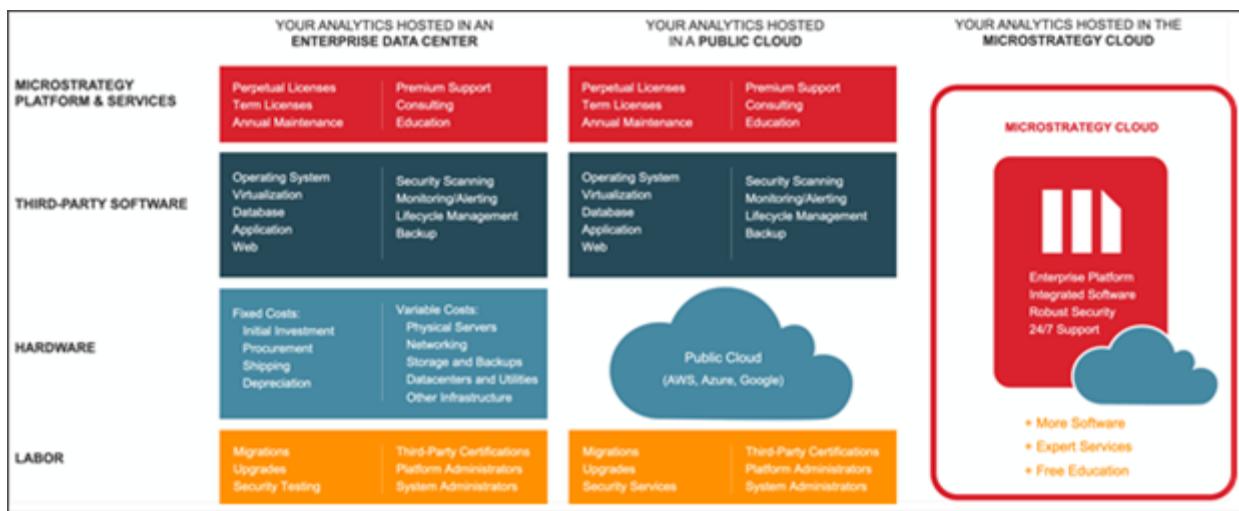
Three-tier environment—In a three-tier analytics environment, Intelligence Server connects to the metadata and data warehouse using the ODBC protocol. Developer connects to Intelligence Server through TCP/IP. The three-tier analytics environment is primarily used by architects and administrators for project architecting (such as creating projects and various schema objects) and Intelligence Server administration, respectively.



MicroStrategy platform deployment options

As MartZon transitions to an Intelligent Enterprise, you lead and advise your team on the deployment, configuration, monitoring, optimization, and upgrading of the MicroStrategy analytics platform.

To assist with this analysis, let's review the deployment options available for MicroStrategy platform:



- **Analytics hosted in an enterprise data center**
 - MicroStrategy Enterprise Platform (MEP)
 - Private infrastructure deployment model, managed by MatZon's internal IT support
- **Analytics hosted in a public cloud**
 - MicroStrategy Cloud Platform (MCP)
 - Public infrastructure deployment model, managed by MatZon's internal IT support
- **Analytics hosted in the MicroStrategy Cloud**
 - MicroStrategy Cloud Environment (MCE)
 - Managed by MicroStrategy Cloud Support

Enterprise data center and public cloud deployments

MicroStrategy Enterprise Platform can be deployed to MartZon's own data center or infrastructure acquired as a service from a public cloud provider. Leveraging a public cloud infrastructure enables MartZon to reduce or eliminate the need to purchase new equipment, manage software, and operate data centers.

In both deployment scenarios, MartZon IT is responsible for managing all aspects of your MicroStrategy environment, including deployment, configuration, performance tuning, troubleshooting issues, and environment monitoring.

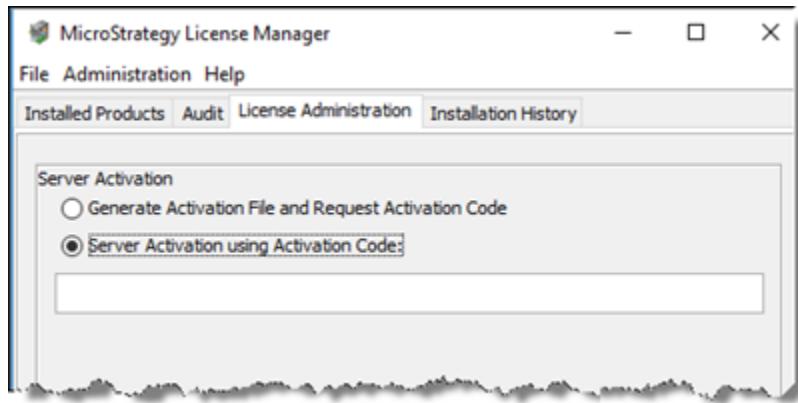
You need to direct your team to install the required platform components and activate the server installation. Your team is responsible for the following:

- **Installing platform components**—You direct the installation of the MicroStrategy software components in all analytics environments using MicroStrategy Installation Wizard or response files to ensure your environments are up and running.

Depending on your licensing agreement, this task may require your team to:

 - Install components such as Intelligence Server, Mobile, and Web to allow users to fetch data from the data sources using different client interfaces, including web browsers and mobile devices
 - Install Collaboration Server to enable users to collaborate with each other (via comments, notifications, and user invites)
 - Install MicroStrategy clients products and tools (such as Integrity Manager and Command Manager) to allow your team to manage the analytics platform environments
- **Server activation and licensing**—Using tools such as the Installation Wizard or License Manager, your team apply the MicroStrategy licensing key and

activation code to ensure that all servers installations are active, and MicroStrategy software installation complies with the license contract.



MicroStrategy server activation is a licensing technology that ensures that installations of MicroStrategy server products are authentic and have been legitimately licensed. Installing, modifying, or upgrading MicroStrategy server installations generates an activation XML file that contains information about the installation. This XML file can be uploaded to MicroStrategy, either automatically in the installation routine, or through License Manager, or manually via a secure web site, <https://licensing.microstrategy.com>.

MicroStrategy then creates a machine-specific activation code which is applied automatically or manually to the server installation to activate the server installation.

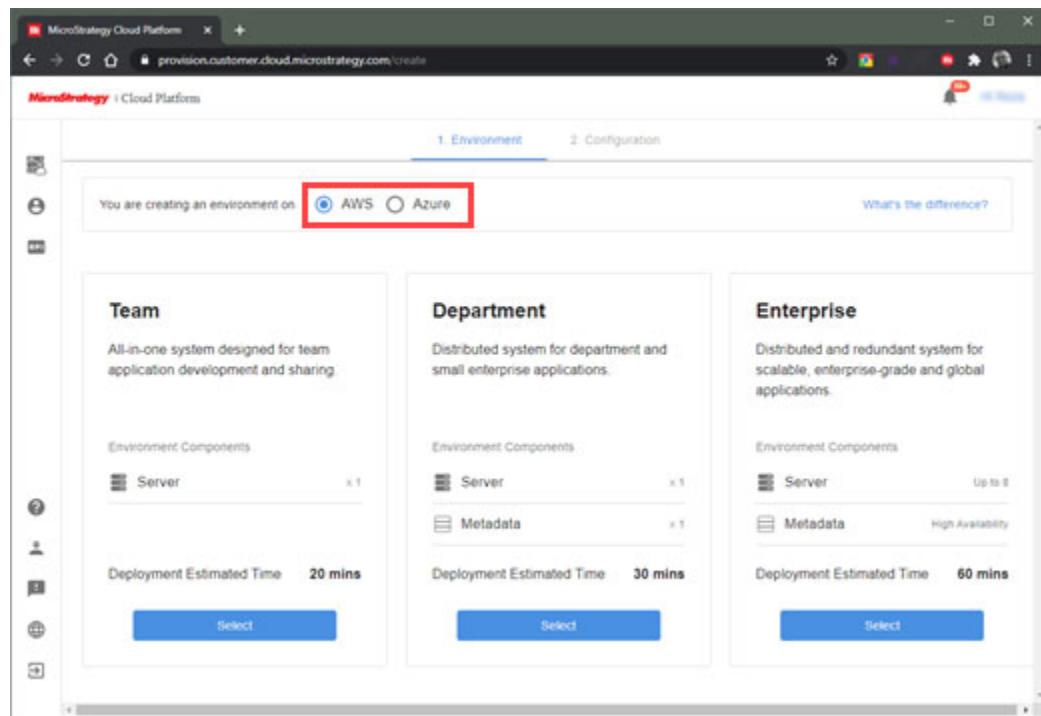
MicroStrategy Cloud Platform deployment

As you evaluate various options and deployment strategies for MartZon, consider the expertise needed for users to create cloud architectures including virtual machines, networks, disk, firewalls, and other infrastructure resources. Another factor to consider is the amount of time and effort required to deploy, configure, and optimize MicroStrategy software on the provisioned resources. Tasking MartZon IT support to manually facilitate deployments for users would be an enormous waste of your organization's time and resources. Cloud computing provides the ability to manage Infrastructure as a Code (IaC), presenting you with the opportunity to automate your infrastructure deployments. That means having the ability to create, schedule, and terminate various environments on-demand, based on user requirements such as location, size, operating system, and platform version.

MicroStrategy Cloud Console is a deployment management solution developed by MicroStrategy to automate repeatable deployment tasks, improving reliability, stability, and speed of your MCP deployments. Authorized MartZon users can

launch and manage enterprise analytics, mobility, and HyperIntelligence projects in environments hosted on Microsoft Azure or Amazon Web Services (AWS).

MartZon users from different departments and business units can launch their own dedicated, fully-configured development, UAT, or production environments operating MicroStrategy Cloud Platform with complete flexibility. Your users can choose the cloud platform they wish to deploy to and maintain full control over the AWS account or Azure subscriptions in which their MCP is hosted.



You have the flexibility of choosing the specific compute infrastructure that meets your needs based on cost or usage requirements. You can manage your MicroStrategy environment (such as Start, Stop, Restart, and Resize), on demand or use the Scheduling services.

To support your corporate security policies, you can deploy, manage, upgrade, and resize your MicroStrategy environment on your own Amazon Virtual Private Cloud (VPC) or Azure Virtual Network (VNet).

MicroStrategy Cloud Environment: Fully managed option

In this fully-managed cloud deployment option, the MicroStrategy Cloud Support team creates and manages your private cloud instance, providing maximum performance, security, and customization. This service includes a complete

environment with features like intrusion detection, firewalls, load balancers, and networking pre-configured for the MicroStrategy environment. The MicroStrategy Cloud Support team also provides upgrades, full-stack maintenance and patching, monitoring, compliance certification management, and 24x7 technical support.

The fully-managed cloud deployment option offers the following advantages:

- Fast startup with zero capital expenditure and minimal financial and operational risk
- The ability to build once and use across many interfaces
- The ability to easily and quickly scale up and scale out

Choosing the right cloud option

With on-premises or enterprise data center (private cloud) deployments, you get complete control and responsibility for the hardware, software, infrastructure, and the analytics platform implementation, operation, infrastructure security, tuning, maintenance, and troubleshooting. The on-premises deployment may also be necessary for regulatory compliance reasons, or if you already have a data center with a trained staff.

With MicroStrategy on cloud options, you get seamless access to the latest platform features, with built-in infrastructure security and compliance. These options enable you to deliver business value quickly as you can start using the analytics platform right away after deployment. You can also adapt quickly to organizational growth and changes as you can start small and then resize the environments as your business grows.

Best
Practice

Platform architecture best practices

Planning for the platform environments architecture involves licensing and installation of various platform components. Your platform administration team should follow the following best practices related to the installation and licensing of the MicroStrategy Enterprise platform.

Best practices for installing MicroStrategy Enterprise Platform

- 1 **Install package download**—When downloading the install package, it is recommended to place and unzip the installation package locally to the machine where you plan to install MicroStrategy instead of running the installation from a network or shared drive.

- 2 License key**—Before you begin the installation, obtain the license key for the MicroStrategy analytics platform.
- 3 Server type**—As the Intelligence Server performs a wide variety of memory-intensive processes, including caching, scheduling, and cube generation, you should always install the Intelligence Server on its own dedicated server in production environments

Similarly, as most users access the MicroStrategy platform through a browser, you should install MicroStrategy Web on its own dedicated web server.

Other components of the platform can be installed together in varying combinations, depending on the available machine resources. Similarly for a development or a test environment, all the components can be installed in a single server, as the stress level on this environment will be limited to a single user or a small team.

- 4 Automatic restart**—To speed up the installation process, when you use the Installation Wizard to install the MicroStrategy products, you can select to automatically restart the machine after the installation is complete. This allows you to complete other tasks while the installation proceeds. After the machine reboots, you can log in to complete the configuration.
- 5 Services and ports**—Depending on your selections at the start of the installation, the following services listed in the table may be installed and configured.

MicroStrategy service/process	Default port
Apache Kafka	9092
Apache Tomcat (Dossier Web)	8080
Apache Zookeeper	2181
PostgreSQL	5432
Collaboration Server	3000
Export Engine Micro-Service	20100
MicroStrategy REST Server	34962
MicroStrategy Intelligence Server	34952

Where feasible, you should use the default ports for MicroStrategy services and processes to avoid any issues.

Additionally, the user who performed the installation or a user who has permissions to access all the install files should run the Intelligence Server process.

6 Hardware and software requirements—Each MicroStrategy product and tool should be installed on a certified configuration, taking into consideration parameters such as the following:

- Operating system (type and version)
- CPU architecture
- Minimum hardware requirement (RAM, disk space)
- Recommended filesets

You should refer to the Readme for the version you are installing and ensure that your environment meets the specified hardware and software requirements.

The information in the Readme should be considered as general guidance on hardware requirements to support the MicroStrategy product suite. Your actual hardware requirements may vary depending on the complexity of your MicroStrategy environment, the deployment strategy of MicroStrategy features, user requirements, expected peak usage requirements, and response time expectations.

7 Install process termination—Unless absolutely necessary, you should not terminate an installation setup midway. Instead go through the entire setup process to complete the installation.

8 Reboot during install—If the installation setup requests that the system be rebooted, you should reboot immediately and restart the setup after the reboot.

9 Install cancellation—After starting the installation, if you cancel prior to completion, be aware that certain components may not be removed automatically, and stay installed on your machine.

10 Install files deletion—Do not delete the install files manually. You should always use your operating system's add/remove program options to uninstall applications from your machine.

11 Silent install—If performing silent Installation, be aware that the parameters specified in the response.ini files can change across versions. While an effort is typically made to ensure backwards compatibility of the MicroStrategy platform components, occasionally new parameters are required. You should

always refer to the provided sample response.ini files included with the installation for each release.



Response.ini is a text file that is used to automate the installation of MicroStrategy platform. It allows you to predefine all the selections you want to make during the installation in the Installation Wizard.

Best Practice

Best practices for MicroStrategy Cloud Platform deployment

- 1 **Configuration option**—MicroStrategy offers different pre-configured deployment options (Team, Department, or Enterprise) for both Azure and AWS cloud infrastructures. You should choose the deployment option based on your cost or usage requirements, and customize it based on various factors such as MicroStrategy platform version, operating system, server instance size, and geographic region.
- 2 **Instance type**—MicroStrategy on AWS environments have both the R3 and the R4 instance types; you should choose the appropriate instance type based on your needs.

R3 instances are geared towards running on memory-intensive applications and offer good input/output (I/O) performance, constant memory bandwidth, support for reduced latency, and maximum packet per second performance.

R4 is the newer generation of Amazon EC2 memory-optimized instances. R4 instances are well-suited for memory-intensive, latency-sensitive workloads required for Business Intelligence (BI), data mining and analysis, in-memory databases, distributed web scale in-memory caching, and applications performing real-time processing of unstructured big data. Due to the use of Enhanced Networking on R4 instances, you also benefit from significantly improved network performance. All the R4 instance sizes deliver high packet per second performance with consistently low latencies.

Exercise 3.2: Explore the cloud environment

Your platform administration team has provisioned a MicroStrategy on Cloud environment. This environment consists of a Windows machine and a Linux machine.

The Windows machine contains various client tools and applications such as MicroStrategy Developer, Putty, and so forth.

Intelligence Server, MicroStrategy Web, MicroStrategy Mobile, metadata, and the data warehouse are installed on the Linux machine.

 As we are in a training environment, there are multiple platform components installed on the same machine. In production environments, you should not install the MicroStrategy applications or the databases on the same machine. For example, you should install Intelligence Server and MicroStrategy Web each on a dedicated machine.

In this exercise, you access the Windows machine in your environment. The login credentials and other information you need for accessing your environment are included in the email titled *Welcome to MicroStrategy on Cloud*.

Next, you learn how to access a three-tier MicroStrategy environment. In a three-tier environment, a project source is used to establish connectivity between the Developer and the Intelligence Server which in turn is connected to the metadata and the warehouse. The project source, MicroStrategy on AWS I-Server, is already pre-configured in your environment. You log in to this project source and access the MicroStrategy Tutorial project which you will be using for completing the subsequent exercises in this course.

Lastly, as most MicroStrategy users typically work in a four-tier environment, you learn how to access MicroStrategy Web in your cloud environment.

Access your cloud environment

- 1 On your local machine, in the MicroStrategy Cloud email, click **Access MicroStrategy Platform**.
- 2 In the **User name** and **Password** boxes, enter the login credentials provided in the MicroStrategy Cloud email. Click **Login**.

The MicroStrategy Cloud landing page is displayed.

- 3 On the landing page, hover over **Remote Desktop Gateway** and click **Launch**.
- 4 In the Remote Desktop Connection window, in the **User name** and **Password** boxes, enter the user name and password listed in the MicroStrategy Cloud email.
- 5 Under All Connections, click **Developer Instance RDP**.

You are connected to the Windows machine in your cloud environment.

Access Developer in a three-tier environment

- 1 On the Windows desktop, double-click the **Developer** shortcut.
- 2 In Developer, double-click the **MicroStrategy on AWS I-Server** project source name.
- 3 In the Login window, in the Login id and Password boxes, type the **User Name** and the **Password** provided in the MicroStrategy Cloud email and click **OK**.

You can now access the MicroStrategy Tutorial (and other available projects) in project source.

Access MicroStrategy Web a four-tier environment

- 1 On your local machine, in the MicroStrategy Cloud email, click **Access MicroStrategy Platform**.
- 2 In the **User Name** and **Password** boxes, type (or copy and paste) the login credentials provided in the MicroStrategy Cloud email.
- 3 Click **Login**. The MicroStrategy Cloud landing page is displayed
- 4 On the landing page, hover over **MicroStrategy Web** and click **Launch** to open the MicroStrategy Tutorial project in MicroStrategy Web.
- 5 Minimize the MicroStrategy Web window.

Exercise 3.3: Access applications on the Linux machine

In your cloud environment, Intelligence Server Universal, MicroStrategy Web Universal, MicroStrategy Mobile server, and the metadata and the data warehouse databases reside on the Linux machine. To access the Linux machine from the Windows machine, you can use various tools such as Putty and VNC.

Putty is an SSH client that enables you to connect to the remote Linux machine using a Command Line Interface (CLI).

VNC enables you to connect to the remote Linux machine using a GUI. It makes use of an X VNC-server running on the host (Linux) machine, while the user connects using the VNC client on the guest machine. Compared to Putty, VNC is relatively resource heavy and uses a lot of bandwidth as it constantly sends screen displays to the client machine.

This exercise shows you how to use VNC for accessing the Linux machine from the Windows machine in your cloud environment. From the Linux machine, you can launch various MicroStrategy applications and tools. In this exercise, you will launch Command Manager.

Next, you will again launch Command Manager but instead of using VNC, you will use Putty with Xming for accessing the Linux machine.

After completing this exercise, you will be familiar with the use of console-based Putty and GUI-based VNC. Depending on your preference, you can use either Putty or VNC for accessing the Linux machine, where needed, for completing the subsequent exercises in this course.

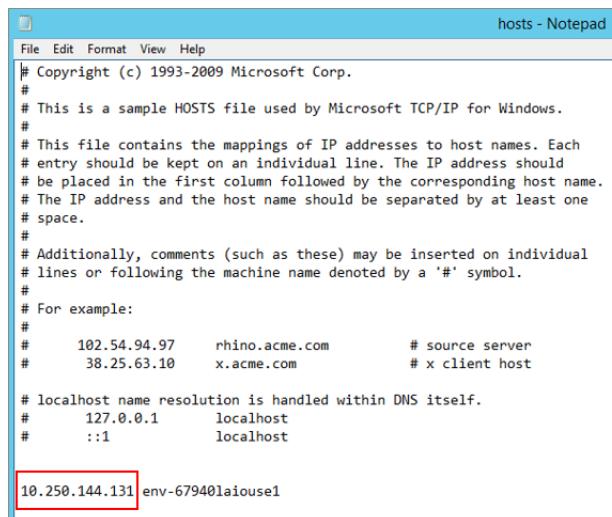
Launch VNC

VNC is an open source, remote desktop application for accessing the Linux environment. To use VNC, you install a server portion of the application on the Linux machine. You can then access the Linux environment remotely using the client part (VNC Viewer) of the application.

- 1 On your Windows desktop, double-click **VNC-Viewer**.
- 2 In the VNC Viewer window, on the **File** menu, click **New connection**.
- 3 In the Properties window:
 - In the VNC Server box, type the IP address of the Linux server:5901. For example, if the IP address is 10.250.144.131, type:

10.250.144.131:5901

The IP address of the Linux server is same as for the Intelligence Server. To find the IP address, on the Windows machine of your cloud environment, double-click the shortcut for the **hosts** file that is available on your desktop and open it in Notepad. You will find the IP address listed at the bottom of the hosts file.



A screenshot of a Microsoft Notepad window titled "hosts - Notepad". The window displays the contents of a hosts file. At the bottom of the file, there is a new entry: "10.250.144.131 env-67940laiousel". This line is highlighted with a red rectangular box.

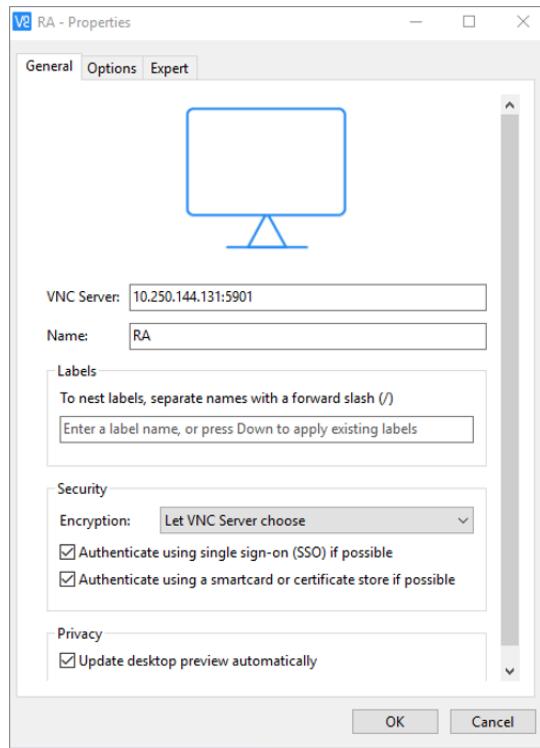
```
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host name.  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#      102.54.94.97    rhino.acme.com        # source server  
#      38.25.63.10     x.acme.com            # x client host  
  
# localhost name resolution is handled within DNS itself.  
#      127.0.0.1        localhost  
#      ::1              localhost  
  
10.250.144.131 env-67940laiousel
```



You can also find the IP address of the Intelligence Server on the MicroStrategy Cloud landing page.

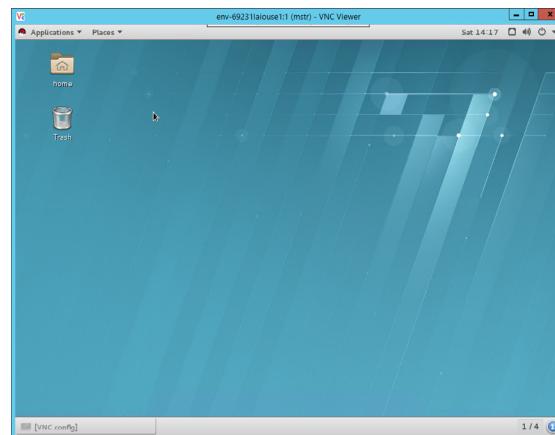
- In the Name box, type your initials as the name of your VNC connection.

Accept all other defaults and click **OK**.



- 4 In the VNC Viewer window, double-click your connection name. In the Encryption window, click **Continue**.
- 5 In the Authentication window, type the password for the **mstr** user provided in the MicroStrategy Cloud email. Select **Remember password** and then click **OK**.

Your remote session opens as illustrated below:



Access the bin directory on your Linux machine

In your cloud environment, Command Manager and other MicroStrategy tools and applications can accessed from the /opt/mstr/MicroStrategy/bin subdirectory.

- 6 Double-click the **Home** icon.
- 7 In the left pane, click **Other Locations**. In the right pane, click **Computer**.
- 8 In the right-pane, double-click **opt**. Under opt, navigate to the **/mstr/MicroStrategy/bin** subdirectory. Locate and double-click **mstrcmdmgrw**.
Command Manager Login window displays. As you do not need to use Command Manager at this time, click **Cancel** and then exit Command Manager.
- 9 Leave your VNC session open as you will use it later for another exercise.

Access Linux machine using Putty

You will now use Xming and Putty for accessing Command Manager on the Linux machine.

Xming is an open source X-Windows terminal emulator that runs natively on Microsoft Windows operating system. It enables you to use client applications like Putty to connect to remote computers and securely forward X11 sessions from Windows computers.

The credentials for logging in to Putty are provided in your MicroStrategy Cloud email. The host name (IP address or the machine name) can be obtained by double-clicking the shortcut for the hosts file located on the Windows desktop of your cloud environment, and opening the hosts file using Notepad.

Launch Xming

- 1 On your Windows desktop, double-click **Xming**.
- 2 Minimize the Xming window.

Launch Putty

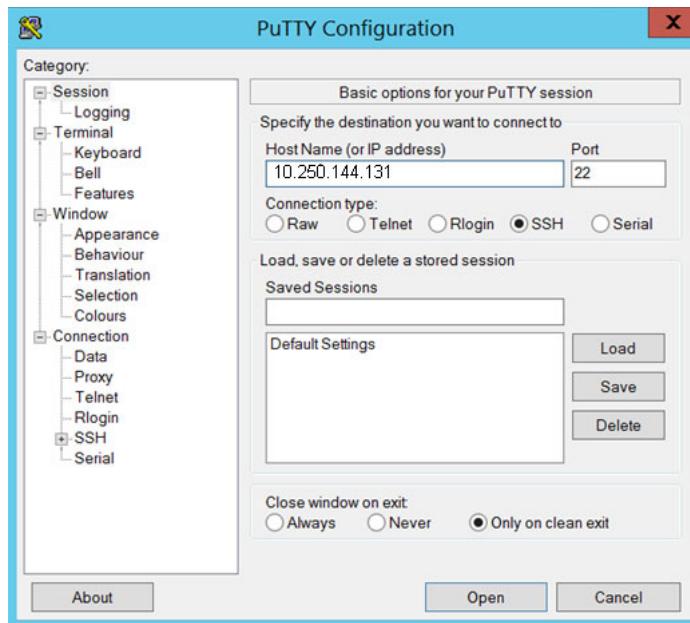
Putty is an open source software that is used as a Secure SHell (SSH) and telnet client for the Windows platform to connect to a remote Linux machine.

- 3 On your Windows desktop of your cloud environment, double-click **Putty**.



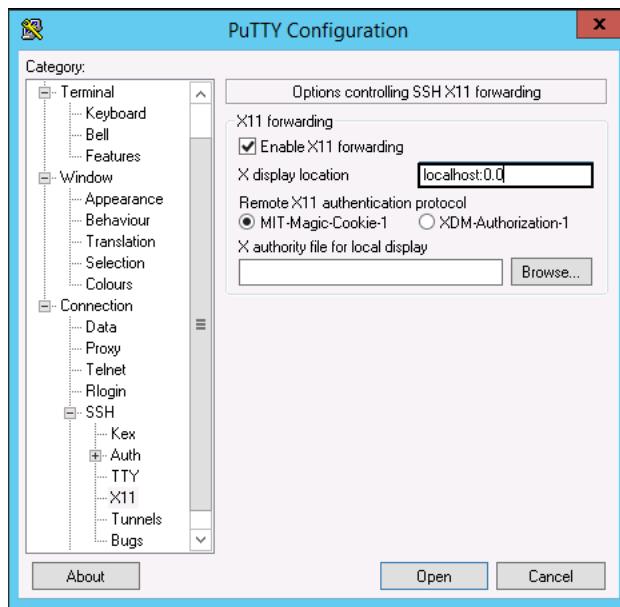
- 4 In the Putty Configuration window, select the **Session** category, and in the **Host Name (or IP address)** box, type the IP address of the Intelligence Server that is listed in the hosts file.

Leave the Port value as **22**.



- 5 In the Putty Configuration window, under Connection, expand **SSH**, and select **X11**.

- 6 Select the **Enable X11 forwarding** check box, and in the **X display location** type **localhost:0.0**.



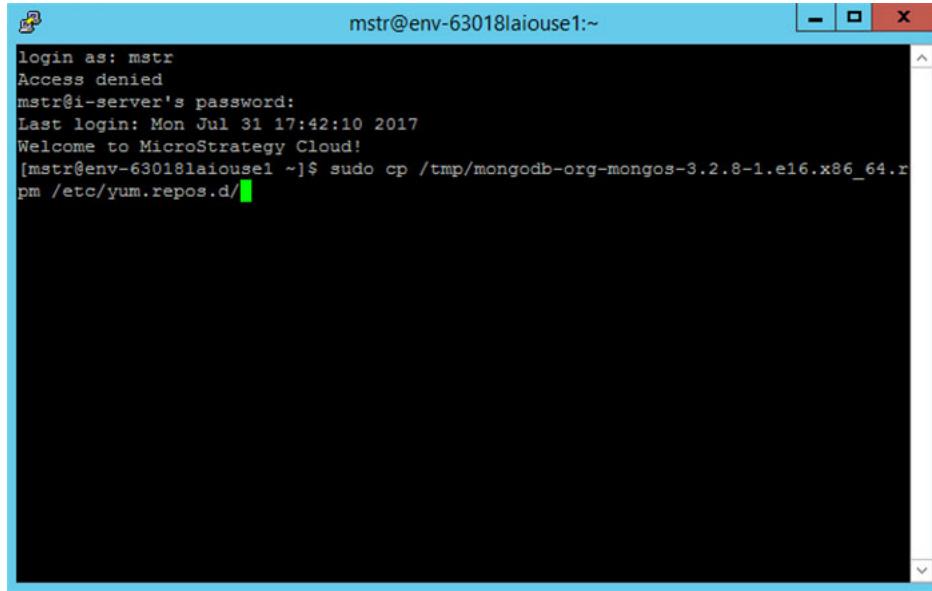
- 7 Under Session, in the **Saved Sessions** box, type **MySession**, and click **Save**.



Saving the session information enables you to access it again later without having to re-enter information in the Putty Configuration window.

- 8 Double-click **MySession** to access your Linux machine.
- 9 Click **Yes**, in the PuTTY Security Alert window.
- 10 When prompted for a login, type the user name (**mstr**) for your SSH session and press **Enter**. You can find the user name (and password) information in the MicroStrategy Cloud email.

11 When prompted, type the password for your SSH session.

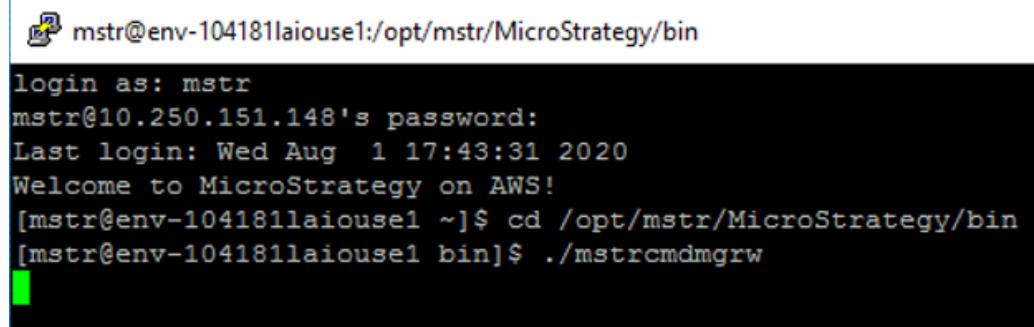


```
mstr@env-63018laiouse1:~  
login as: mstr  
Access denied  
mstr@i-server's password:  
Last login: Mon Jul 31 17:42:10 2017  
Welcome to MicroStrategy Cloud!  
[mstr@env-63018laiouse1 ~]$ sudo cp /tmp/mongodb-org-mongos-3.2.8-1.e16.x86_64.r  
pm /etc/yum.repos.d/
```

Launch Command Manager on your Linux machine

You will now navigate to the bin subdirectory under /opt/mstr/MicroStrategy. You can launch Command Manager and other MicroStrategy applications and tools from the bin subdirectory in your cloud environment.

- 1 On the console, type **cd /opt/mstr/MicroStrategy/bin** and press **Enter**.
- 2 To launch Command Manager, type **./mstrcmdmgrw** and press **Enter**.



```
mstr@env-104181laiouse1:/opt/mstr/MicroStrategy/bin  
login as: mstr  
mstr@10.250.151.148's password:  
Last login: Wed Aug 1 17:43:31 2020  
Welcome to MicroStrategy on AWS!  
[mstr@env-104181laiouse1 ~]$ cd /opt/mstr/MicroStrategy/bin  
[mstr@env-104181laiouse1 bin]$ ./mstrcmdmgrw
```

- 3 Maximize the Xming window.
- 4 Leave your Putty window open as you will use it later for another exercise.

Now that you are familiar with both Putty and VNC, you can use either of these tools for accessing the Linux machine, where needed, for completing the subsequent exercises in this course.

Exercise 3.4: Deploy MicroStrategy Web Universal

An instance of MicroStrategy Web is already installed in your MicroStrategy Cloud environment. You can access it either through the link for MicroStrategy Web on the landing page, or by pasting the following URL in a web browser:

<https://env-xxxxxx.customer.cloud.microstrategy.com/MicroStrategy/servlet/mstrWeb>

where xxxxxx is your environment number. The environment number can be found in the URL of the landing page.

In this exercise, you will guide your team in deploying an instance of MicroStrategy Web using the MicroStrategy.war available on your Linux machine. As the existing instance of MicroStrategy Web already uses MicroStrategy as the context root, you will rename the MicroStrategy.war as Martzon.war and then deploy it on the Linux machine using WinSCP.

Access WinSCP and download MicroStrategy.war

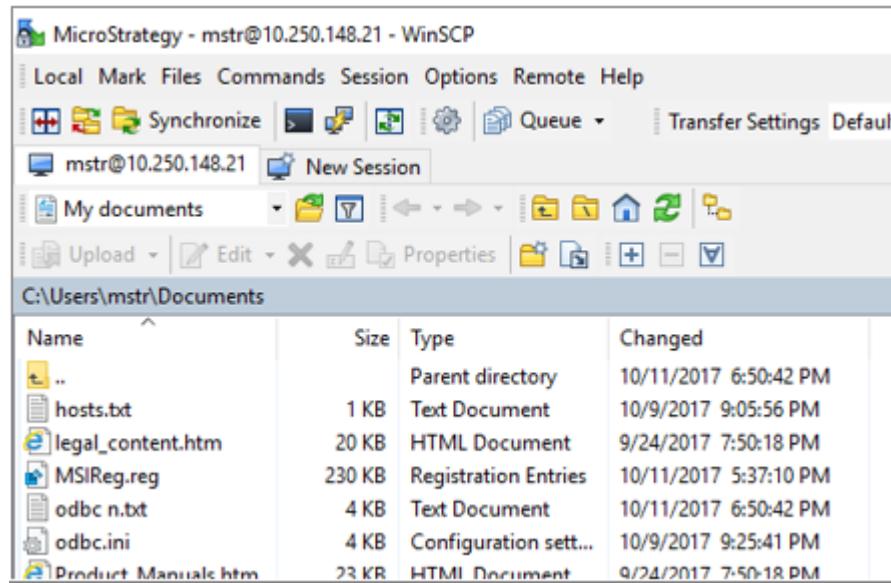
Access WinSCP

- 1 On your Windows desktop, access **WinSCP**.

Download MicroStrategy.war

You will now download MicroStrategy.war which is installed during the MicroStrategy platform installation in the /opt/mstr/MicroStrategy/install/WebUniversal directory.

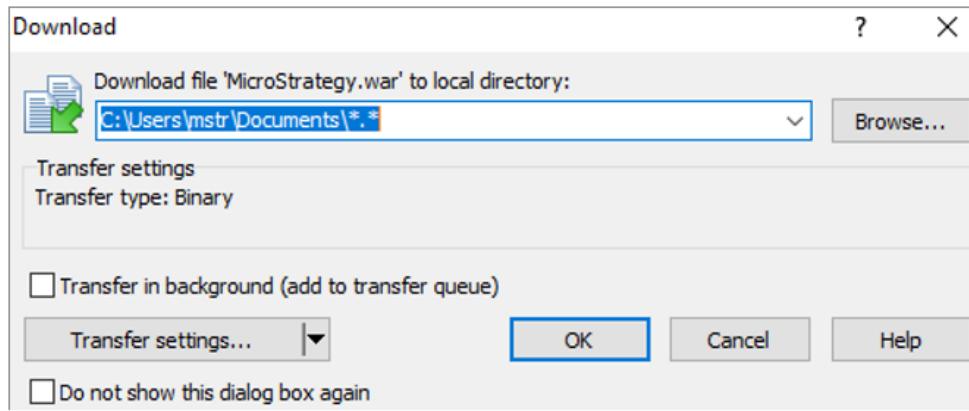
- 2 In the WinSCP window, in the drop down list on top of the right pane, from the root directory, browse to the **/opt/mstr/MicroStrategy/install/WebUniversal** directory.



- 3 Right-click **MicroStrategy.war**, point to **Download**, and select **Download**.



-
- 4 Accept the default options in the Download window and click **OK**.

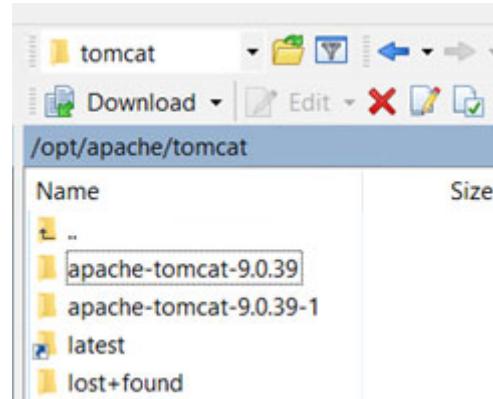


The MicroStrategy.war file will be downloaded to the **C:\Users\mstr\Documents** directory.

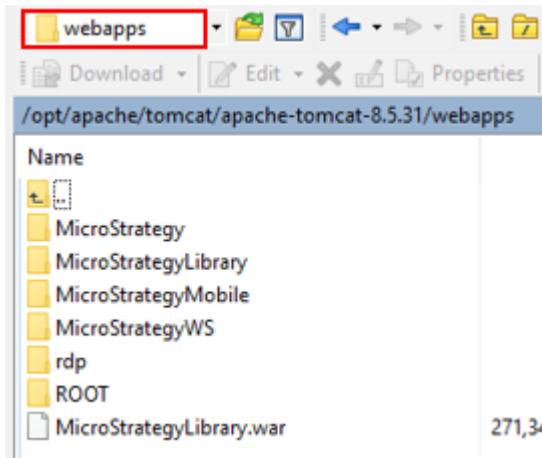
Rename the war file

As the existing instance of MicroStrategy Web already uses MicroStrategy as the context root, you will rename the MicroStrategy.war as Martzon.war and then deploy it to the webapps subdirectory.

- 5 In WinSCP, in the left pane, ensure that you are in the **C:\Users\Documents** directory and then right-click **MicroStrategy.war** and select **Rename**.
- 6 Rename the file to **MartZan.war**.
- 7 In the WinSCP window, in the drop down list on top of the right pane, from the root directory, browse to **/opt/apache/tomcat/latest**.



- 8 Double-click **webapps** to navigate to that subdirectory.



- 9 Drag **MartZon.war** from the left pane of WinSCP to the right pane under the **/opt/apache/tomcat/latest/webapps** directory.

If displayed, in the Upload window, click **OK**.

You have now deployed a new instance of MicroStrategy Web called PlatformAdmin.

- 10 On the toolbar on the right pane of WinSCP, click the **Refresh** icon. You should see a directory called **MartZon**.



You may have to click **Refresh** a couple of times to display the **MartZon** directory. If for any reason, the directory does not display in your WinSCP window even after refreshing a couple of times, access Putty and type **service mstr tomcatrestart** to restart Tomcat. After Tomcat has restarted, you should see the **MartZon** directory.

Enterprise platform management

As a Platform Administrator, you need to address the full spectrum of enterprise analytics needs. These range from self-service data discovery for the autonomous business user, to advanced and predictive analytics for the data scientist, to the automated distribution of personalized reports and dossiers to thousands of users across your organization.

MicroStrategy Workstation simplifies administration of enterprise analytics environments by bringing all key workflows into a single unified tool for content deployment, task automation, architecture management, and system monitoring. This allows MartZon enterprise BI developers and administrators to streamline

common tasks associated with creating and deploying sophisticated enterprise and mobile analytics applications.

Exercise 3.5: Create a self service application

Senior management in your IT department want to provide your business users with access to data discovery functionality and self-service analytics to meet their demands for agility. Users need to gather, analyze, and visualize their data at their own pace and convince. As part of this effort, you need to allow users to create, manage, and share datasets, dimensions, and KPIs to enhance their productivity.

In this exercise, using MicroStrategy Workstation, you connect to your environment and create a new application for your self-service BI users. Through Workstation, you can connect to and manage multiple environments, and access different projects and datasets in one interface.

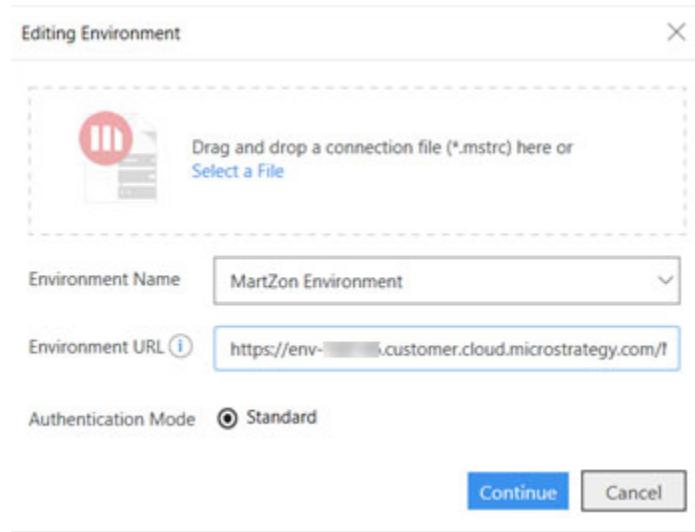
Connect Workstation to your Intelligence Server

- 1 On your Windows machine, double-click the **MicroStrategy Workstation**  shortcut to launch Workstation.
- 2 From the navigation pane, select **Environments**.
- 3 Click **Add New Environment Connection**.
- 4 Type **MartZon Environment** for the **Environment Name**.
- 5 For the **Environment URL**, copy and paste the following URL:

**[https://env-xxxxxx.customer.cloud.microstrategy.com/
MicroStrategyLibrary](https://env-xxxxxx.customer.cloud.microstrategy.com/MicroStrategyLibrary)**

Replace xxxxxx with your environment number.

6 Click **Continue**.



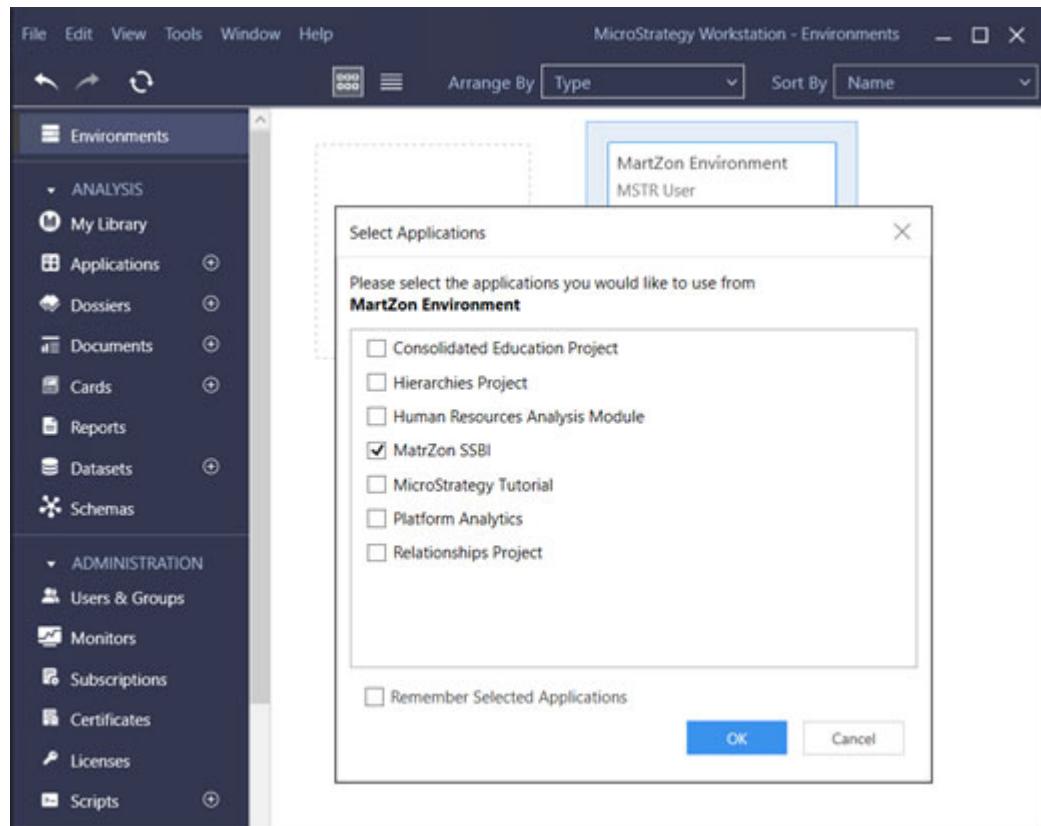
- 7 Enter the **Username** and **Password** from your MicroStrategy Cloud email, then click **Connect**.
- 8 Select **MicroStrategy Tutorial** to include this application in your environment and click **OK**.
- 9 From the navigation pane, select **Applications** to view the applications selected in your environment.
- 10 From the **File** menu, select **New Application** to create a new application in your environment.
- 11 For **Application Name**, enter **MatrZon SSBI**.
- 12 For **Description**, enter **MartZon self-service BI Application**, then click **Create**.
- 13 After your application has been created, click **OK** to continue.

Remove the Tutorial Project from available applications

Since you no longer need to manage any objects in the MicroStrategy Tutorial application, remove it from the list of applications available in your

environment. You can always add or remove applications from your environment.

- 1 From the navigation pane, select **Environments**, then right-click **MartZon Environment** and select **Update Application List**.
- 2 Clear the check box next to **MicroStrategy Tutorial**, then select **OK**.



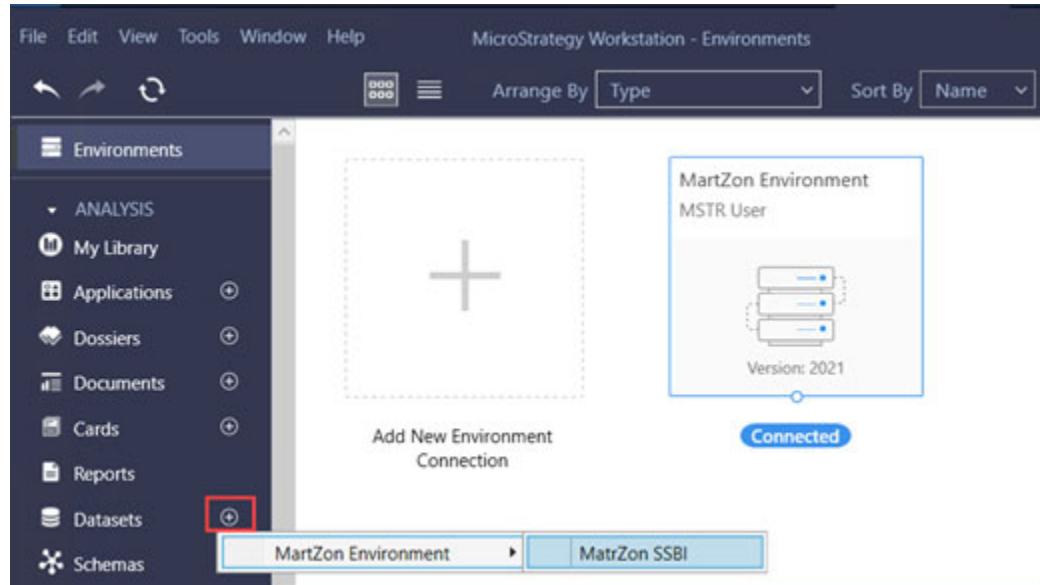
Exercise 3.6: Connect and import data from an external data warehouse for your project users

As an administrator, you can improve the effectiveness of your BI environment by importing, cleaning, and curating data that informs your users. In this exercise, you connect to an external data warehouse hosted on PostgreSQL and create a data import cube to make it easier for your users to discover and analyses sales data.

When you import data, each column in a table becomes a separate attribute. A common issue when importing data is when attribute forms are imported as if they are separate attributes. For example, your company's products are represented in your data by a Category attribute, but this attribute is imported as an attribute called Category ID and another, separate attribute called Category DESC. As part of this exercise, you create multiform attributes to combine the two forms into a single attribute.

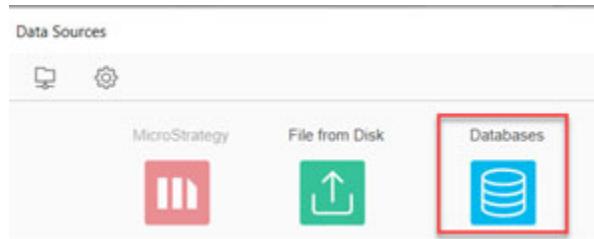
Create a connection to PostgreSQL

- 1 From the Navigation pane, click the **Create a New Data** icon next to Datasets.

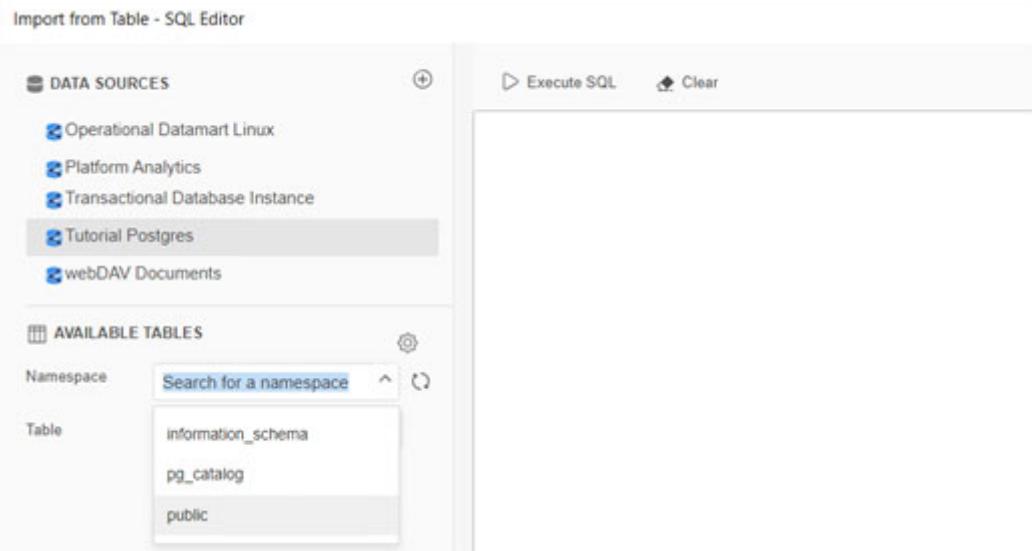


- 2 Select **MartZon environment**, then **MartZon SSBI** application.

3 In the Data Sources window, select **Databases**.



- 4 Select **Type a Query** and select **Next**.
- 5 Maximize the import from table - SQL Editor window.
- 6 From the left pane, under data sources, select **Tutorial Postgres**, then from the **Namespace** drop-down list, select **public**.



- 7 Navigate to your student exercise folder, and open the **code_snippet.txt** file.

8 Copy and paste the SQL for this exercise into the editor pane, then click Execute SQL.

Import from Table - SQL Editor

```

select a13.CATEGORY_ID AS CATEGORY_ID,
       max(a17.CATEGORY_DESC) AS CATEGORY_DESC,
       a12.SUBCAT_ID AS SUBCAT_ID,
       max(a13.SUBCAT_DESC) AS SUBCAT_DESC,
       all.ITEM_ID AS item_id,
       max(a12.ITEM_NAME) AS ITEM_NAME,
       a12.SUPPLIER_ID AS SUPPLIER_ID,
       max(a16.SUPPLIER_NAME) AS SUPPLIER_NAME,
       all.ORDER_DATE AS DAY_DATE,
       sum((all.QTY_SOLD * all.UNIT_COST)) AS Cost,
       sum((all.QTY_SOLD * ((all.UNIT_PRICE - all.DISCOUNT) - all.UNIT_COST))) AS Profit,
       sum((all.QTY_SOLD * (all.UNIT_PRICE - all.DISCOUNT))) AS Revenue
  from public.order_detail all
  join public.lu_item a12
    on (all.item_id = a12.item_id)
  join public.lu_subcateg a13
    on (a12.SUBCAT_ID = a13.SUBCAT_ID)
  join public.lu_day a14
    on (all.ORDER_DATE = a14.DAY_DATE)

```

The screenshot shows the MSTRSQL SQL Editor interface. On the left, there's a sidebar titled "DATA SOURCES" listing various database connections like "Operational Datamart Linux", "Platform Analytics", etc. Below it is the "AVAILABLE TABLES" section, which lists tables under the "public" namespace such as "lu_promo_type_tch", "lu_custstatus_tch", "pmt_inventory", "lu_store", "lu_catalog", "lu_region", and "inventory_q1_2016". The main pane contains the SQL query provided above, with the "Execute SQL" button visible at the top.

Wrangle the data and create multi-form attributes

1 Select **Prepare Data** to wrangle your data before importing.

The screenshot shows the MSTRSQL Preview window. On the left, there's a "Preview" section with a placeholder message "Add a new table". To its right is a "Custom Query" panel containing a list of attributes and metrics:

- Attributes:** Category Desc, Category Id, Day Date, Item Id, Item Name, Subcat Desc, Subcat Id, Supplier Id.
- Metrics:** Cost, Profit, Revenue.

Below this is a data grid with four columns: "Category Desc", "Category Id", "Day Date", and "Item Id". The data in the grid is as follows:

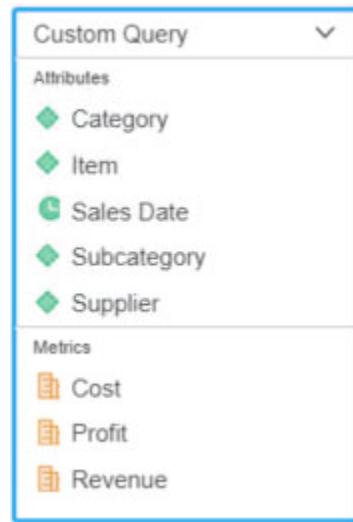
Category Desc	Category Id	Day Date	Item Id
Books	1	2014-12-02 00:00:00.000	1
Books	1	2014-12-03 00:00:00.000	1
Books	1	2014-12-04 00:00:00.000	1

- 2 To create the Category attribute containing both the ID and description, click **Category Id**, then press **CTRL+ Category Desc**.
- 3 With both attributes selected, right-click and select **Create Multi-form Attribute**.
- 4 Change the attribute name to **Category** and clear the **ID Display Form** to show the descriptions only when placed on a grid or visualization.
- 5 Click **Submit**.

Form Category	Display Form
Category Id	<input type="text" value="ID"/> <input checked="" type="checkbox"/> Category Id <input type="button" value="X"/>
Category Desc	<input type="text" value="DESC"/> <input checked="" type="checkbox"/> Category Desc <input type="button" value="X"/>

- 6 Repeat the same steps to create multi-form attributes for:
 - **Item**, using **Item Id** and **Item Name**
 - **Subcategory**, using **Subcategory Id** and **Subcategory Desc**
 - **Supplier**, using **Supplier Id** and **Supplier Name**

Your dataset should resemble the following image:



Save your dataset

- 1 Click **Finish** to save your dataset.

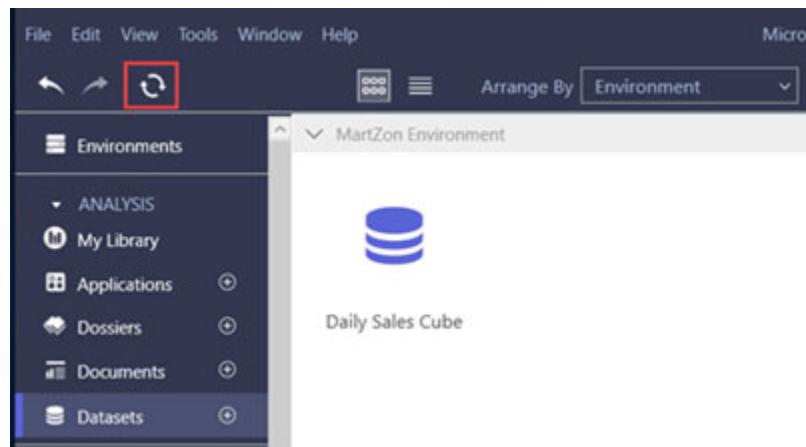
You are presented with two options that determine how your dataset is saved:

Connect Live: Retrieves dataset results directly from a data source. Dragging dataset objects onto visualizations using Connect Live mode can be slower than when using In-Memory mode, since the system retrieves dataset results before they appear. However, the performance of a large dataset can improve by retrieving data incrementally.

In-Memory: Retrieves dataset results from a data source and stores them in memory. Visualizations in the dossier that contain data from the dataset display subsets of these stored results, which can increase the speed in which data appears. You want to save your data as an in-memory dataset.

- 2 Select **Import as in-memory Dataset**.
- 3 Double-click the **MartZon SSBI** folder, then double-click the **Public Objects** folder.
- 4 Double-click the **Reports** folder, name your dataset **Daily Sales Cube** and click **Save**.

- 5 From the Navigation pane, select **Datasets**, then click the refresh icon to see your dataset.



PLATFORM CONFIGURATION

Before you can make the MicroStrategy analytics platform available for use at MartZon, the analytics environments must be configured to allow communication between different components securely. As the Platform Administrator, you want to introduce platform configuration and security guidelines to be used by your team to create a well-tuned and secure BI enterprise environment for the users.

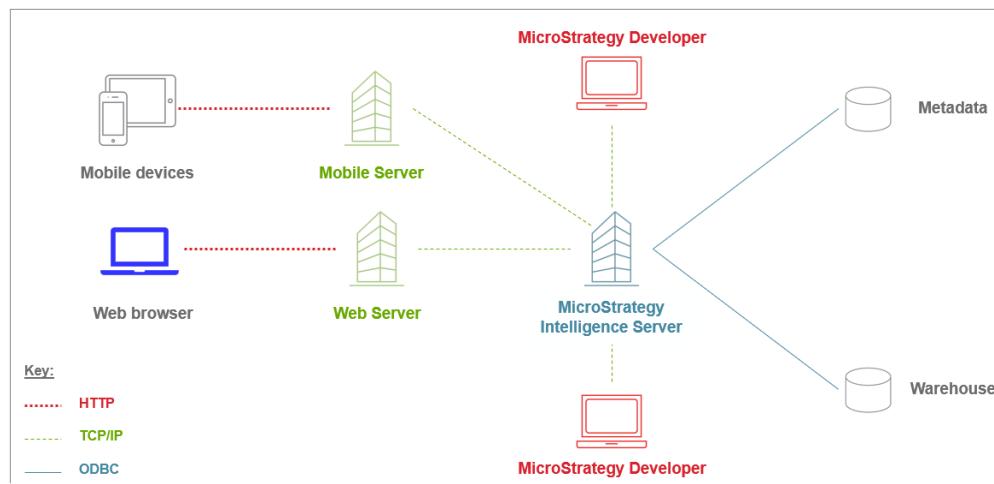
In this chapter, we will review:

- Configuration of connectivity between different components in the MicroStrategy analytics environment
- The platform security architecture, including MicroStrategy security model and security-related best practices.
- Configuration of Distribution Services to ensure a reliable and efficient distribution system

Configuring connectivity

After the MicroStrategy products have been installed, the Platform Administrator needs to ensure that his team establishes connectivity between various platform components in the analytics platform environment. The following image shows a

summary of connectivity in a four-tier architecture using both MicroStrategy Web and MicroStrategy Mobile:



For example, the platform administration team needs to:

- Configure a server definition to establish connectivity between the Intelligence Server and the metadata.
- Create database instances, database connections, and database logins to establish connectivity between the Intelligence Server and the data sources.



The MicroStrategy platform comes with many predefined connectors but it also offers flexibility for configuring custom connectors. In addition, the platform also offers Data Import functionality to connect with a wide variety of data sources such as Twitter, Salesforce, cloud data sources, big data sources, databases, and others.

- Establish connectivity between the Intelligence Server and the MicroStrategy Web server through the MicroStrategy Web Administrator page.
- Establish connectivity between the Intelligence Server and the MicroStrategy Mobile Server through the MicroStrategy Mobile Administrator page.
- If needed, configure internationalization (for data, metadata, Web, and Mobile Servers) to ensure that the analytics environments are available to a multilingual audience.

As the Platform Administrator, you should work with the System Administrator, Analytics Architect, and other Intelligence Center users in developing policies, rules, and standards that govern which data is collected, how it is stored, arranged, integrated, and used. In addition, you should develop standards for the

creation of server definitions and other objects to enable your team to consistently create supporting objects in any MicroStrategy environment.

For example, you should:

- Identify the data sources that will need to be accessed in the MicroStrategy environments
- Recommend tools that will be used to create DSN connections such as Connectivity Wizard
- Recommend the ports that should be used by Intelligence Server, MicroStrategy Web server, and other components in your environment
 - Where feasible, you should use the default ports for MicroStrategy services and processes to avoid any issues.
- Develop the naming conventions that should be used for server definition, DSNs, and other objects
- Assign the login credentials that should be used by the Intelligence Server to connect to the data sources
- Identify the languages and other parameters that will be required to support internationalization

Best Practice

Best practices for creating data connectors

1 The platform administration team should use the MicroStrategy Connectivity Wizard when creating DSN connections for MicroStrategy drivers. This tool is specifically designed to configure connectivity to all data sources supported by the MicroStrategy platform. It offers the following advantages over other tools (such as the Microsoft ODBC Administrator):

- It lists only the database drivers supported by the MicroStrategy platform, while other tools list all database drivers that are installed on a machine. If you use other non-MicroStrategy tools, you need to know all the supported MicroStrategy platforms beforehand.
- It lists only the DSN settings you are required to configure, while other non-MicroStrategy tools list all the settings, including the ones that are optional.
- It automatically configures certain parameters which are necessary for the optimal functioning of the MicroStrategy driver.

■ In case of non-MicroStrategy-related drivers, you may use either MicroStrategy Connectivity Wizard or other tools (such as Microsoft

ODBC Administrator in Windows environments) for creating DSNs. However, you should only use MicroStrategy Connectivity Wizard when creating DSNs for MicroStrategy drivers. Alternatively, you can also update the odbc.ini file.

- 2 Develop and follow good naming conventions for the DSN and DSN-less connections. For example, you could have all DSNs for production environments in the following format: PRD_DataSourceName.
- 3 Unless necessary, the platform administration team should use DSN instead of DSN-less connections.

DNS-less connections are backed up as part of the metadata. If your Development (DEV), User Acceptance Test (UAT), or Production (PROD) environments point to different databases, then copying the metadata from one environment to another would require editing these DNS-less definitions to ensure they point to the correct databases. As a result, while DSN-less connections provide for an easy way for connections to be defined on the fly, they are harder to maintain compared to DSNs and lead to deviations from using standard naming conventions.

Additionally, making configuration changes to a DSN-less connection is more complex compared to DSN connections for which you can simply update the odbc.ini file.



A DSN-less connection is typically used in reference to the data import feature which enables you to import data from sources such as Excel, CSV and databases into MicroStrategy without any architecting steps. Using a DSN-less connection, you can directly connect to a database without having to create a physical database connection (DSN) on the MicroStrategy Web server or Intelligence Server machine.

Exercise 4.1: Automate platform-warehouse connectivity

As you have learned in this chapter, you need a database instance to establish connectivity between your project and your data warehouse. You can use Developer to create your database instance manually or you can automate the process using Command Manager scripts.

In this exercise, you perform a series of administrative tasks related to the creation of a database instance and establishing connectivity between the MicroStrategy Tutorial project and your data warehouse.

In this workflow, you perform the following tasks:

- Create a DSN named PlatAdmin_WH to your data warehouse. Although you can create a DSN using Connectivity Wizard, you want the platform administration team to simplify the DSN creation task by directly updating the odbc.ini file on the Linux machine using WinSCP.

You will use WinSCP which provides a GUI to easily access odbc.ini and other files on the Linux machine.

- Automate the creation of a database instance, including the database connections and the database logins, by using a Command Manager script. You will create two separate database logins; you will be using one of these database logins for connection mapping in a subsequent exercise.

Use the following information for creating:

- First database login:
 - Name: **PlatAdminDBLogin1**
 - Login: **ffdemo**
 - Password: Use the password listed in your MicroStrategy Cloud email
- Second database login:
 - Name: **PlatAdminDBLogin2**
 - Login: **mstr**
 - Password: Use the password listed in your MicroStrategy Cloud email
- Database connection:
 - Name: **PlatAdminDBConnection**

- ODBCDSN: **PlatAdmin_WH**
- DEFAULTLOGIN: **PlatAdminDBLogin1**
- Database instance:
 - Name: **PlatAdminDBinstance**
 - DBConntype: **PostgreSQL**
 - DBConnection: **PlatAdminDBConnection**
- Using Developer, assign the PlatAdminDBinstance to the MicroStrategy Tutorial project.

Create the DSN

You will first create the DSN and later use that DSN for creating the database connection. To access the odbc.ini file using WinSCP.

Initiate a ftp session using WinSCP

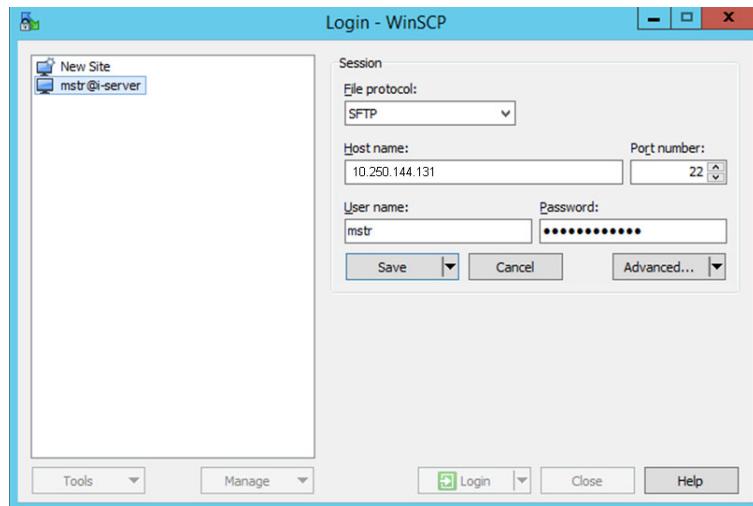
You will launch WinSCP on the Windows machine of your cloud environment.

- 1 Connect to your remote Windows machine, and double-click **WinSCP** on the desktop to launch the application.



- 2 In the Host name box, type the IP address of the Intelligence Server (listed in the **hosts** file on your desktop).

- 3** In the User name and Password boxes, enter the User name and password provided in the MicroStrategy Cloud email.



- 4** Click **Save**, click the **Save password** check box and then save your session information for future use.

- 5** Click **Login**.

If a message window opens, click **Yes**.

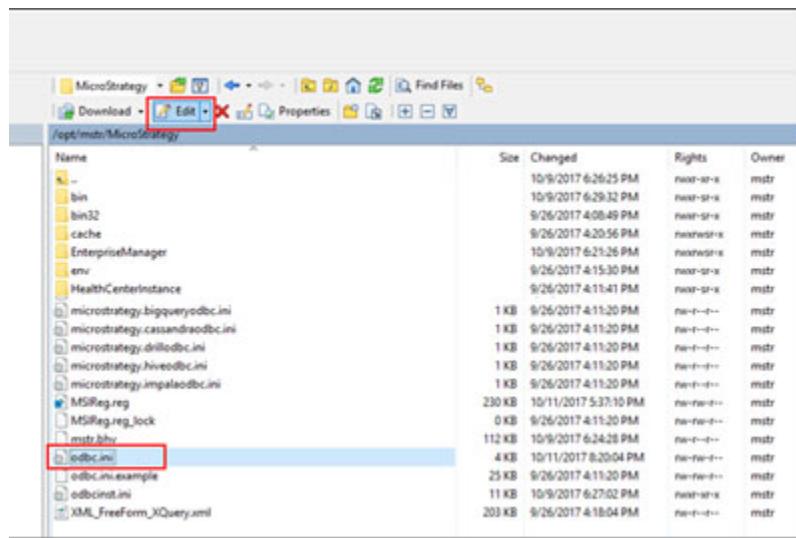
- 6** In the Authentication Banner window, click **Continue**.



- 7** In the WinSCP window, in the drop down list on top of the right pane, from the root directory, browse to **/opt/mstr/MicroStrategy**.



- 8 In the WinSCP window, in the right pane, select **odbc.ini** and on the top of the right pane, click **Edit**.



odbc.ini displays in the text editor.

- 9 In the **odbc.ini** file, under **[ODBC Data Sources]** but above **[ODBC]**, enter the following information:

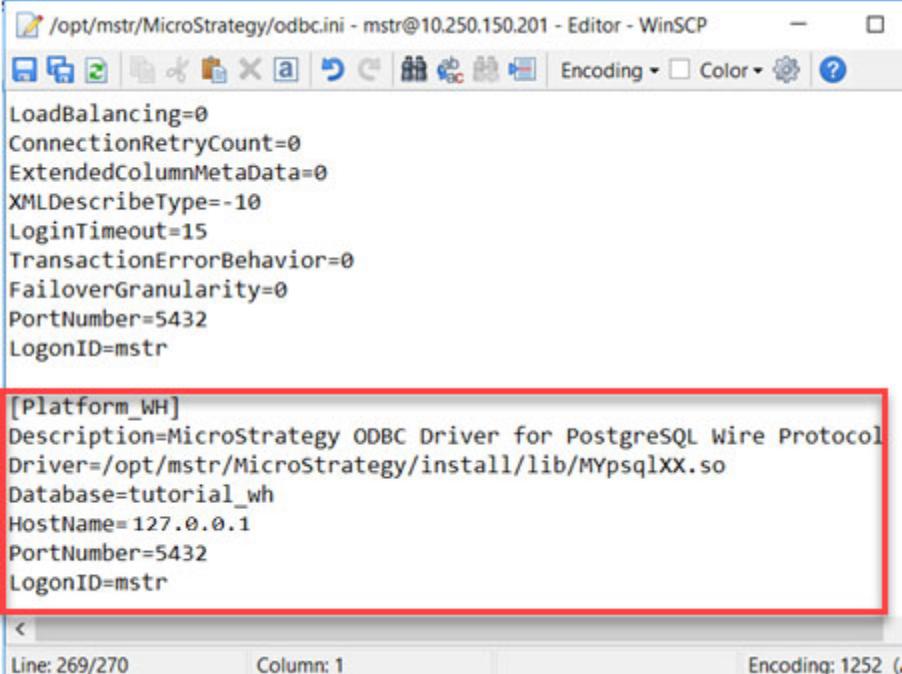
PlatAdmin_WH=MicroStrategy ODBC Driver for PostgreSQL
Wire Protocol

```
[ODBC Data Sources]
USHER_INTEL_WH=MySQL ODBC 5.x Driver
USHER_INTEL_DEMO_WH=MySQL ODBC 5.x Driver
METADATA=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
TUTORIAL_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
OPERATIONAL_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
PLATFORM_ANALYTICS_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
HRAM_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
ADVWD_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
FORECAST DATA=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol
Platform_WH=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol

[ODBC]
Trace=0
TraceFile=odbctrace.out
TraceDll=/opt/mstr/MicroStrategy/install/lib/MYtrcXX.so
InstallDir=/opt/mstr/MicroStrategy/install
IANAAppCodePage=106
UseCursorLib=0
UNICODE=UTF-8
```

10 At the end of the odbc.ini file, enter the following information:

```
[PlatAdmin_WH]  
  
Description=MicroStrategy ODBC Driver for PostgreSQL  
Wire Protocol  
  
Driver=/opt/mstr/MicroStrategy/install/lib/  
MYsqlXX.so  
  
Database=tutorial_wh  
  
HostName=127.0.0.1  
  
PortNumber=5432  
  
LogonID=mstr
```



The screenshot shows a WinSCP Editor window displaying the contents of the /opt/mstr/MicroStrategy/odbc.ini file. The file contains several connection parameters and two DSN definitions. The second DSN definition, which includes the new configuration, is highlighted with a red rectangular box.

```
LoadBalancing=0  
ConnectionRetryCount=0  
ExtendedColumnMetaData=0  
XMLDescribeType=-10  
LoginTimeout=15  
TransactionErrorBehavior=0  
FailoverGranularity=0  
PortNumber=5432  
LogonID=mstr  
  
[Platform_WH]  
Description=MicroStrategy ODBC Driver for PostgreSQL Wire Protocol  
Driver=/opt/mstr/MicroStrategy/install/lib/MYsqlXX.so  
Database=tutorial_wh  
HostName=127.0.0.1  
PortNumber=5432  
LogonID=mstr
```



In the above syntax:

- PlatAdmin_WH is the name of the DSN
- PostgreSQL ODBC is description of the driver
- 127.0.0.1 represents the localhost. In your environment, both IS and database servers are on the same machine.
- 5432 represents the value of the TCP/IP port number for PostgreSQL server

- tutorial_wh is the name of your data warehouse
- /opt/mstr/MicroStrategy/install/lib/MYpsqlXX.so is the value of the driver and the installation path for the driver

11 Save the **odbc.ini** file. Close the file.

You have now created a DSN to your data warehouse.

12 Exit WinSCP.

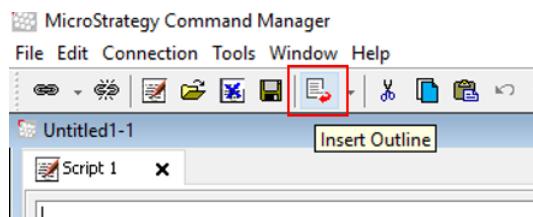
Create the database instance

You now use a Command Manager script to automate the creation of database logins, database connection, and database instance. Using a single Command Manager script, you can create multiple objects in a matter of seconds. The Command Manager script can be executed in real time or you can schedule it to run based on an event or at specified time.

When users run any queries in a MicroStrategy project against the data warehouse, the Intelligence Server uses the information embedded in the database instance associated with the project to connect to the data warehouse.

Create the Command Manager script

- 1 Using the search box on the taskbar find and launch **Command Manager**.
- 2 For the Project Source select **MicroStrategy on AWS I-Server**.
- 3 Use the credentials from your Welcome email to log in.
- 4 On the toolbar, click the **Insert Outline** icon to open the Choose Outline window.



- 5 In the Choose Outline window, expand the **DBLogin_Outlines** folder and select **Create_DBLogin_Outline**, then click **Insert**.

6 Customize the syntax to the following:

```
CREATE DBLOGIN "PlatAdminDBLogin1" LOGIN "ffdemo"  
PASSWORD "<password>";
```

```
CREATE DBLOGIN "PlatAdminDBLogin2" LOGIN "mstr"  
PASSWORD "<password>";
```



Replace <password> with the password listed in your MicroStrategy Cloud email.

```
Script 1  
CREATE DBLOGIN "PlatAdminDBLogin1" LOGIN "ffdemo" PASSWORD "Qwerty123456";  
CREATE DBLOGIN "PlatAdminDBLogin2" LOGIN "mstr" PASSWORD "Qwerty123456";
```

Create the database connection

As a database connection is part of the database instance, it must be created before creating the database instance.

- 1 While still in the same Script window, on the toolbar, click **Insert Outline**.
- 2 In the Choose Outline window, expand the **DBConnection_Outlines** folder and select **Create_DBConnection_Outline**. Click **Insert**.
- 3 Modify the syntax to:

```
CREATE DBCONNECTION "PlatAdminDBConnection" ODBCDSN  
"PlatAdmin_WH" DEFAULTLOGIN "PlatAdminDBLogin2";
```



You will use database login PlatAdminDBLogin1 in a subsequent exercise on connection mapping.

- 4 While still in the same Script window, on the toolbar, click **Insert Outline**.
- 5 In the Choose Outline window, expand the **DBInstance_Outlines** folder and select **Create_DBInstance_Outline**. Click **Insert**.

6 Modify the script to:

```
CREATE DBINSTANCE "PlatAdminDBinstance" DBCONNTYPE  
"PostgreSQL" DB CONNECTION "PlatAdminDBConnection";
```



```
CREATE DBLOGIN "PlatAdminDBLogin1" LOGIN "fffdemo" PASSWORD "Oqjlkj2345jk";  
CREATE DBLOGIN "PlatAdminDBLogin2" LOGIN "mstr" PASSWORD "Oqjlkj2345jk";  
CREATE DBCONNECTION "PlatAdminDBConnection" ODBCDSN "PlatAdmin_WH" DEFAULTLOGIN "PlatAdminDBLogin2";  
CREATE DBINSTANCE "PlatAdminDBinstance" DBCONNTYPE "PostgreSQL" DB CONNECTION "PlatAdminDBConnection";
```

Check syntax and run the script

- 1 On the toolbar, click **Check Syntax**. If an error message displays, correct the syntax of the script and then re-check it.
- 2 Execute the script. A message displays on the Messages tab specifying that the various objects have been created.
- 3 Create a directory on the C:\ drive named **MSTR**.
- 4 Save the script in **C:\MSTR** and name it **DBInstanceObjects**.
Command Manager saves the script as a Script File with a .scp extension.
- 5 Close the script but leave the Command Manager open as you will use it again for a subsequent exercise.

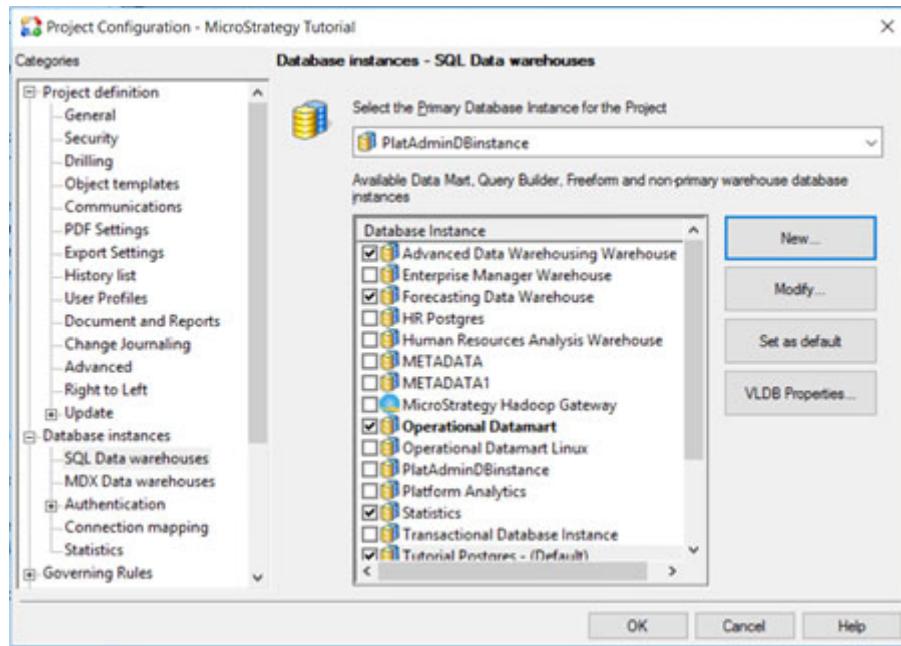
Assign database instance to the project

You will now assign your PlatAdminDBinstance database instance to the MicroStrategy Tutorial project so as to use it for establishing connectivity between the Intelligence Server and the data warehouse.

You use Project Configuration Editor to assign database instance to your project.

- 1 In Developer, in the **MicroStrategy on AWS I-Server** project source, right-click **MicroStrategy Tutorial** and select **Project Configuration**.
- 2 In the Project Configuration Editor, under **Database instances** select **SQL Data warehouses**.

- 3 On the right pane, select the **PlatAdminDBinstance** database instance and click **OK**.



- 4 Click **OK** in the message window.

Disconnect and reconnect to the project source for the database instance change to take effect.

- 5 Right-click **MicroStrategy on AWS I-Server** and select **Disconnect from Project Source**.
- 6 Log back in to the **MicroStrategy on AWS I-Server** project source as the **mstr** user.

The MicroStrategy Tutorial now uses PlatAdminDBinstance as the primary database instance.

Verify use of the new database instance and database login

Run a report in Developer to confirm that Intelligence Server uses the new database instance and database login.

- 1 In the **MicroStrategy Tutorial** project, navigate to the **Public Objects\Reports\Subject Areas\Human Resources Analysis** folder.

2 Execute **Call Center Timeliness** and then switch report view to **SQL View**.

Report:	Call Center Timeliness
Job:	222
Data Rows:	15
Data Columns:	1
Report Cache Used:	No
Query Engine Execution Start Time:	8/1/2018 8:54:34 PM
Query Engine Execution Finish Time:	8/1/2018 8:54:38 PM
Number of Rows Returned:	15
Number of Columns Returned:	3
Number of Temp Tables:	0
Total Number of Passes:	2
Number of Datasource Query Passes:	2
Number of Analytical Query Passes:	0
DB User:	PlatAdminDBLogin2
DB Instance:	PlatAdminDBinstance
Tables Accessed:	lu_call_ctr

3 Close the report, without saving it.

Fix the database connection character encoding

If the character set encoding of the database server and the client do not match, your data may not display correctly. To configure the character set encoding you can access your database connection via the database instance editor.

- 1 Expand **Administration, Configuration Manager**, then select **Database Instance**.
- 2 Right-click **PlatAdminDBinstance** and select **Edit**.
- 3 In the Database connection (default) pane, select **PlatformAdminDBConnection** and click **Modify**.
- 4 Select the **Advanced** tab.
- 5 Select **UTF-8** for both Windows and Unix drivers.
- 6 Click **OK** to close the database instance editor.

Ignore the warning message about restarting the Intelligence Server. You will change the database instance in a later exercise.

Implementing platform security

An Enterprise BI platform interacts directly with your critical business data, so it's essential to have security measures in place that protect sensitive information at every level of the BI architecture. As MartZon transitions to a secure Intelligent Enterprise, it is your responsibility as the Platform Administrator to develop guidelines for addressing the security requirements of your organization's analytics environment.

You collaborate with other members in the Intelligence Center, developing the framework and processes in your BI environment that protect your organization's data from unauthorized viewing, tampering, or modification.

The workflows and processes you define, deliver security across three areas:

- **User authentication**—Implementing an authentication mode for the analytics platform to identify users and manage access rights. Several authentication modes are supported in the MicroStrategy environment, from standard authentication to linked authentication with third-party security systems to integration with single sign-on solutions.
- **User authorization**—Defining the guidelines for implementing role-based access control to enforce security policies governing the functionality, data, and application objects for which the user is authorized.
- **Data transmission security**—Configuring data transmission settings (such as ports and SSL & TLS certificates) in the MicroStrategy platform to encrypt communications across the entire analytics ecosystem.

Managing user authentication

Intelligent Enterprises employ a strong defense approach, using several layers of security throughout the IT system. As the Platform Administrator, you want to implement those layers of security to safeguard environment applications and data at all times. The first line of defense in securing enterprise applications is user authentication.

Authentication is the process by which the system identifies the user. In most cases, a user provides a login ID and password which the system compares to a list of authorized logins and passwords. If they match, the user is able to access certain aspects of the system, according to the access rights and application privileges associated with the user. The login credentials and other user profile information for a MicroStrategy user can be stored in the MicroStrategy metadata or in an external repository, such as an LDAP directory.

This flexibility allows you to use an external authentication mechanism to authenticate a user. By synchronizing MicroStrategy applications with existing corporate directories, group membership and security is maintained in the LDAP directory and your administrators do not need to recreate user logins. You can link or import LDAP users and groups into MicroStrategy.

A broad range of authentication modes are supported in the MicroStrategy Enterprise Platform. The main difference between the modes is the authentication authority used by each mode. The authentication authority is the system that verifies and accepts the login/password credentials provided by the user.

Listed below are the available authentication types for the MicroStrategy platform.

- **Standard**— In the standard (default) authentication mode, the administrator is responsible for creating the users in the MicroStrategy metadata and the Intelligence Server is the authentication authority. When users access a project, they are prompted to enter their MicroStrategy user login ID and password for initial system access.
- **Anonymous**— When using anonymous authentication, users log in as guests and do not need to provide a password. Each guest user assumes the profile defined by the Public group.

Anonymous authentication can be useful when all users in your environment need the same privileges for a project and can view the same data from the data source. For example, a public health analytics environment that allows users to access readily available Centers for Medicare and Medicaid data to compare cancer rates among different states.

- **Database Warehouse**—The data source is the authentication authority. Users can log into a MicroStrategy project using their data source credentials. As a result, you can leverage the security views set up at the data source level during data retrieval.
- **LDAP (Lightweight Directory Access Protocol)**— An LDAP server is the authentication authority. If you use an LDAP directory to centrally manage users in your environment, you can implement LDAP authentication in MicroStrategy.

Group membership can be maintained in the LDAP directory without having to also be defined in MicroStrategy. LDAP authentication identifies users in an LDAP directory which MicroStrategy can connect to through an LDAP server.

Supported LDAP servers include Novell Directory Services, Microsoft Active Directory, OpenLDAP for Linux, and Sun ONE 5.1/iPlanet.

- **Windows Authentication**— Users are not prompted to enter a login ID and password. The user's Windows account is linked to a MicroStrategy user. The system identifies users by the Windows network login ID with which they are logged in to the Windows network. This mode is enabled for Windows-based Web servers.
- **Single sign-on**—Single sign-on (SSO) allows enterprise network users to access all authorized network resources seamlessly, on the basis of a single authentication that is performed when they initially access the network. SSO encompasses several different third-party authentication methods, including:
 - **SAML Authentication**—SAML is a two-way setup between your MicroStrategy application and your Identity Provider (IdP). SAML support allows MicroStrategy to work with a wide variety of SAML identity providers for authentication.

To configure a MicroStrategy application for SAML authentication, you will need to create SAML configuration files for your application, register the application with your IdP, establish trust to MicroStrategy Intelligence Server, and link SAML users to MicroStrategy users.

- **Integrated authentication**—A domain controller using Kerberos authentication is the authentication authority. Users log in once to their Windows machine and do not need to log in again to either MicroStrategy Web or Developer.
- **Trusted Authentication Request** (or third-party authentication)—Once a user is authenticated in the third-party system, their permissions are retrieved from a user directory, such as LDAP, and access is granted to the MicroStrategy application.

Trusted authentication providers include Tivoli, SiteMinder, Oblix, PingFederate, and Oracle Access Manager.

The following table lists the available authentication modes in MicroStrategy client applications:

MicroStrategy Web	MicroStrategy Mobile	MicroStrategy Library	MicroStrategy Workstation
Anonymous	Anonymous	Anonymous	
Database			
Integrated	Integrated	Integrated	
LDAP		LDAP	LDAP
SAML	SAML	SAML	
Standard	Standard	Standard	Standard
Trusted			
Windows	Windows		

Configuring LDAP authentication

An LDAP authentication system consists of two components: an LDAP server and an LDAP directory. An LDAP server is a program that implements the LDAP protocol and controls access to an LDAP directory of user and group accounts. An LDAP directory is the storage location and structure of user and group accounts on an LDAP server.

The following section describes a high-level overview of the general flow of information between Intelligence Server and an LDAP server when an LDAP user logs into a MicroStrategy client application such as Developer, Web, Library, or Workstation.

LDAP user login information flow

- 1 When an LDAP user logs in to a MicroStrategy client application, Intelligence Server connects to the LDAP server using the credentials for the LDAP administrative user, called an authentication user.
- 2 The authentication user is bound to LDAP using a Distinguished Name (DN) and password set up in the user's configuration.
- 3 The authentication user searches the LDAP directory for the user who is logging in via MicroStrategy client application, based on the DN of the user logging in.
- 4 If this search successfully locates the user who is logging in, the user's LDAP group information is retrieved.

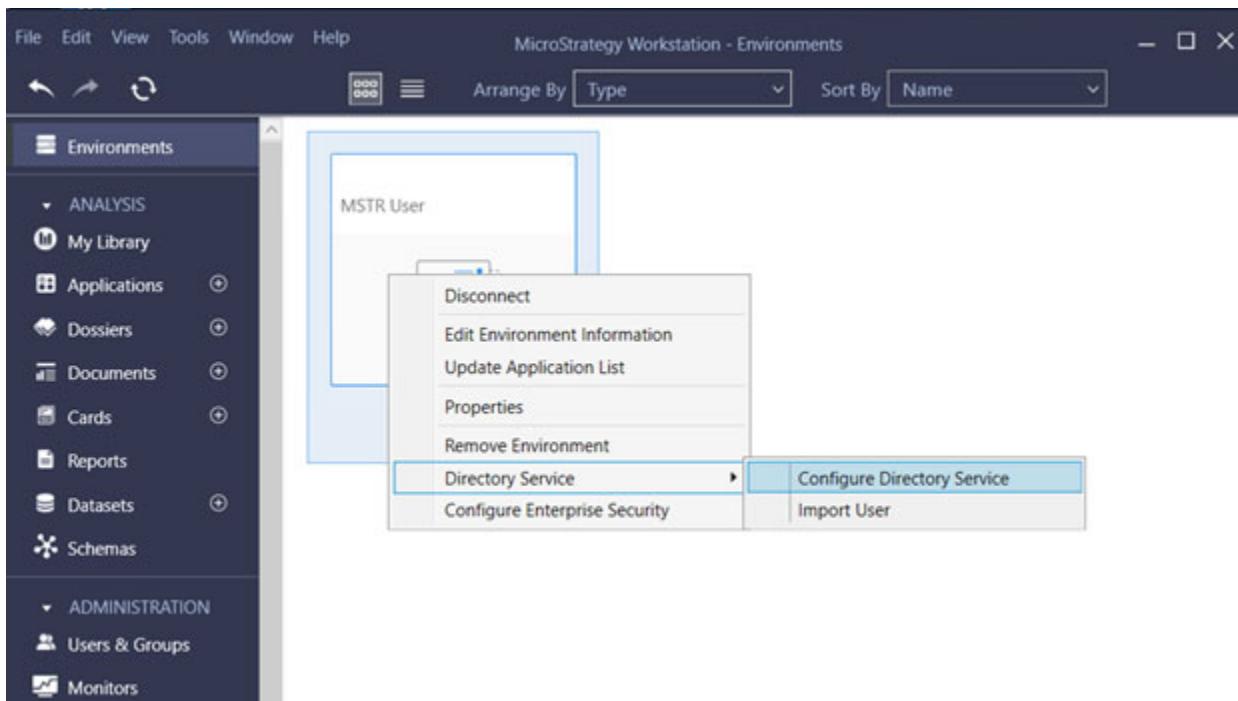
- 5 Intelligence Server then searches the MicroStrategy metadata to determine whether the DN of the user logging in is linked to an existing MicroStrategy user or not.
- 6 If a linked user is not found in the metadata, Intelligence Server refers to the import and synchronization options that are configured. If importing is enabled, Intelligence Server updates the metadata with the user and group information it accessed in the LDAP directory.
- 7 The user who is logging in is given access to MicroStrategy, with appropriate privileges and permissions.

Connecting LDAP Server to MicroStrategy using LDAP Connectivity Wizard

Before information from an LDAP directory can be searched and retrieved, a connection to the LDAP server must be established. When you have collected the connection information for your LDAP server and your LDAP SDK, you can use the LDAP Connectivity Wizard to set up your LDAP connection. The LDAP Connectivity Wizard helps step you through the initial setup of using your LDAP server to authenticate users and groups in MicroStrategy.

You can use MicroStrategy Workstation to access the LDAP Connectivity Wizard and connect your LDAP server with the Intelligence Server.

To launch the wizard, you connect to your environment, and select Configure Directory Server.



For example, if integrating with an LDAP provider, you will need parameters such as the following:

- **Host**—The host name or the IP address of the LDAP server.
- **Port**—Port number of the LDAP server. Port 389 is the default for the clear text LDAP, and Port 636 is the default for SSL.
- **Base Distinguished Name**—The trusted LDAP Authentication User used by Intelligence Server to access, search in, and retrieve information from the LDAP directory when authenticating, importing, and synchronizing new user accounts. The distinguished name (DN) for the trusted LDAP Authentication User is used to search the LDAP repository

The Authentication User must be set up in the LDAP repository before configuring LDAP settings in MicroStrategy. The user must have, at a minimum, read and search access rights to the required user and group objects.

- **SSL**—Whether the LDAP server is accessed using clear text, or over an encrypted SSL connection. If you select SSL, you will need to provide a valid certificate from your LDAP server and save it on the machine where Intelligence Server is installed.

Configure Directory Service

General

Attribute Mapping

Advanced

Server Information

Host* localhost

Port* 636

Directory Server Type* Microsoft Active Directory

SSL

PEM Drag the Server Certificate File (pem) here or Select a File

Base Distinguished Name(DN)* Input Value...

Bind Distinguished Name(DN)* Input Value...

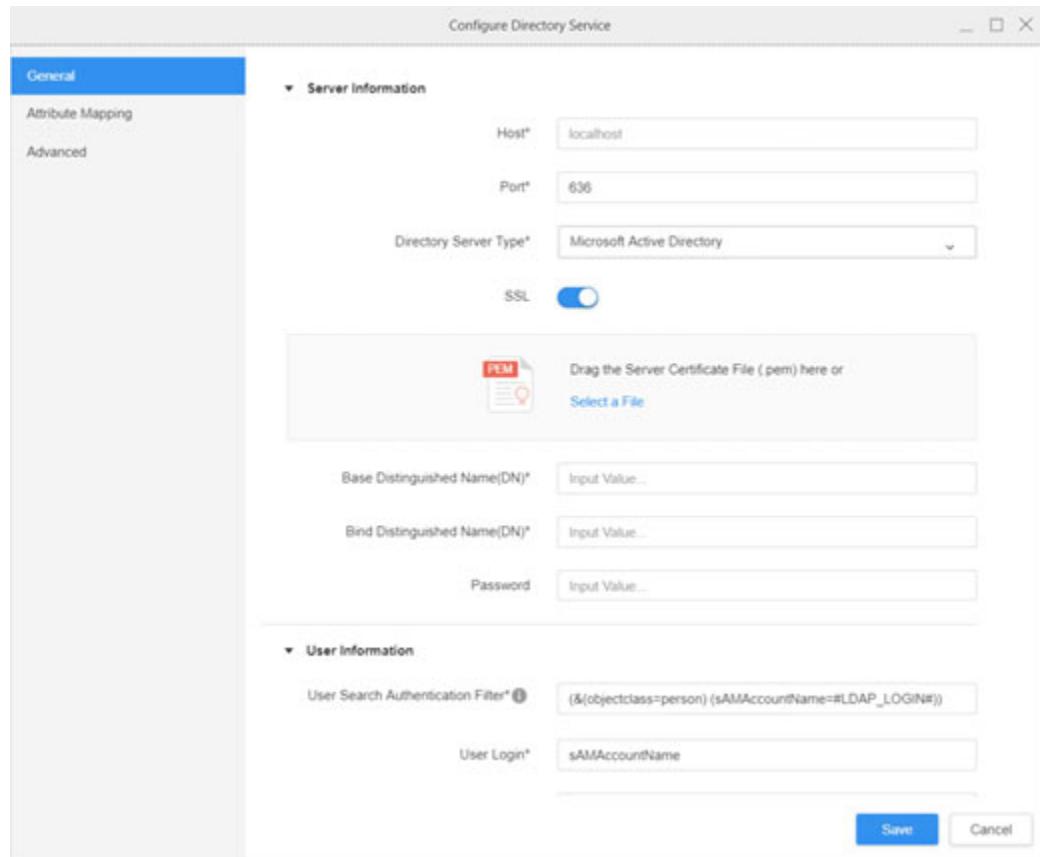
Password Input Value...

User Information

User Search Authentication Filter* (&(objectclass=person) (sAMAccountName=#LDAP_LOGIN#))

User Login* sAMAccountName

Save Cancel



- **Authentication User**—The user name and password of an LDAP user who can search the LDAP directory.
- **LDAP SDK Library**— Displays the location of connectivity file libraries (DLLs) that MicroStrategy uses to communicate with the LDAP server. This value is displayed on the Advanced tab.

Configure Directory Service

Name	Value
Reuse Connection	<input checked="" type="checkbox"/>
Search Timeout	120
Synchronize user/group information with LDAP during Windows authentication an...	<input checked="" type="checkbox"/>
Synchronize user/group information with LDAP during Trusted authentication	<input checked="" type="checkbox"/>
User login fails if LDAP attribute value is not read from the LDAP server	<input checked="" type="checkbox"/>
Import User	<input checked="" type="checkbox"/>
Synchronize User	<input checked="" type="checkbox"/>
Search User Import Filter ⓘ	<input type="button" value="Input Value..."/>
Import Group	<input checked="" type="checkbox"/>
Synchronize Group	<input checked="" type="checkbox"/>
Search Group Import Filter ⓘ	<input type="button" value="Input Value..."/>
LDAP SDK Library ⓘ	<input type="button" value="Input Value..."/>

Managing LDAP authentication

While implementing LDAP authentication, you may want to explore the following topics in more detail depending on your implementation requirements:

- Selecting schedules for importing and synchronizing users
- User privileges and security settings after import
- Using LDAP attributes in security filters
- Controlling project access with LDAP attributes
- Using LDAP with Single Sign-On authentication systems

When integrating the analytics platform with the third-party directory/identity providers, the Platform Administrator needs to work closely with the System Administrator to obtain the required parameters of the directory/identity provider.

Enabling LDAP authentication

Once you have established connectivity to your LDAP server, you need to enable LDAP authentication for the project source and each MicroStrategy client application as follows:

- 1 Enabling LDAP authentication for your **project source****
 - a In Developer, select **Modify Project Source**
 - b In the Advanced tab, select **Use LDAP authentication**
- 2 Enabling LDAP authentication for **MicroStrategy Web****
 - Navigate to Web Administrator URL
 - Select **Default Properties** for your Intelligence Server
 - **Enable LDAP authentication**
- 3 Enabling LDAP authentication for **MicroStrategy Library****
 - Navigate to Library Admin URL
 - Select the **Library Web Server** tab
 - Select **LDAP** in the list of available Authentication Modes
- 4 Enabling LDAP authentication for **MicroStrategy Workstation****
 - From the Environments tab, click **Add New Environment Connection**
 - Enter an environment name and the URL of the Library Server
 - Select **LDAP** as the Authentication Mode

Managing user security

MicroStrategy uses a multi-layered security model, with various security options at different levels. The MicroStrategy platform includes the following security features to control more nuanced access to the MicroStrategy application

functionality such as the ability to create reports, view or modify an object, ability to see data, and so on:

Level	Security Options
Project Source	<ul style="list-style-type: none">• Users and groups• Privileges• Security roles
Project	<ul style="list-style-type: none">• Security filters• Connection mappings
Object	<ul style="list-style-type: none">• Permissions (Access Control Lists, or ACLs)

The following diagram shows the various components of MicroStrategy application-level security:



Managing access to application functionality

The Platform Administrator should develop guidelines for managing access to the features or functionality once a user has logged into the MicroStrategy application. For security, performance, and licensing purposes, users should generally be provided access only to the application features that are required for their roles.

The platform administration team can use the following security options at the project source level to control access to the MicroStrategy application functionality:

- **Users**—MicroStrategy users access projects with their reports and documents, using client applications such as Developer and MicroStrategy

Web through user accounts. A user account is a metadata object that exists across projects.

- **Groups**—A group is a collection of users or other groups. Group accounts provide a convenient means for assigning privileges and object access to multiple users at one time.
- **Security roles**—A security role enables you to assign unique sets of privileges to users or groups on a project-by-project basis. You can also use this feature to determine whether a user or group can access a specific project.
- **User Privileges**—Privileges give users access to specific application functionality, such as creating a report, viewing report SQL, or using the administration monitors.

Privileges can be assigned to users and groups directly or through security roles. The difference is the following:

- Privileges assigned directly to users and groups grant functionality across all projects in the project source.
- Privileges assigned through security roles grant functionality just within a specific project
- **MicroStrategy Badge**—Digital badges provide an added layer of authentication for safeguarding physical and logical assets. Organizations can provide secure digital authentication using geo-fencing and time-fencing capabilities, thus restricting user access based on location and time.

Securing data access: Connection mapping and security filters

The Platform Administrator needs to develop guidelines and recommend features that his team can use to ensure that users are able to access only the data that they are authorized to view. The MicroStrategy platform provides the following security features that the platform administration team can leverage for controlling access to the data in the data sources:

- Connection mapping
- Security filters

Connection mapping

Connection mapping enables you to link users to different database connections and logins in a single project. By default, all users in a MicroStrategy project use the same database connection (DSN) and database login when connecting to the

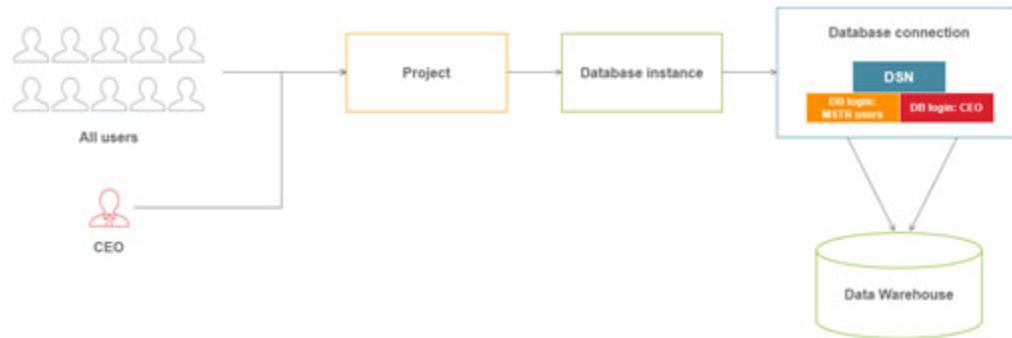
database, which is defined by the database instance you assigned to the project. This means that all users have the same security level on the database side.

No Connection Mapping—Different MicroStrategy requests appear as coming from the same entity



You can use connection mapping to differentiate MicroStrategy users from each other at the data warehouse level or if you need to direct them to separate data warehouses. For example, in the following image, the CEO user connects as CEO on the database side (using the new database login called CEO) and all other users use the default database login MSTR users.

Connection Mapping is applied—Requests from the CEO User are distinguished on the database side



Connection mapping is not required. You create it and assign it to users or groups only if your environment can benefit from it.

Security filters

A security filter is a filter object that you assign to users or groups to restrict the result set when they execute reports or browse elements. The criteria specified by the security filter is added to the WHERE clause of every SQL pass, for the jobs that the user executes where there is a relationship between the attribute that defines the security filter and the objects in the report.

The following images illustrate the use of security filters when an administrator with no security filter and a regional manager with the Northeast security filter run the same report:

With no security filter

Employee Revenue and Profit Report					
Region	Call Center	Employee		Profit	Revenue
Central	Milwaukee	Gale	Loren	\$253,254	\$1,669,290
		Torrison	Mary	\$259,485	\$1,690,350
		Zemlicka	George	\$124,807	\$822,500
	Fargo	Ellerkamp	Nancy	\$126,778	\$847,227
Mid-Atlantic	Washington, DC	Bernstein	Lawrence	\$158,930	\$1,060,632
		Folks	Adrienne	\$159,074	\$1,047,776
		Hollywood	Robert	\$155,195	\$1,026,874
	Charleston	Brown	Vernon	\$51,231	\$331,735
		Corcoran	Peter	\$49,395	\$325,147
		Ingles	Walter	\$34,588	\$229,439
		Smith	Thomas	\$33,368	\$221,379
		Young	Sarah	\$31,303	\$209,634
	Northeast	Boston	De Le Torre	Sandra	\$607,895
			Kieferson	Jack	\$584,933
			Sonder	Melanie	\$295,108

With Northeast security filter

Employee Revenue and Profit Report					
Region	Call Center	Employee		Profit	Revenue
Northeast	Boston	De Le Torre	Sandra	\$93,100	\$607,895
		Kieferson	Jack	\$87,470	\$584,933
		Sonder	Melanie	\$43,925	\$295,108
	New York	Kelly	Laura	\$357,994	\$2,350,720
		Sawyer	Leanne	\$368,219	\$2,411,912
		Yager	Beth	\$350,024	\$2,303,847

Best Practice

Security filters are applied at the project level, and are applied to all queries. This simplifies the administrative work, and provide a way for securing the data. It is always a good idea to review on a regular basis the security filter definitions and the users who have privileges to modify the security filters to ensure that a tight access to security filter editing is maintained. Typically in development environments, you should restrict developers access to security filter editing and during object migration, make sure to review the definition of the security filter as an additional step before importing a package.

You should also determine whether to control access to data using security filters or at the database level. One advantage of using the database-level security mechanisms (such as security views) to secure data is that all applications accessing the database benefit from those security measures.

Additionally, as a user who belongs to more than one group inherits multiple security filters, you should merge the security filters in such situations to meet the business requirements. Merging ensures data security, while simplifying and reducing costs of development.

Managing application functionality privileges through roles

Security roles enable you to assign application-specific functionality to users or groups by granting them sets of privileges on an application-by-application basis. Workstation allows you to grant access and roles for applications to users and user groups. You can create a group of users, providing a convenient way to manage a large number of users. You can access Security Roles in Workstation by:



The steps below are not to be performed as an in-class exercise. They are for reference only.

- 1 Launch Workstation from the Windows desktop of your RDP session.
- 2 Create a new environment connection by providing the connection URL to MicroStrategy Library.
- 3 Enter your environment credentials and connect to your environment.
- 4 From the left panel, select Users & Groups.
- 5 In the environment panel, select the environment you wish to control.
- 6 Select Security Roles, to view the available security roles in your environment.
- 7 To create a new security role, select the plus icon next to Security Roles.

- 8 To edit an existing security roles, right-click the role in the right pane, then select Properties.

The screenshot shows the MicroStrategy Workstation interface with the title "MicroStrategy Workstation - Users & Groups". The left sidebar has sections for ADMINISTRATION (Users & Groups, Monitors, Subscriptions, Certificates, Licenses, Scripts), RESOURCES (Getting Started, Community, Expert.Now), and a Help section. The main area is titled "ENVIRONMENT" with a dropdown set to "MartZon Environment". On the right, there is a table titled "Security Role" listing various roles with their icons:

Security Role
Analyst
Analytics Architect
Application Administrator
Application Architect
Certifier
Collaborator
Consumer
Customers
Database Architect
Embedded Analytics Architect

Instead of assigning privileges to hundreds of users individually, privileges can be assigned to all users at one time as a group. Privileges are assigned to users individually, through groups, or with security roles. Available security role assignments include:

- **Application Administrator:** Users granted this role have access to all application specific tasks.
- **Analyst:** Users granted this role have authoring capabilities.
- **Certifier:** Users granted this role can author and certify objects.
- **Collaborator:** Users granted this role can view and collaborate on dossiers or documents they have access to.

- **Consumer:** Users granted this role can view dossiers or documents they have access to.

Security Role Assignments

Privileges	Application Administrator	Certifier	Analyst	Consumer	Collaborator
Access Data from Databases, Google BigQuery, BigData, OLAP, BI tools	✓	✓	✓		
Add Notes	✓	✓	✓		
Assign security filter	✓				
Assign Security Role	✓				
Bypass all object security access checks	✓				
Can certify content	✓	✓			
Configure governing	✓				
Create and Edit Security Filter	✓				
Create Application Objects	✓	✓	✓		
Edit notes	✓	✓	✓		
Email Screenshot From Device	✓	✓	✓	✓	✓
Execute Report That Uses Multiple Data Sources	✓	✓	✓	✓	✓
Execute Transaction	✓	✓	✓	✓	✓
Export to .mstr File	✓	✓	✓	✓	✓

For a list of available security role assignments, you can refer to MicroStrategy product help, accessible through MicroStrategy Community.

Exercise 4.2: Use connection mapping to secure data access

MartZon IT uses security views in the data warehouse to restrict information access based on user permissions. Using the connection mapping feature of MicroStrategy, you can leverage security views implemented at the database level to establish data security.

In this exercise, you create a connection mapping between MicroStrategy users and the database login they use for accessing the data warehouse. Users in the MicroStrategy Architect group need to connect to the warehouse using a database connection called *data*. While users in the Mobile Users group need to connect to the warehouse using a database login called *PlatAdminDBLogin2*.

Using Command Manager scripts, you learn how to implement this requirement.

Create the Command Manager script

- 1 On the Windows machine, launch Command Manager and log in using your credentials from the Welcome email.
- 2 From the **Edit** menu, select **Insert / Search Objects**, then select **Outlines**.
- 3 In the left pane, expand the **Connection_Map_Outlines** and select **Create_Connection_Map_Outline**, then click **Insert**.
- 4 Customize the syntax to the following:

```
CREATE CONNECTION MAP FOR USER GROUP "MicroStrategy
Architect" DBINSTANCE "PlatAdminDBInstance"
DBCONNECTION "PlatAdminDBConnection" DBLOGIN "Data"
FOR PROJECT "MicroStrategy Tutorial";

CREATE CONNECTION MAP FOR USER GROUP "Mobile Users"
DBINSTANCE "PlatAdminDBInstance" DBCONNECTION
"PlatAdminDBConnection" DBLOGIN "PlatAdminDBlogin2"
FOR PROJECT "MicroStrategy Tutorial";
```

- 5 From the **Connection** menu, click **Check Syntax** to ensure the code has no errors.
- 6 Execute the script. A message displays on the Messages tab specifying that the various objects have been created.

7 Save the script to **C:\MSTR** and name it **ConnectionMap**.

8 Minimize Command Manager.

Verify the new connection map

You will now disconnect and reconnect to the project source in Developer for the connection map-related change to take effect, and then confirm the new connection map for the MicroStrategy Tutorial project.

- 1 In Developer, right-click **MicroStrategy on AWS I-Server** and select **Disconnect from Project Source**.
- 2 Log back in to the **MicroStrategy on AWS I-Server** project source as the **mstr** user.
- 3 Right-click the **MicroStrategy Tutorial** project and select **Project Configuration**. Under Database instances, select Connection mapping. You can see the connection map that you created using the Command Manager script.

Database Instance	User	Language	Database Connection	Database Login
PlatAdminDBinstance	Default Users	(Default)	PlatAdminDBConnection	PlatAdminDBLogin2
Advanced Data Warehousing Warehouse	Default Users	(Default)	ADWDW_WH	Data
Forecasting Data Warehouse	Default Users	(Default)	Forecast Data warehouse	Data
Operational Datamart	Default Users	(Default)	OpWHLinux	Data
PostgresWarehouseInstance	Default Users	(Default)	PostgresWarehouseConnection	PostgresWarehouseLogin
Statistics	Default Users	(Default)	STATISTICS	STATISTICS
webDAV Documents	Default Users	(Default)	webDAV Documents	webDAV Documents
XQuery	Default Users	(Default)	XQuery	Data
PlatAdminDBinstance	Mobile Users	(Default)	PlatAdminDBConnection	PlatAdminDBLogin2
PlatAdminDBinstance	MicroStrategy Architect	(Default)	PlatAdminDBConnection	Data

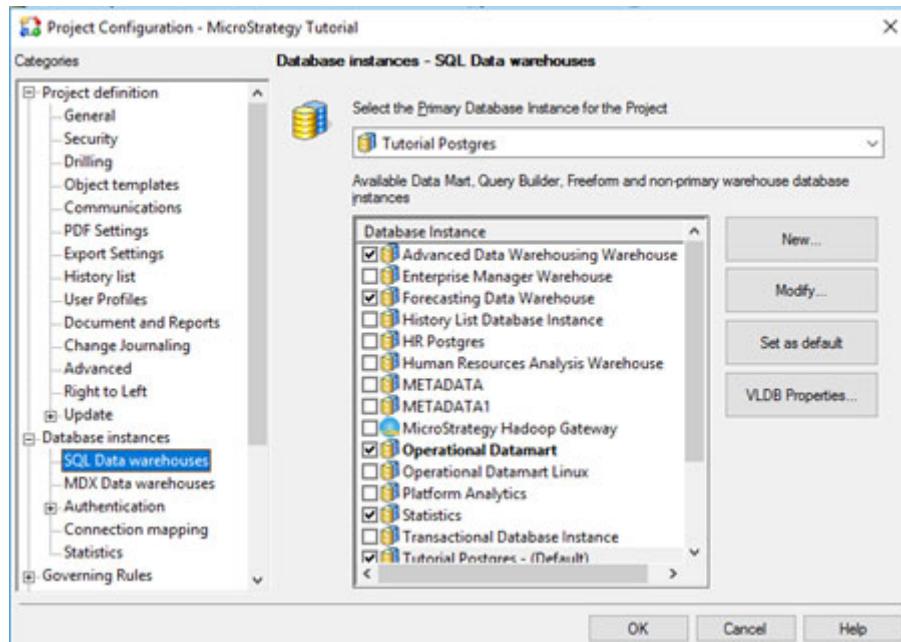
- 4 Click **OK** to close the Project Configuration window.

Update primary database instance for the Tutorial project

The rest of your exercises in this course require the PostgreSQL database instance. Change the primary database instance for the Tutorial project.

- 1 Open Project Configuration for the MicroStrategy Tutorial project.

- 2 In the Project Configuration Editor, under **Database instances**, select **SQL Data warehouses**.
- 3 In the Select the Primary Database Instance for the Project drop-down, select **Tutorial Postgres** and click **OK**.



- 4 In the message window, click **OK**.
- 5 Disconnect, then reconnect to the Project Source again.

Managing access to MicroStrategy objects: Permissions and ACLs

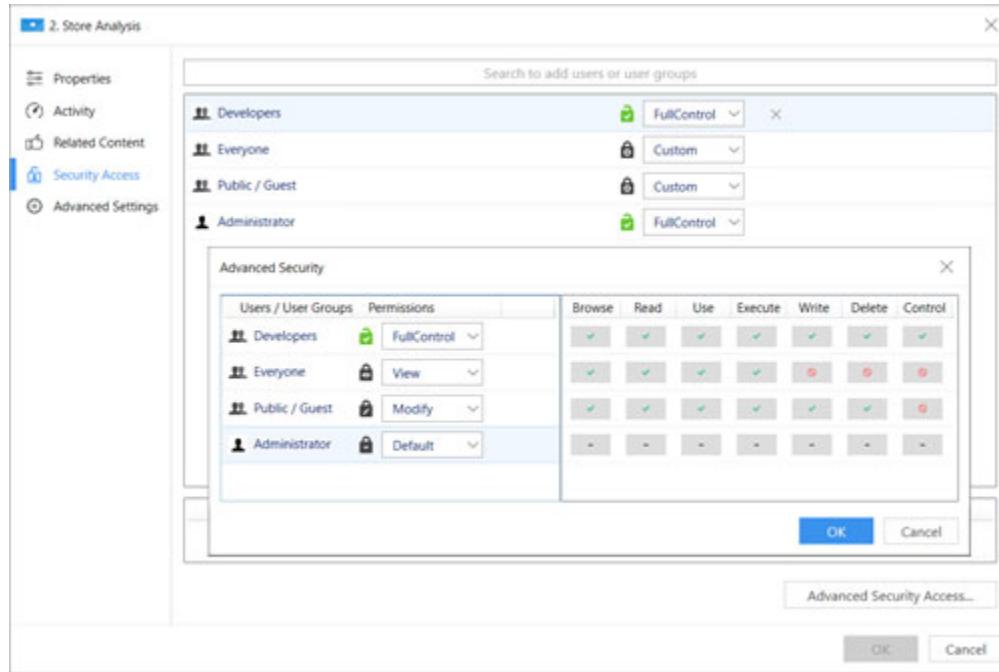
In addition to ensuring data security, the Platform Administrator needs to develop guidelines and recommend tools that his team can use to secure access to the objects in the metadata. The MicroStrategy platform provides the following security features that the platform administration team can leverage for controlling access to the MicroStrategy objects:

- **Permissions**—Define the type of access users have over individual objects or folders in the system.
- **Access Control List (ACL)**—Represents a list of users and groups and the access permissions that each of them has for an object. When granting permissions to objects, you must specify the following elements:
 - **User**—Defines what users and groups can access the object.
 - **Object**—Defines the object permission for the user. For example you can assign View access for some users and Modify access for other users.
 - **Children (folders only)**—Defines the object permission for the objects that belong to that folder.

The Platform Administrator needs to ensure that his team has a good understanding of privileges, permissions, and ACLs. While privileges are assigned to users (individually, through groups, or with security roles), permissions are assigned to objects. Permissions define the degree of control users have over individual objects within a MicroStrategy environment, such as the abilities to browse, read, write, control, use, or execute. Permissions are assigned to objects, and each object has an Access Control List (ACL) that specifies which permissions different sets of users have on that object. For example, a user may have permission to view a dossier, but not to delete or modify its definition. Object permissions can be defined in Developer, Workstation, and Web.

Within Workstation, you can set permissions for objects and folders, including all objects within a folder.

Controlling Access to Objects



A user can have permissions directly assigned to an object, and be a member of one or more groups that have a different permission grouping assigned to the object. In this case, user-level permissions override group-level permissions, and permissions that are denied at the user or group level override permissions that are granted at that level. The list below indicates what permissions are granted when permissions from multiple sources conflict:

- Permissions that are directly denied to the user are always denied.
- Permissions that are directly granted to the user, and not directly denied, are always granted.
- Permissions that are denied by a group, and not directly granted to the user, are denied.
- Permissions that are granted by a group, and not denied by another group or directly denied, are granted.
- Any permissions that are not granted, either directly or by a group, are denied.

Best Practice

Platform security best practices

The Platform Administrator should implement the following security-related best practices:

- 1 Collaborate with Intelligence Center users**—Define the requirements for the MicroStrategy platform security configuration in collaboration with the Intelligence Center architects, administrators, and other users.
- 2 Define the security configuration requirements**—When defining the MicroStrategy security configuration requirements, you should:
 - List all your data sources, including any security mechanism defined at the data source-level. For example, you may have defined security views at the data warehouse level.
 - Define the type of access different users will need. For example, the users in the Payroll department may need access to the salary data of everyone in the Executives group, while those in the Operations department may not have access to the salary data of those in the Executives group.
 - Define the approval workflow to be used for assigning authorizations to different users.
- 3 Create a security model matrix**—For documentation and visual representation purposes, it is recommended that you create a security model matrix listing the type of privileges different users or groups will have in different projects. An example of a security matrix is shown in the following image:

Project	Security Role	User Group
Finance	Web-Basic	FPA Group
	Web-Designer	FPA Development
	Web-Power User	Audit Group
Sales	Web-Basic	Sales Group
	Web-Designer	Pre-Sales Group
	Web-Power User	Sales Directors Group

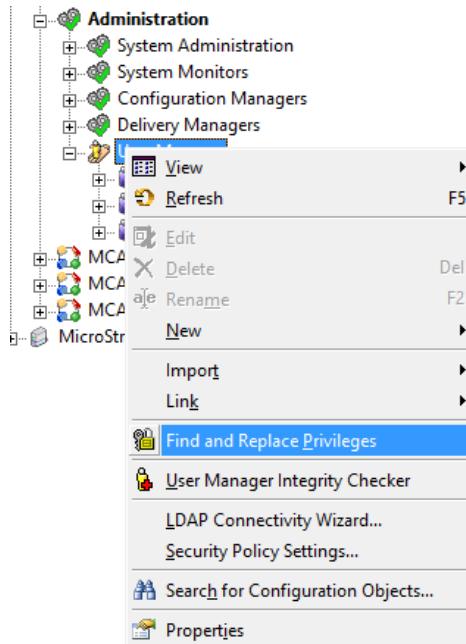
- 4 Simplify security management**—Simplify and automate modes of authentication used by different groups of users where possible. For example, if your organization uses LDAP but the MicroStrategy platform is using standard authentication, then it may be a good idea to use LDAP

authentication for the MicroStrategy platform as well. LDAP users or groups can be linked to users or groups in the MicroStrategy environment and imported from the LDAP store into MicroStrategy which simplifies managing password resets and the creation of users.

Similarly, as user management tasks tend to be repetitive, manual, and time consuming, you can automate user management using tools such as Command Manager, MicroStrategy Web SDK, or System Manager. These tools enable you to manage multiple users simultaneously while reducing operational costs,. Some of the examples of user automation scenarios include:

- Creating automated schedules to import users and user groups from LDAP
- Using custom APIs to automatically import and synchronize users from a third-party provisioning system.
- Using bulk load operation involving user import from a comma-separated values (.csv) file.

Additionally, you can use the Find and Replace Privileges tool in Developer to update privileges of multiple users simultaneously. You can access the Find and Replace Privileges tool, in Developer by right-clicking User Manager and selecting Find and Replace Privileges.



 For detailed instructions on how to find and replace privileges, see the MicroStrategy Developer Help.

5 Synchronize MicroStrategy user management with HR systems

You should align the management of MicroStrategy users closely with the Human Resources (HR) systems and processes to ensure that the MicroStrategy platform accurately reflects the current users with the correct privileges.

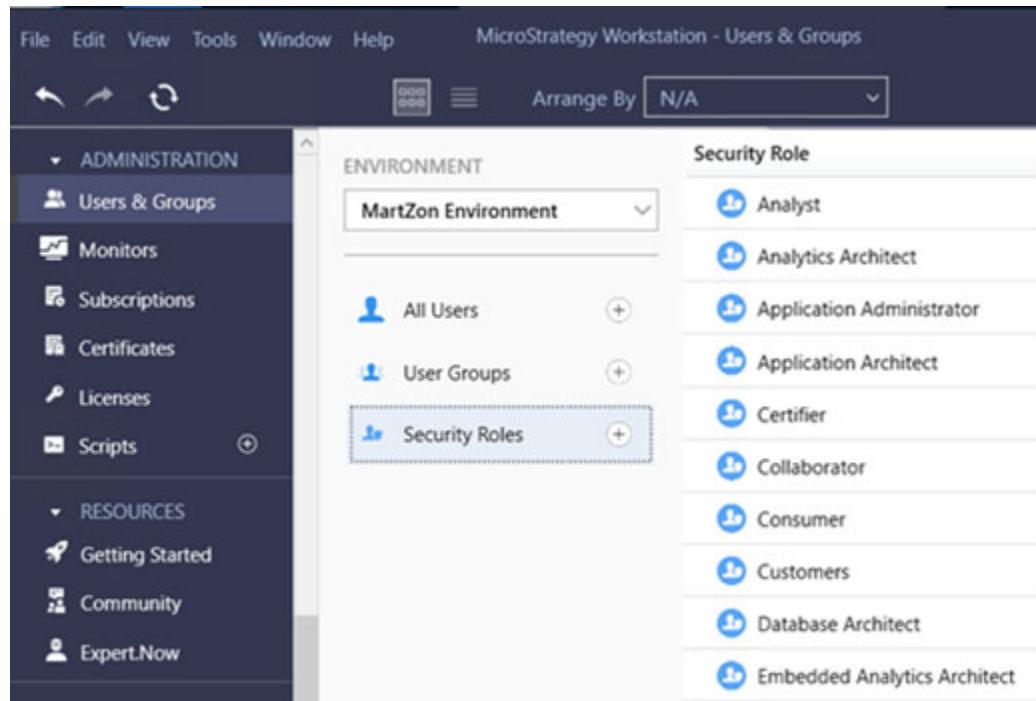
To understand how HR-related changes affect the MicroStrategy platform, the platform administration team should strive to learn both the HR processes and the intentions behind each process. This will help the team with the following:

- **Determining proper authorization**—Understanding how a person's attributes will be used to describe who they are and what access they need. It will also identify user attributes that are confidential, and should not be shared with others. For example, SSN or date of birth should not be part of the user login IDs.
- **User creation, modification, and deletion**—Determine how the administrator is notified if a person is hired, changes departments, or leaves the company.

6 Assigning privileges

Privileges can be assigned to users and user groups directly or through security roles. The difference is that the former grants functionality across all projects while the latter only apply within a specified project.

 You can access user privileges in MicroStrategy Workstation, by navigating to **Users and Groups**, and selecting **Security Filters**.

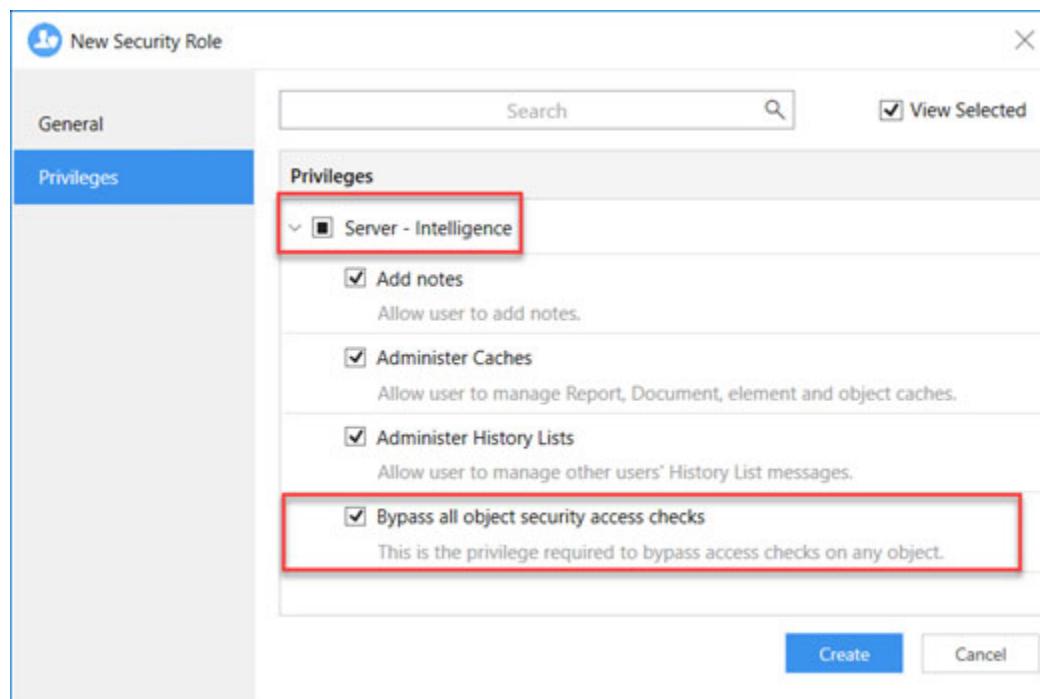


As users inherit privileges at the user, group, and security roles level, you should assign privileges to security roles only and then assign security roles to specific user groups (especially when rolling out a project to a larger user groups) to simplify and streamline user maintenance.

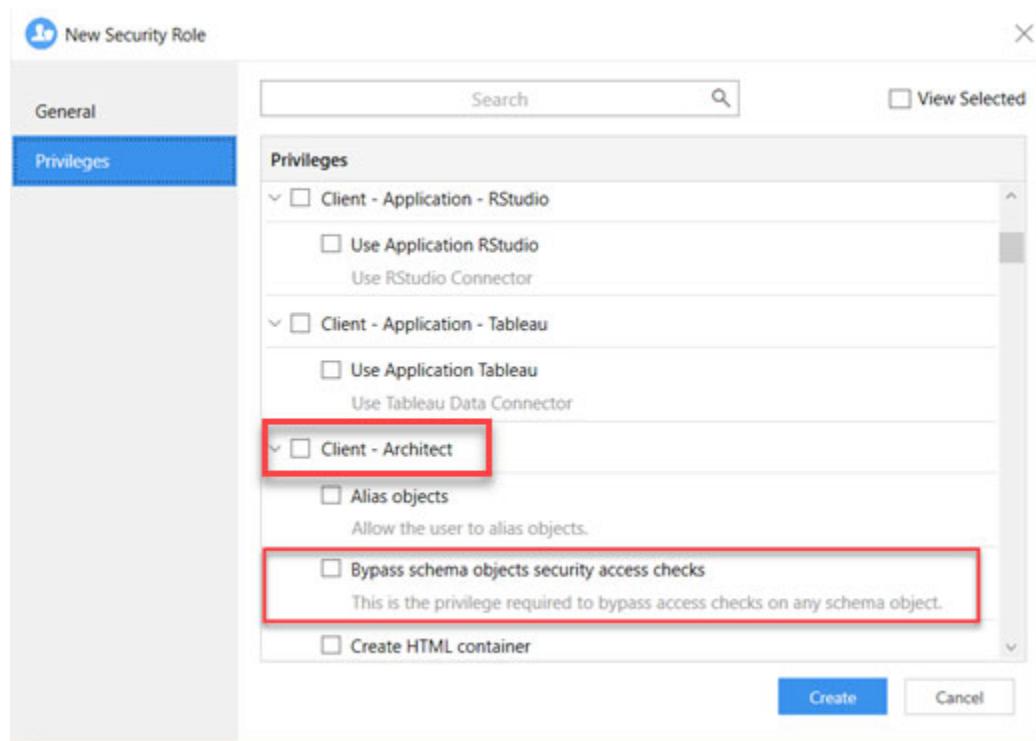
To provide a consistent and manageable security model, no privileges, permissions or data access should be configured for individual users.

Additionally, you should also be careful in assigning the following special privileges as they cause the normal access checks to be bypassed:

- **Bypass all object security access checks**—This Intelligence Server privilege allows the user to ignore the access checks for all objects.



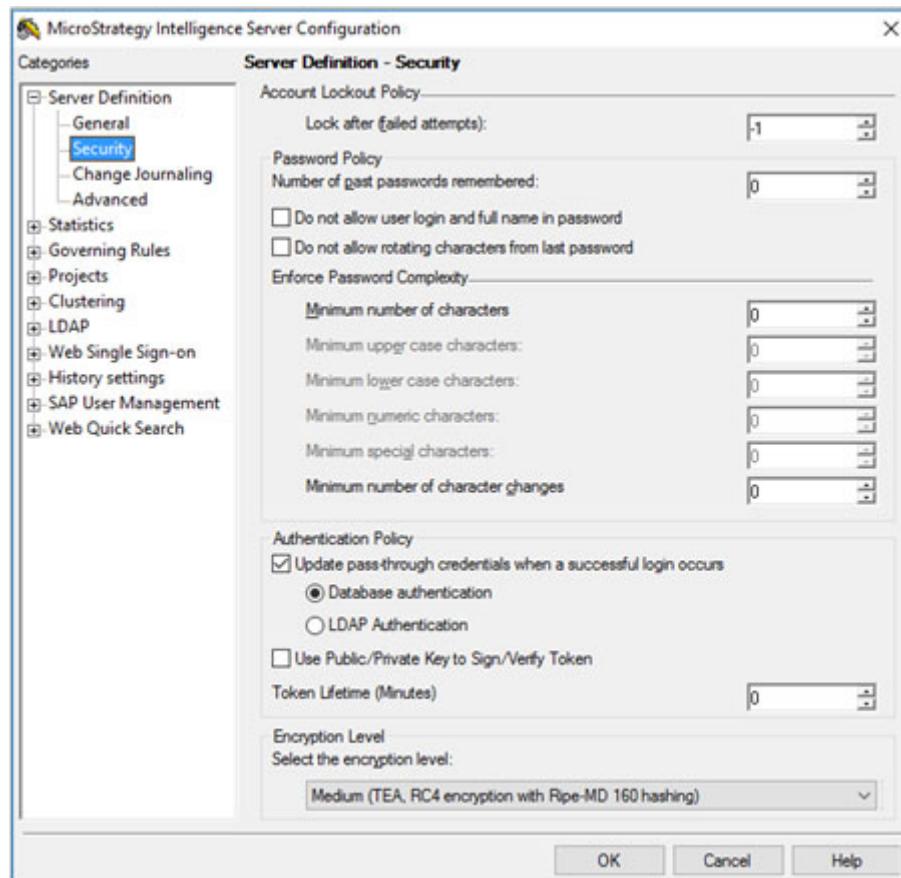
- **Bypass schema object security access checks**—This Client - Architect privilege allows the user to ignore the access checks for schema objects.



These two privileges should not be typically be assigned to any user other than the administrators as the users with these privileges are not restricted by access control permissions and are considered to have full control over all objects and schema objects, respectively.

- 7 **Assigning permissions, ACLs, and security filters**—Where feasible, ACLs, permissions, and security filter should be applied at the group level rather than user level to simplify an administrator’s tasks. In addition, when using the Denied All permission, assign it to a special group (or a special user, if necessary) so that, even if permission is granted at another level, the permission is still denied.
- 8 **Protect passwords**—Always maintain tight control of the administrator’s login information to ensure the security of your environment and provide accountability for any changes made as an administrator.

Additionally, configure the MicroStrategy application to enhance password security and account lockout options.



9 Using guest authentication—If using guest or anonymous authentication, modify privileges and permissions for guest users. Anonymous access is by default disabled at both the server and the project levels. Do not assign the Delete permission on any object to the guest user account.

In addition, ensure that guest users have access to the Log folder in the following location:

- **Windows**—C:\Program Files (x86)\Common Files\MicroStrategy
- **Linux/Unix**—<MSTR_LOG_PATH> which is the directory specified as the Log directory during MicroStrategy installation

This ensures that any application errors that occur while a guest user is logged in can be written to the log files.

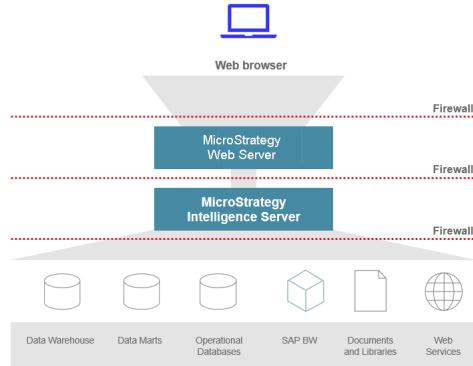
10 Secure externally-accessible MicroStrategy Web/Mobile

Servers—Configure effective Demilitarized Zones (DMZ) for security purposes. In production environments, most users typically access the

MicroStrategy platform by connecting to MicroStrategy Web using a browser or connecting to MicroStrategy Mobile Server using the MicroStrategy Mobile app.

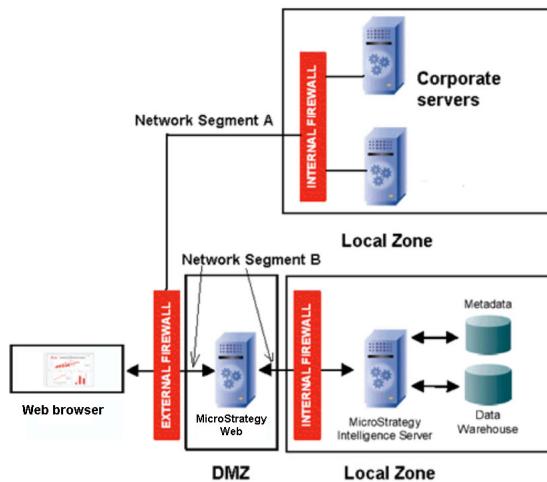
The core component of a MicroStrategy Web or MicroStrategy Mobile deployment is a secure MicroStrategy Web server or a MicroStrategy Mobile Server, respectively. This server facilitates communication between the browsers and Intelligence Server through a trusted connection.

The server coordinates data query requests with the Intelligence Server, by taking user requests, passing the requests to the Intelligence Server, and finally displaying the results received from Intelligence Server on the users' browsers or the MicroStrategy Mobile apps. Firewalls should be used for security purposes. The following image shows a four-tier Web environment using firewalls:



Enterprises typically install the MicroStrategy platform on more than one server to distribute the workload. Secure communication across these servers is often governed by layers of firewalls constructed into DMZ.

Using multiple firewalls, two distinct DMZs are created with one DMZ protecting the mobile and web servers. The second DMZ secures the infrastructure of the data sources and Intelligence Server.



An effective DMZ is characterized not only by the presence of firewalls, but also an architectural component that accesses the database, which resides behind a firewall. The Intelligence Server is the core of MicroStrategy's analytics platform, and is the only component that has access to the database. It resides between two firewalls in the same way that the MicroStrategy Web (or Mobile) server resides between two firewalls.

- 11 Create a User Manager administrator**—Depending on your user base, it is generally a good practice to create a User Manager administrative account to manage user accounts, including creating new users, granting privileges, and moving users from one group to another. The account is created in the User Administrators group and does not take up an Administrator license.

Allowing multiple people to have user management privileges reduces bottlenecks and ensures redundancy in case the dedicated platform administrator is out of office. It also increases transparency and traceability as users will be identified by their own unique login IDs, as opposed to the generic *administrator* login ID.



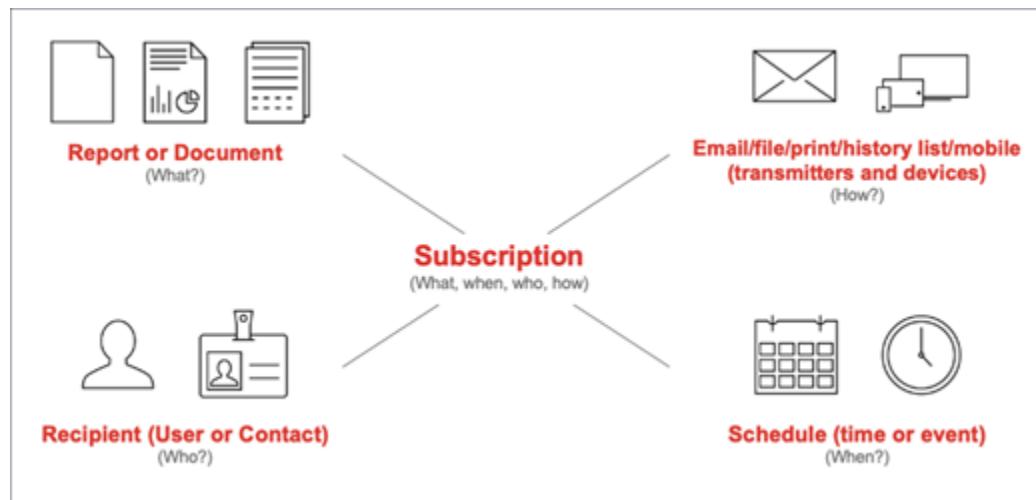
You can create a User Manager administrator, with limited administrative capabilities, by making the user a member of the User Administrators group. Additional configuration will be required if you want to provide this administrator additional capabilities such as creating a new security role.

- 12 Limit the number of connection maps**—If using connection mapping, you should be aware that even though, there is no hard limit on the number of connection mappings to be implemented in the MicroStrategy analytics

platform, having several mappings will impact system performance due to the fact that each mapping will open a new connection to the data warehouse from Intelligence Server. These connections impact both the Intelligence Server and the data warehouse being accessed, and the performance decreases proportionally to the number of mappings in place.

Configuring Distribution Services

MartZon has decided to use Distribution Services which enables the scheduling and delivery of reports, dossiers, and documents to subscribers' email accounts, file locations, and printers. The following image shows the components necessary to create a subscription:



You need to standardize the creation of the Distribution Services-related components, such as devices and transmitters, to enable your platform administration team to consistently create such objects. For example, you might develop guidelines related to the following:

- **Requirements gathering**—What data your platform administration team needs to gather for creating a subscription. For example, in order to create a subscription, the subscription creator should know the delivery format, dataset details (such as the name of the report), recipients, subscription schedule, and so forth.
- **Default delivery format**—Which default delivery format your platform administration team needs to use, if a user does not have any preference for it.
- **Data compression**—Whether your platform administration team needs to enable the zipping feature for a subscription, if a user does not have any preference for it.

- **Scheduling window**—What default scheduling time your platform administration team needs to use, if a user does not have any preference for it.

Best Practice

Distribution Services best practices

When configuring Distribution Services, your platform administration team should use the following best practices.

- 1 **Use faster delivery formats for subscriptions**—When configuring Distribution Services subscriptions, the delivery performance can vary based on your delivery format. When specifying delivery format for subscriptions, you should be aware that PDF, plain text, and CSV file formats generally offer the fastest delivery performance.

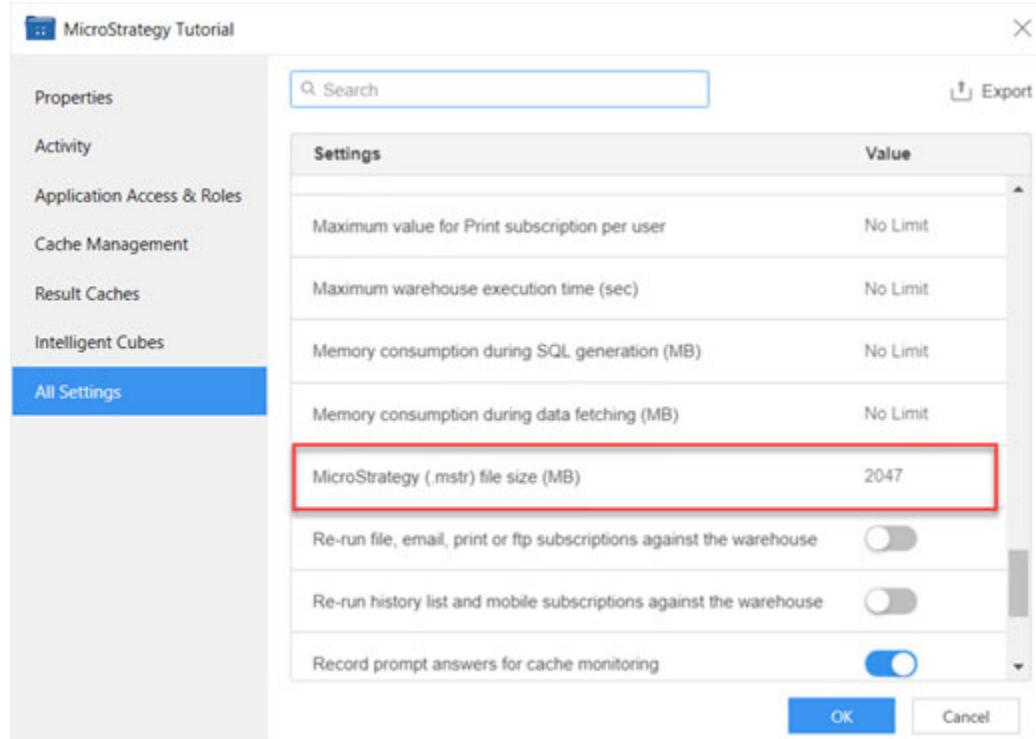
 Delivery performance also is dependent on other factors such as hardware, operating system, network connectivity, and so on. For example, the performance of the print delivery method depends on the speed of the printer.
- 2 **Manage delivery of large-sized reports, dossiers, or documents**—When sending large-sized reports, dossiers, or documents, you should:
 - Enable the zipping feature for the subscription so that files are smaller.
 - Use bulk export instead of the CSV file format.

 For details on bulk exporting, refer to the Advanced Reporting Guide product manual.
 - Schedule subscription deliveries to occur when your Intelligence Server is experiencing low traffic.
- 3 **Use prompts**—If your organization is processing a smaller number of subscriptions, such as 100 or fewer, better performance may be achieved by sending each subscription to the largest number of recipients possible. This can be achieved by designing reports, dossiers, or documents that answer business questions for the widest variety of analysts and by adding prompts to the report, dossier, or document.

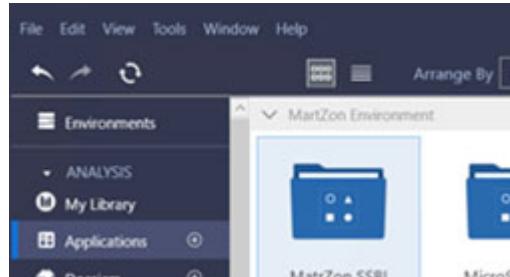
 For details on creating and adding prompts to a report, refer to the Basic Reporting Guide product manual.
- 4 **Use clustering for large number of subscriptions**—If your organization is processing many subscriptions, such as 1,000 or more, better performance can be achieved by using a cluster environment as subscription jobs are distributed across nodes. For performance optimization purposes, to avoid

overloading a single node in a clustered environment, it is recommended whenever possible to create subscriptions with similar number of recipients.

- 5 **Use bulk export**—If you are processing many subscriptions, consider using the bulk export feature.
- 6 **Assign address to contacts**—When creating contacts, make sure that each contact has at least one address for each delivery type. Otherwise the contact does not appear in the list of contacts for subscriptions that are for a delivery type that the contact has no address for. For example, if a contact does not have an email address, when an email subscription is being created, that contact does not appear in the list of contacts.
- 7 **Provide prompt answer**—When selecting reports to be subscribed to, make sure none of the reports have prompts that require an answer and have no default answer. If a report has a prompt that requires an answer but has no default answer, the subscription cannot run the report successfully, and the subscription is automatically removed from the system.
- 8 **Configure maximum file size**—The maximum file size of dossier (.mstr) files that can be sent through Distribution Services is defined by the MicroStrategy (.mstr) file size (MB) setting. You can set this governing rule at the application level using MicroStrategy Workstation.



To access governing rules for applications using Workstation, navigate to **Applications**, right-click your application, then select **Properties**. In the application properties windows, select the **All Settings** tab, then modify the value in the right pane and select **OK** to save your modification.



9 Monitor subscriptions usage—You should periodically monitor subscriptions usage in Enterprise Manager to ensure that you have optimized the configuration of Distribution Services. For example, you should analyze:

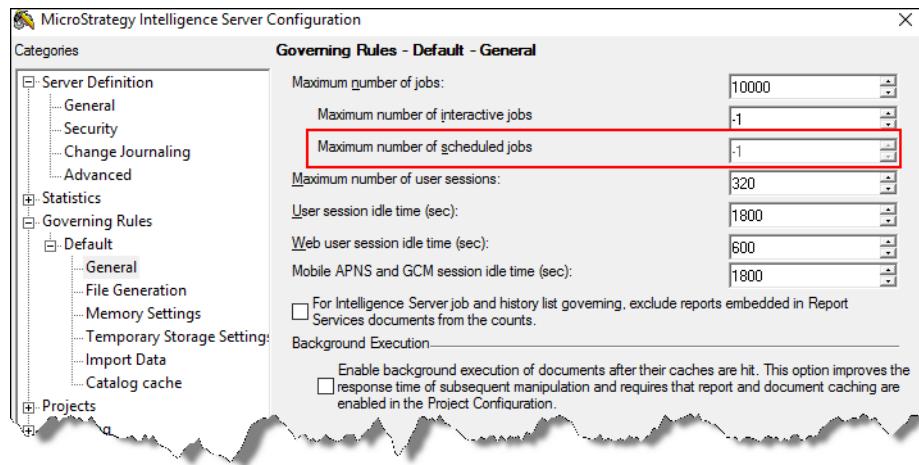
- Weekly subscription trends by delivery type
- Top subscribed reports, dossiers, documents, and contacts
- Longest executing subscriptions
- Unused Subscriptions in the past 6 months (or more)



Subscriptions that are no longer used should be deleted

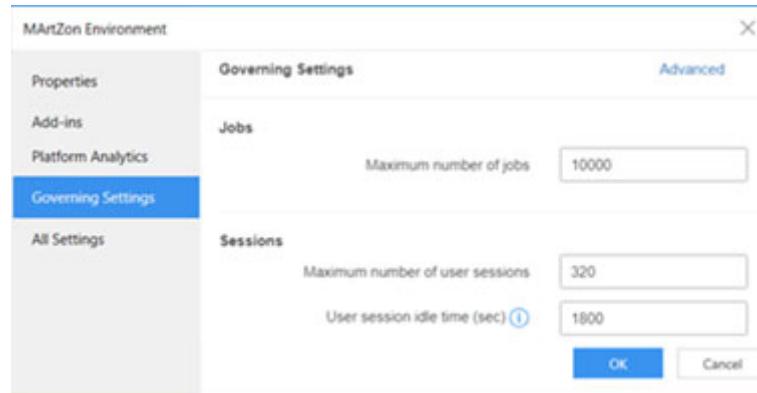
10 Distribute subscription load evenly—Schedule subscription concurrency execution to distribute the load evenly. To do so:

- **Configure the number of concurrent scheduled jobs**—The number of concurrent scheduled jobs is a governing setting that controls the amount of jobs that trigger at the same time on Intelligence Server based on a schedule. Setting this server-level governor correctly is an important step to optimize performance, because this step defines the concurrency rate in Intelligence Server.



 By default, the value for Maximum number of scheduled jobs is set to -1 which indicates no limit.

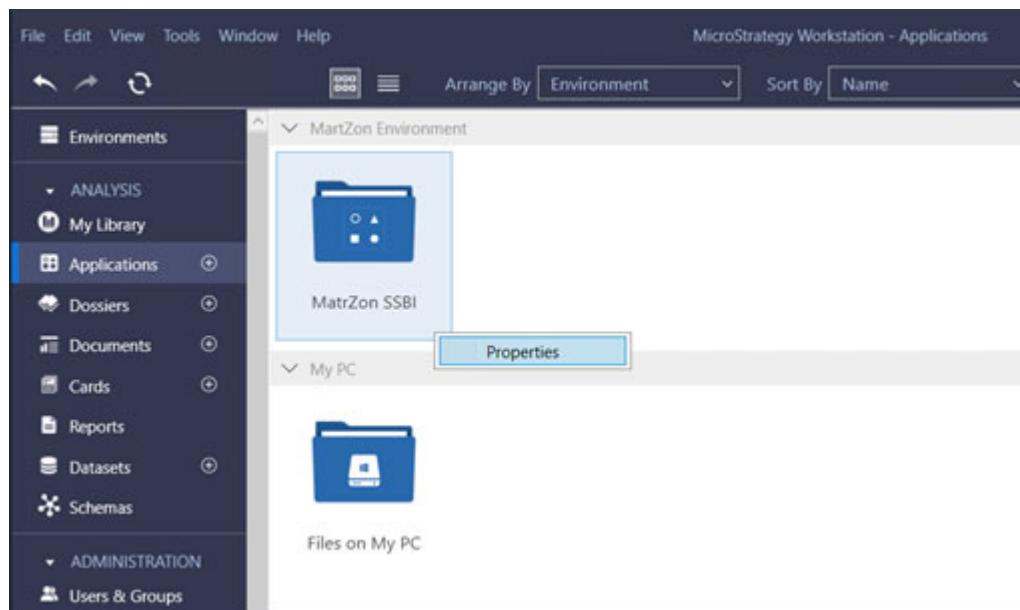
- **Configure the number of database instance threads**—The speed at which Intelligence Server processes jobs is directly related to the amount of warehouse threads and the concurrency that exists with other unscheduled processing jobs. For example, if a subscription generates 200 jobs and Intelligence Server has only 50 available threads, the remaining 150 jobs will have to wait in a queue.
- **Configure the maximum number of jobs**—Processing jobs consumes Intelligence Server memory. Consider the memory usage on Intelligence Server for processing jobs. For example, if the available memory on Intelligence Server only allows for processing of 100 jobs at a given time, it is recommended to define the Maximum number of jobs governor setting to 100 or slightly less. Because it minimizes the possibility of memory swapping to disk, this change will result in faster overall processing times.



 The Maximum number of jobs setting limits the number of concurrent jobs that may exist on this Intelligence Server. Concurrent jobs include

report, element, and auto-prompt requests that are executing or waiting to execute. Finished (open) jobs, cached jobs, or jobs that returned errors are not counted. A value of -1 indicates no limit. By default, this is set to 10000.

To access governing settings for your environment in Workstation, navigate to **Environments**, right-click your Environment, then select **Properties**. In the application properties windows, select the **Governing Settings** tab, then modify the value in the right pane and select **OK** to save your modification. You can select the **All Settings** tab to access all of your environment settings.



- **Schedule batch subscription jobs outside the peak time periods**—To optimize the performance of your environment, you should schedule batch subscription jobs outside the peak time periods. Alternatively, you can also stagger the execution if you need to run subscriptions during the peak time periods.

Exercise 4.3: Configure Distribution Services error handling

As the Platform Administrator, you want to send only relevant information to MartZon's users. You want to ensure that Distribution Services only delivers reports, documents, and Dossiers that return some data. You also want to limit the number of reports, documents, or dossiers a user can send to an email address to 30 at time. Finally, you wish to include a legal disclaimer for all the emails that the distributions services delivers to your users.

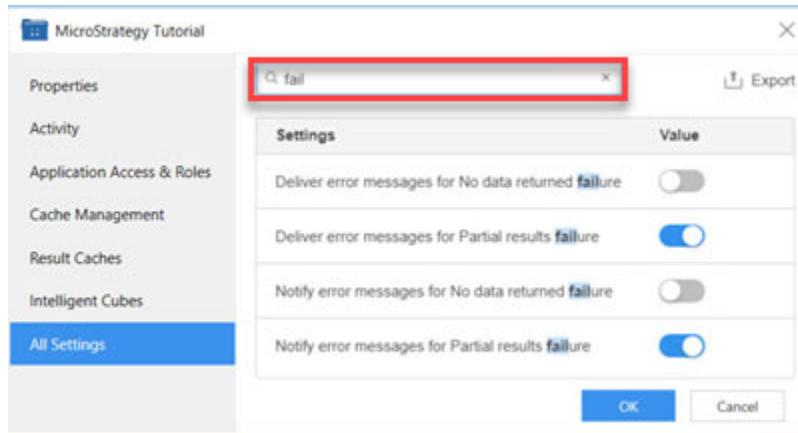
In this exercise, you use MicroStrategy Workstation to configure Distribution Services so that it does not send out any empty reports/documents in the MicroStrategy Tutorial project.

Configure Distribution Services

In order to prevent Distribution Services from sending emails containing empty reports/documents but send if they contain some data, change the Distribution Services error handling rule using Workstation.

- 1 On your remote Windows environment, launch **Workstation**.
- 2 From the navigation pane, select **Environments**, then right-click **MartZon Environment** and select **Update Application List**.
- 3 Select **MicroStrategy Tutorial**, then select **OK**.
- 4 From the navigation pane, select **Environments**, right-click the **MicroStrategy Tutorial** application, then select **properties**.

- 5 In the left pane, select **All Settings**, and in the top search box type **fail**.

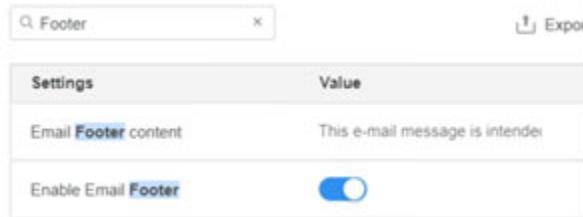


- 6 Disable the setting for **Deliver error messages for No data returned failure**.
7 Enable the setting for **Deliver error messages for Partial results failure**.
8 Disable the setting for **Notify error messages for No data returned failure**.
9 Enable the setting for **Notify error messages for Partial results failure**.



A report/document returns partial results when the size of the report/document exceeds the memory governing setting for Maximum memory consumption for PDF files and Excel files.

- 10 Type **email** in the search box, set the value for **Maximum value for Email subscriptions per user** to **30**.
11 To include a disclaimer at the end of emails sent by the distribution services, type **Footer** in the search box.



- 12 Enable the setting for **Enable Email Footer**.
13 Copy and paste the following message as the value for **Email Footer content**.

This e-mail message is intended only for MartZon users and may contain information that is privileged, confidential and exempt from disclosure under applicable law.

14 Click OK.

From now onwards, if a report/document returns some data, Distribution Services will deliver it but no delivery will be made if a report/document returns no data.

Exercise 4.4: Configure Distribution Services components

In the next set of exercises, you will automate subscriptions execution. You will start by configuring the Distribution Services components.

When using Distribution Services, you need to configure appropriate transmitters for delivering subscription content to users who are subscribing to report and document deliveries.

You also need to configure appropriate devices. A device defines the formatting and other properties for the specific type of delivery, for example, the specifics needed to format and send a file via email delivery as compared to the specifics needed to send a file to a network location. You can use an existing device and change its settings, or configure a new device for the formatting and transmission properties for the subscription content.

In this exercise, you:

- 1** Configure the email transmitter
- 2** Create an email device
- 3** Create a user with an address based on the device that you created. The user requirements are:
 - The user name is Tina Smith.
 - Tina is a member of the MicroStrategy Web Professional group.
 - She has all privileges under Deliveries.
 - She has the following two delivery addresses:
 - An email address that uses the Email for End-Users device
 - The physical address is your email address
- 4** Create a contact with an address based on the device that you created. The End-User A contact has two addresses:
 - Email address, Email A, that uses the Email for End-Users device
 - The physical address is your email address
- 5** Create a Contact Group, called External Clients. Add End-User A to this group, and link this group to the MSTR End-User user.

In the subsequent exercises, you will

- *Create a cache update subscription in Developer*
 - *Trigger “Database Load” event using Command Manager and automate its execution via Windows Task Scheduler*
-

Configure the email transmitter

- 1 Launch Developer and log into the **MicroStrategy on AWS I-Server** project.
 - 2 From the Folder List pane, expand **Administration**, then expand **Delivery Managers**.
 - 3 Select **Transmitters**, then double-click **Email** in the right pane.
 - 4 Select **Do not appear in To or CC (use Bcc delivery)** under Recipients.
 - 5 Select the **Message Output** tab, then select **Send messages to recipients via SMTP**.
 - 6 Clear **Save to file**, then click **OK** to save your changes.
-

Create an email device

- 1 From the Folder List pane, expand **Administrator**, then expand **Delivery Managers**.
 - 2 Right-click **Devices**, point to **New**, and select **Device**.
 - 3 Select **Email**, then click **OK**.
 - 4 In the **Name** box, type **Employees**.
 - 5 Click **OK**.
-

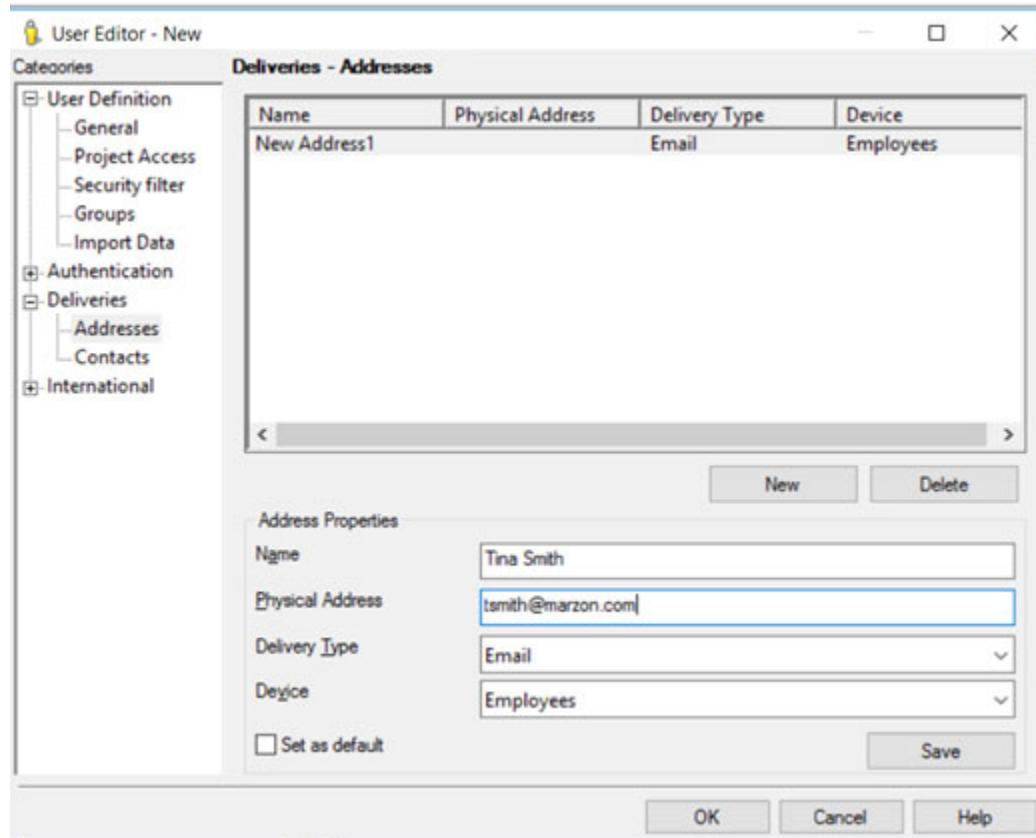
Create a MicroStrategy user

- 1 In the Folder List pane, expand **Administration**, then expand **User Manager**.

- 2** Expand **MicroStrategy Groups**, **MicroStrategy Web Reporter** and **MicroStrategy Web Analyst** user groups.
- 3** Right-click **MicroStrategy Web Professional** point to **New**, then select **User**.
- 4** In the User Editor, in the General category, specify the following information for the user:
 - **Developer login:** tsmith
 - **Full name:** Tina Smith
 - **Password:** hello1!
 - **Confirm Password:** hello1!
- 5** Select the **User cannot change password** check box.
- 6** In the Categories pane, expand **User Definition**, then select **Project Access**.
- 7** Select **Server - Distribution** to grant this privilege to your user.
- 8** In the Categories pane, expand **Deliveries**, and select **Addresses**.
- 9** Click the **New** button, then configure an email address:
 - a In the **Name** box, type **Tina Smith**.
 - b In the **Physical Address** box, type **tsmith@martzon.com**.
 - c In the **Delivery Type** drop-down list, leave the default option of **Email**.
 - d In the **Device** drop-down list, select the **Employee** device that you created in the previous set of steps.

10 Click **Save**.

The following image shows the User Editor with the address created:



Users with the appropriate privileges can create their own addresses in MicroStrategy Web.

11 In the User Editor, click **OK** to save the user.

Create a Contact

- 1 From the Folder List pane, expand **Administration**, then expand **Delivery Manager**.
- 2 Right-click **Contacts**, point to **New**, and select **Contact**.
- 3 In the Contact Editor, click the **General** tab.
- 4 In the **Name** box, type **James Hill**.

- 5 Click the **Addresses** tab.
- 6 On the Addresses tab, click **New** to define an address for the contact.
- 7 Under Address Properties, in the **Name** box, type **James Hill - Email**.
- 8 In the Physical Address box, type **jhill@acme.com**.
- 9 In the **Delivery Type** drop-down list, select **Email**.
- 10 In the **Device** drop-down list, select **Employees**.
- 11 Click **Save**.
- 12 In the MicroStrategy Developer window, click **Yes**.

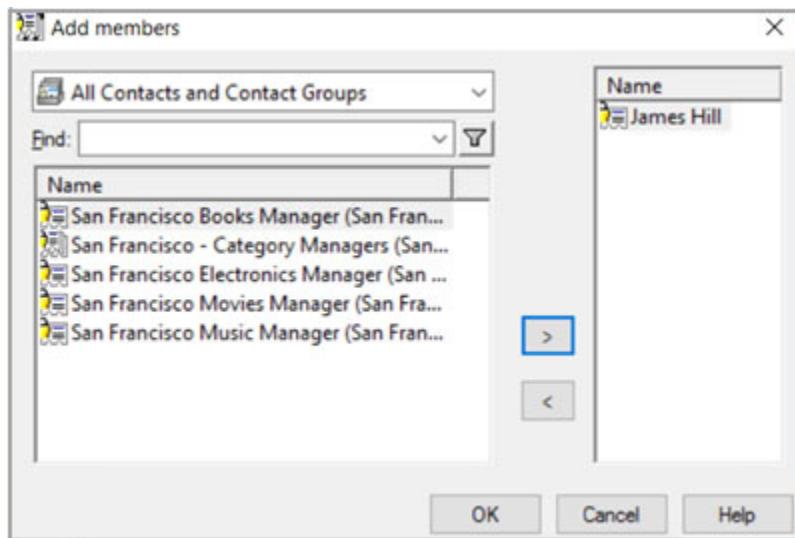
James is not yet linked to any users. You will link him to Tina via a contact group association later.
- 13 In the Contact Editor, click **OK** to save the contact.

Create a Contact Group

Create a Contact Group and add James to the group.

- 1 In the Folder List, right-click **Contacts**, point to **New**, and select **Contact Group**.
- 2 In the Contact Group Editor, click the **General** tab.
- 3 In the **Name** box, type **External Clients**.
- 4 Click the **Members** tab.
- 5 Click **Add** to add contacts to the group.
- 6 In the Add Members window, in the **Name** list, select **James Hill**.

- 7 Click the > button to add a contact to the contact group, then click **OK**.



- 8 Click the **Security** tab. Click **Browse**.
- 9 In the Select User to Link window, in the top drop-down list, double-click the **MicroStrategy Web Professional** group.
- 10 In the **Name** list, select **Tina Smith (tsmith)**.



If the name is not displayed, select the **Show users** check box at the bottom of the window.

- 11 Click **OK**.

- 12 In the Contact Group Editor, click **OK** to save the group.

You can see the contacts that belong to this group, by expanding **External Clients** in the Object Viewer.

Name	Status
Administrator	Enabled
External Clients	Enabled
James Hill	Enabled
James Hill	Enabled
Jen Thompson	Enabled
Mark Anderson	Enabled
MicroStrategy Web User	Enabled
MSTR User	Enabled
Tina Smith	Enabled

Exercise 4.5: Configure a cache update subscription

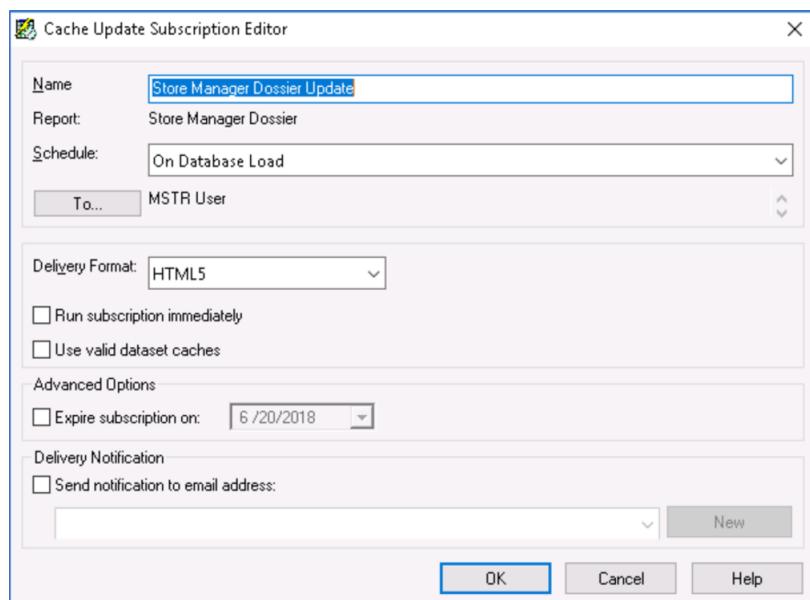
In this exercise, you will create a cache update subscription for the Store Manager Dossier which is located in Public Objects\Reports\Sample Dossiers folder in the MicroStrategy Tutorial project. The cache should be updated based on the Database Load event. You will automate the triggering of this event in a subsequent exercise.

Next, using MicroStrategy Web, you will create an email subscription to the Memphis Detail Performance Analysis document located in the Public Objects\Reports\Subject Areas\Daily Analysis folder.

 The Platform Administrator should work with other Intelligence Center architects in identifying report, documents, and dossiers, along with the schedules for creating cache update subscriptions.

Access the Cache Update Subscription Editor

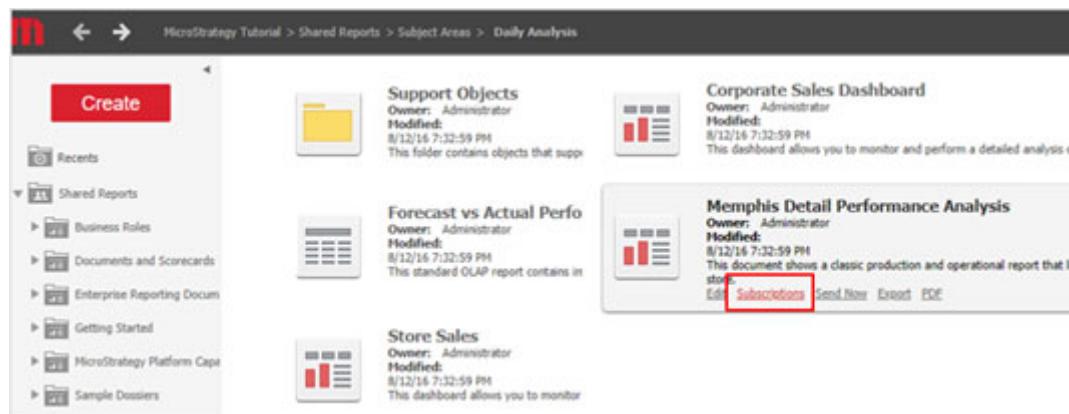
- 1 In Developer, in the MicroStrategy Tutorial project, access the **Public Objects\Reports\Sample Dossiers** folder.
- 2 Right-click the **Store Manager Dossier** and hover over **Schedule Delivery To**, then click **Update Cache**. The Cache Update Subscription Editor displays.
- 3 Click the **Schedule** drop-down, and select **On Database Load**.



- 4 By default, you are the recipient for the subscription. To add additional recipients or to remove yourself from the subscription, click **To**. The Recipients Browser window opens. Select the users and groups that you want to receive the subscription and click **OK**. For this exercise, leave **mstr** as the recipient.
- 5 To execute the report or document immediately when the subscription is saved, select **Run subscription immediately**.
- 6 Click **OK**. Your subscription executes immediately.

Create an email subscription in MicroStrategy Web

- 1 Access MicroStrategy Web, and log in to the **MicroStrategy Tutorial**.
- 2 In the left pane, expand **Shared Reports**, then navigate to **Subject Areas\Daily Analysis** folder.
- 3 Hover over **Memphis Detail Performance Analysis** document, then click **Subscriptions**.



4 Click Add Email Subscription.

Mobile

Subscription Name

You do not have any mobile subscriptions.

Add mobile subscription

Email

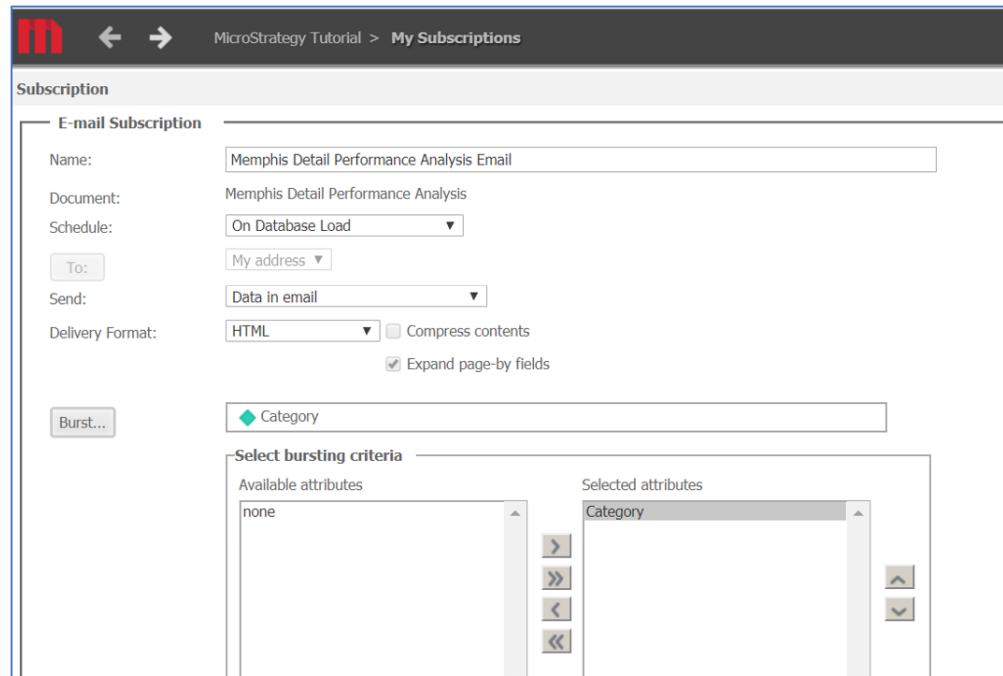
Subscription Name

You do not have any email subscriptions.

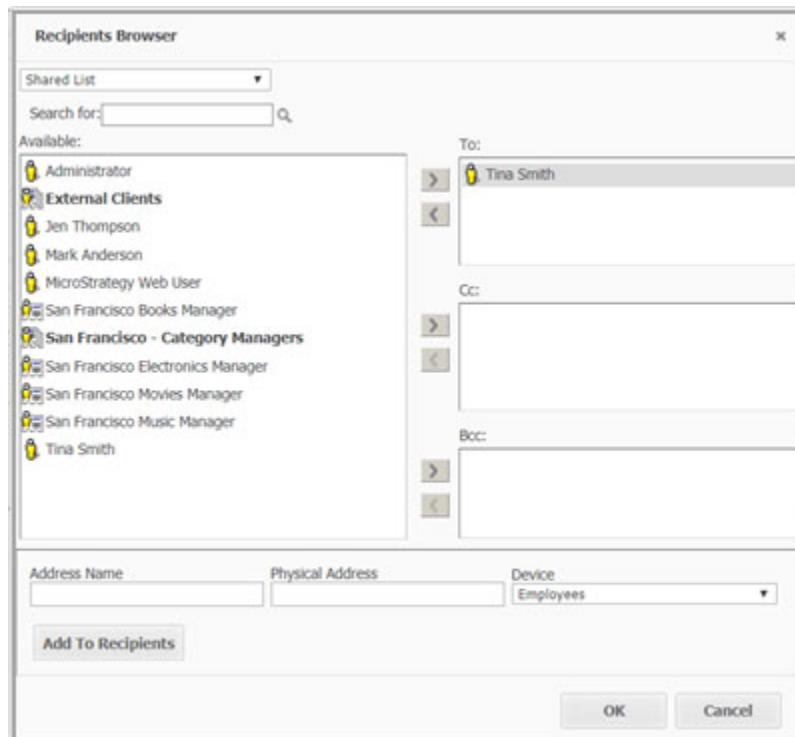
Add email subscription

File

5 In the Subscription window, select the **On Database Load schedule.**



- 6 Click **To**. In the Recipients Browser window, add **Tina Smith** as the recipient and click **OK**.



- 7 In the Subscription window, click **OK**.

A message is displayed, confirming that the subscription was created successfully.

Exercise 4.6: Automate subscription execution

In this exercise, you create a Command Manager script to trigger the Database Load event and then schedule the execution of the script via Windows Task Scheduler.

You first create a subfolder named Scripts under C:\ on the Windows machine of your cloud environment. Next, you create a Windows batch file named Trigger_MSTR_Event.bat which contains the Command Manager script to trigger the Database Load event.

Create a Command Manager script

- 1 From the **Start** menu, navigate to **MicroStrategy Products** and select **Command Manager**.
- 2 Log in to the **MicroStrategy on AWS I-Server** project source as the **mstr** user.
- 3 In Command Manager, create the following script:
TRIGGER EVENT "Database Load";
- 4 Save the script as **TriggerEvent** in **C:\Scripts**.
- 5 Exit Command Manager.

Create a folder and batch file on the Windows machine

- 1 On the Windows machine, create a directory on the C:\ drive named **Scripts**.
- 2 Open **Notepad** and enter the following code:

```
C:\\"Program Files (x86)"\MicroStrategy\\"Command  
Manager"\CMDMGR.exe -n "MicroStrategy on AWS I-Server"  
-u mstr -p "<password>" -f C:\Scripts\TriggerEvent.scp  
-o C:\Scripts\TriggerEvent_out.txt
```

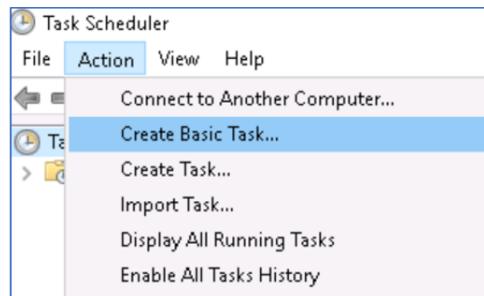
In the preceding syntax, replace **<password>** with your password from the MicroStrategy Cloud email.

- 3 Select **Save as** from the **File** menu in **Notepad**.
- 4 In the Save as Type drop-down, select **All Files**.

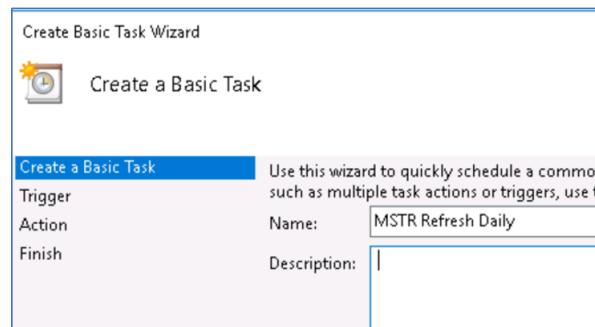
- 5 In the File name box enter **Trigger_MSTR_Event.bat**.
- 6 Save your file in the **C:\Scripts** folder, and close Notepad.

Schedule the task to run daily

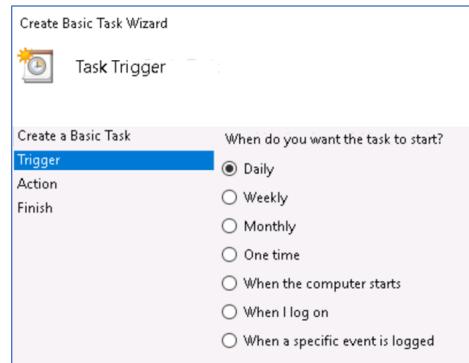
- 1 Click the Windows **Start** button, and then click **Windows Administrative Tools**. Next, double-click **Task Scheduler**.
- 2 Under **Action**, select **Create Basic Task**.



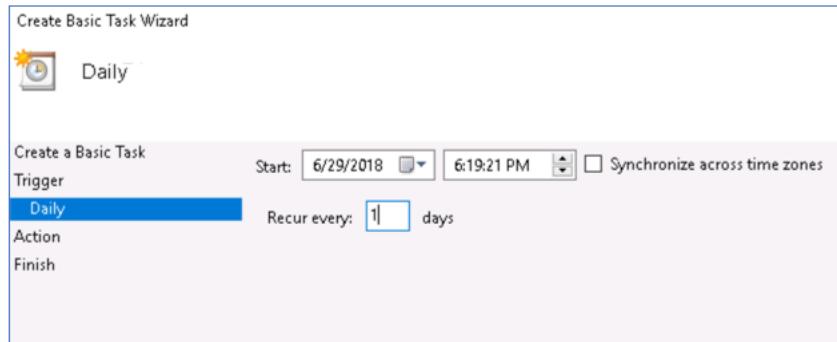
- 3 Name your task as **MSTR Refresh Daily**. Click **Next**.



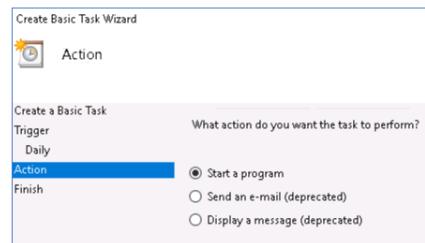
4 In the Trigger section, set it to run daily. Click **Next**.



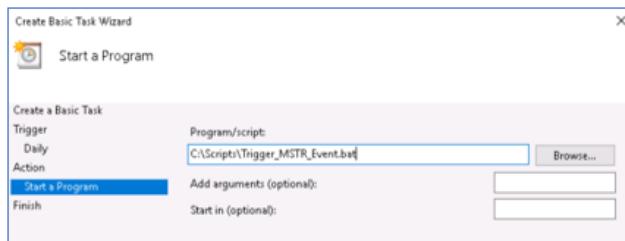
Then specify a time that's 3-4 minutes after the current time for testing purposes. Click **Next**.



5 In the Action section, select **Start a program**.



Then, in the Program/script box, browse to **C:\Scripts\Trigger_MSTR_Event.bat**.



- 6 Click **Next** and then click **Finish**.

Verify results

- 7 After your specified time, on the Windows machine, check the Command Manager output file **C:\Scripts\TriggerEvent_out.txt**. The content should look similar to the following indicating that the script has been executed:

```
6/21/18 3:44:01 PM UTC Version 10.11.0 (Build 10.11.0051.0056)
6/21/18 3:44:01 PM UTC Connected:mstr@MicroStrategy on AWS I-Server
6/21/18 3:44:01 PM UTC Executing task(s)...
6/21/18 3:44:01 PM UTC Checking syntax...
6/21/18 3:44:01 PM UTC Syntax is correct.
6/21/18 3:44:01 PM UTC Syntax checking has been completed.
6/21/18 3:44:01 PM UTC Event 'Database Load' has been triggered successfully.
6/21/18 3:44:01 PM UTC No results returned.
6/21/18 3:44:01 PM UTC Task(s) execution completed successfully.
6/21/18 3:44:01 PM UTC Execution Time: 00:00:00
6/21/18 3:44:01 PM UTC Successfully disconnected. (MSTR) MicroStrategy on AWS I-Server: mstr
#####
```

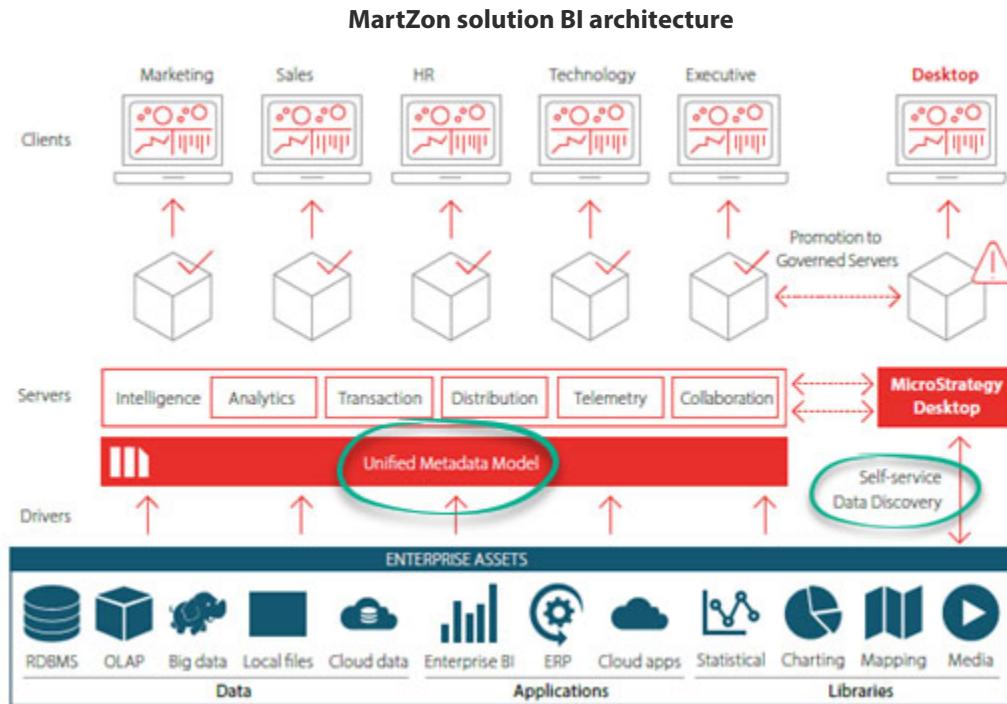
Streamlining your analytics deployment

As the Platform Administrator, you can streamline the implementation of analytics and mobility applications from small teams to departmental, divisional, and enterprise-wide deployments. You want to provide both IT professionals and business analysts at MartZon with access to data they need for building and deploying MicroStrategy analytics projects.

In traditional IT-centric deployments, datasets including reports and OLAP cubes are built based on a centralized semantic layer. In this layer, metadata schema objects such as attributes, hierarchies, and facts map corporate data to the business model. These datasets are typically created by Analytics Architects, and their availability, integrity, and security are managed by Platform Administrators. Security filters, connection mapping, and object permissions (ACLs) are all part of the MicroStrategy security model used to control access to data in an application.

In addition to sanctioned corporate data, MartZon business users require access to personal data from disparate and external data sources. They need to import data from data sources such as an Excel file, a Freeform script, or a Salesforce.com report with minimum project design requirements. Once imported, your business users, analysts, and data scientists can share and promote their datasets back into the certified data model.

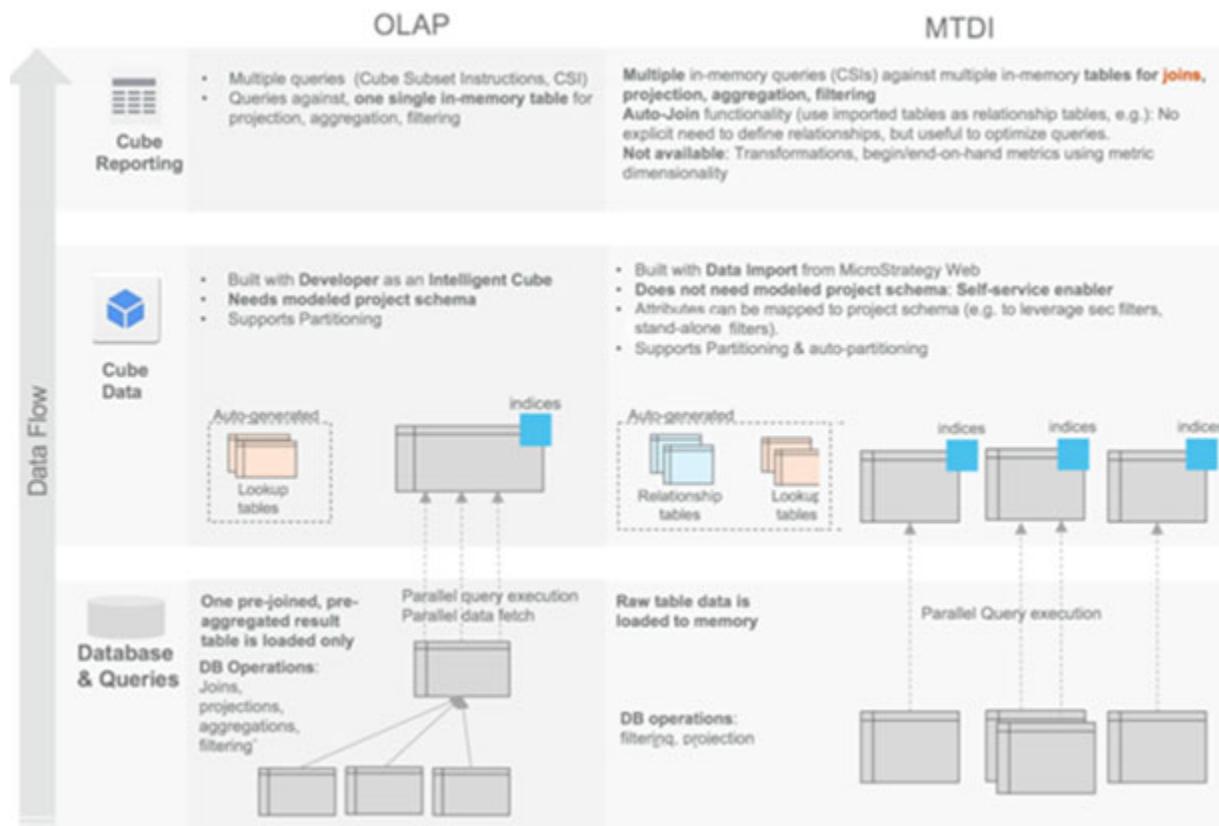
To meet these requirements, you need to employ two BI styles at MartZon. Your delivery approach to BI and analytics should reflect a bimodal management model that combines business agility and IT governance. It is important to allow your business users and analysts to import their data while maintaining control through certifying and securing access to relevant data.



Managing access to imported data cubes

In MicroStrategy, data from multiple sources can be combined and imported into a single Multi-Table Data Import (MTDI) cube. Different types of in-memory cubes include OLAP cubes (traditional Intelligent Cubes) and MTDI cubes. The primary difference between OLAP and MTDI cubes is how they are authored and created. OLAP cubes are authored using MicroStrategy Developer and are tightly coupled to a MicroStrategy project schema. Because of this, OLAP cubes are traditionally built and maintained by advanced BI developers or architects and represent governed data.

Conversely, MTDI cubes support decentralized departmental needs. They are typically authored by business users through MicroStrategy clients such as Web or Workstation. These cubes do not need a project schema. Therefore, business users can create these cubes by directly loading data from one or more sources and determining which columns are attributes, along with their relationships, and which are metrics. MTDI cubes are the self-service in-memory solution that provide agility in distributing data used in analytics.



You can use MicroStrategy Workstation to establish row-level data security by applying filters for datasets created using data import. You can also manage column-level data security, controlling permissions users and groups have for attributes and metrics in a dataset. Business users granted the requisite privilege can certify dossiers, documents, datasets, and cubes within Workstation. Working with your Intelligence Director, you should establish procedures for items to be reviewed by trusted members of your organization before they are considered official sources of content.

Exercise 4.7: User provisioning

In this exercise you use Workstation to provision your business users by creating a user group with two users and managing their security roles and permissions in the Tutorial application. You then deploy content from the Tutorial application to their MicroStrategy Library.

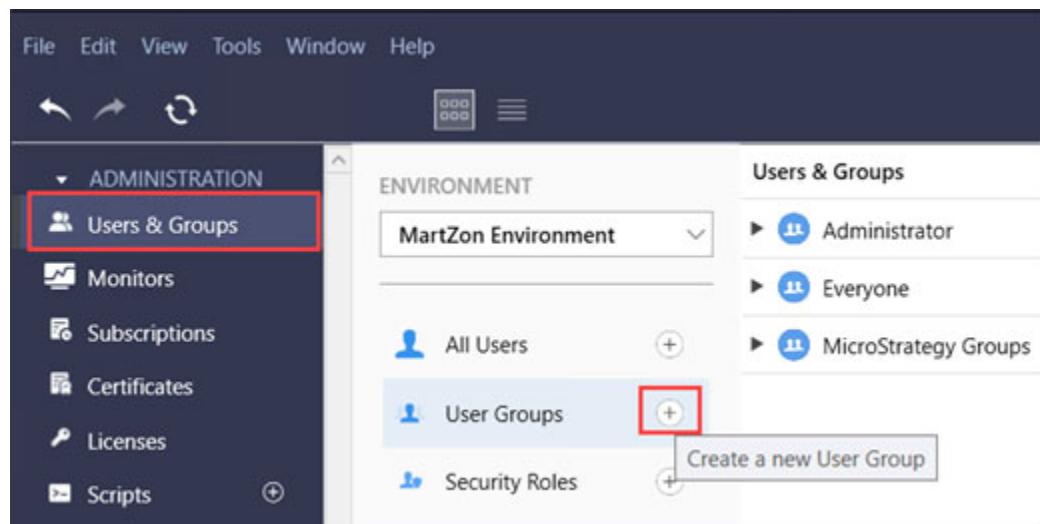
Your user provisioning requirements are as follows:

- User group:
 - Name: Sales
 - Security role: Analyst in MicroStrategy Tutorial application
 - Dossier deployed for all users in this group:
 - Sales Analysis
 - Office Royale Sales
 - Retail Sales Report
- User:
 - Name: Mark Anderson
 - Group: Sales
 - Data security requirement: Sales Data cube, subcategory filter
 - Boots
 - Dress Shoes
 - Heels
 - Additional security role: Certifier in MicroStrategy Tutorial application
- User:
 - Name: Jen Thompson
 - Group: Sales
 - Data security requirement: Sales Data cube, subcategory filter
 - Sandals
 - Slippers

- Sneakers
- Additional object security: Denied access for Cost metric in Sales Data cube
- Additional Library deployments: Documents
 - Company Sales
 - Corporate Sales Dashboard

Provision users and groups

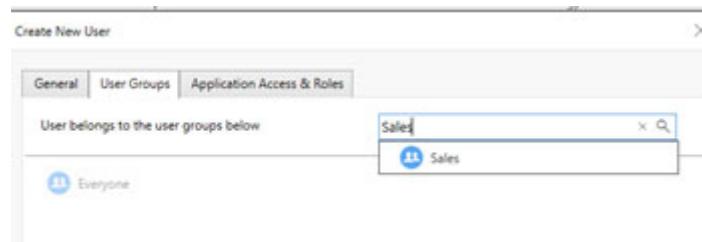
- 1 Launch **MicroStrategy Workstation** if not already open.
- 2 From the navigation pane, select **Environment**, and make sure you are connected to your environment.
- 3 From the navigation pane, click **Users & Groups**, then click the Add icon next to **User Groups**.



- 4 Type **Sales** for **User Group Name**.
- 5 Select **Application Access & Roles** from the left pane.
- 6 From the **MicroStrategy Tutorial** drop-down, select the **Analyst** role and click **OK**.

You are granting Analyst role privileges to the Sales group within the MicroStrategy Tutorial application. Any user added to the Sales group inherits the Analyst role privileges.

- 7 Click **Create Groups** to create your user group.
- 8 From the left pane, click the Add icon next to **All Users**.
- 9 Enter the following information for the first user:
 - Full Name: **Mark Anderson**
 - Email Address: **ma@martzon.com**
 - Login ID: **ma**
 - Password: **ma**
 - Confirm Password: **ma**
- 10 From the left pane, select **User Groups** and in the search box enter **Sales**.
- 11 Select **Sales** to add Mark to the Sales group.



- 12 Click **Create User**.

Create a second user

- 1 From the left pane, click the Add icon next to **All Users**.

2 Enter the following information for the second user:

- Full Name: **Jen Thompson**
- Email Address: **jt@martzon.com**
- Login ID: **jt**
- Password: **jt**
- Confirm Password: **jt**
- User Group: **Sales**

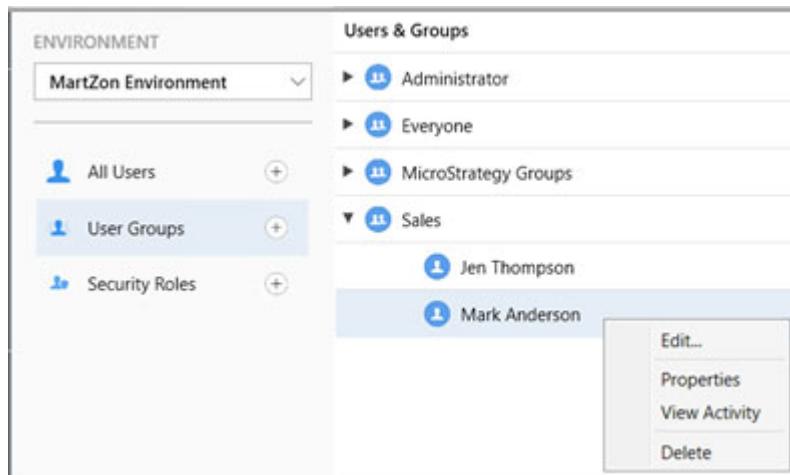
3 From the left pane, select **User Groups** and in the search box enter **Sales**.

4 Select **Sales** to add Jen to the Sales group.

5 Click **Create User**.

6 From the left pane, click **Users Groups**, then expand the Sales group.

7 Right-click **Mark Anderson** and select **Edit**.



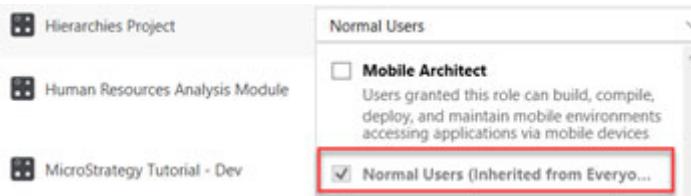
8 From the left pane, select **User Groups**.

What groups does Mark belong to?

- **Sales**: Manually assigned in the previous steps.
- **Everyone**: Automatically assigned. All users, except for the guest users, are automatically members of the Everyone group. This makes it easy for you to assign privileges, security role memberships, and permissions to all users.

9 From the left pane, select **Application Access & Roles** to view the security roles assigned to Mark within each project.

10 Click the Roles drop-down next to **Hierarchies Project**, then scroll down to view the Normal Users role.



Security roles are collections of privileges that are assigned to users and groups to suit their functionality needs for each application. The Normal Users security role has no granted privileges and is automatically assigned to the Everyone group. Since Mark belongs to the Everyone group, he inherits this role.

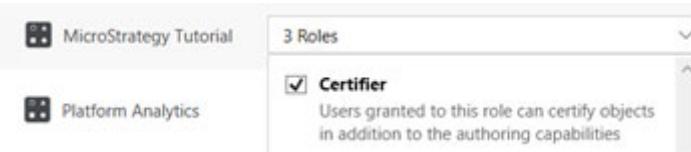
11 Click **OK** to close the Roles drop-down for the Hierarchies Project.

12 Click the Roles drop-down next to **MicroStrategy Tutorial**.

As you can see, the Analyst role privileges have been granted to Mark via his membership in the Sales group.

Mark needs to certify content such as dossiers, documents, and datasets for his team. As he is the only one authorized in his team to certify content, you assign this privilege at the user level.

13 Select the **Certifier** role, then click **OK**.



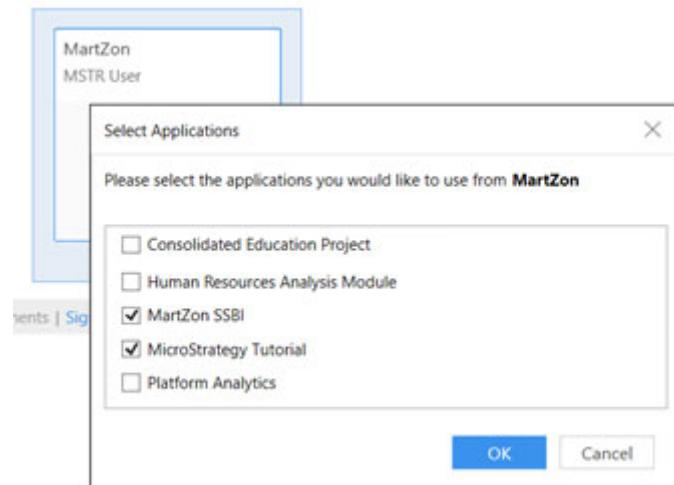
14 Click **Save** to save your user configuration changes.

Do not close Workstation; you will use it in your next exercise.

As part of on-boarding the users in the sales group, you now need to deploy several dossiers and documents from the Tutorial application to their personal portal in MicroStrategy Library.

Deploy dossiers and documents to user Library

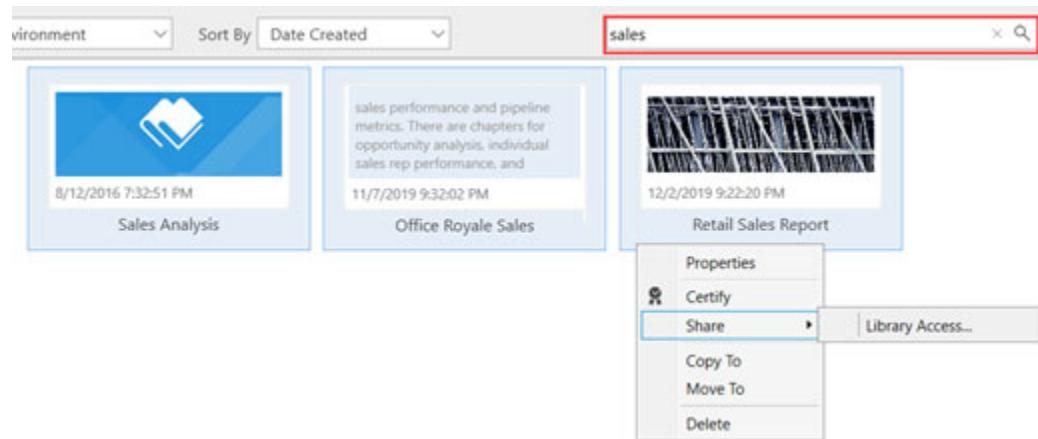
- 1 From the Navigation pane, select **Environments**, then right-click **MartZon Environment** and select **Update Application List**.
- 2 Select **MicroStrategy Tutorial**, then select **OK**.



Including the MicroStrategy Tutorial application in your environment provides you access to objects included in this project, such as documents and dossiers you want to deploy to your users.

- 3 From the Navigation pane, select **Dossier** to view the list of dossiers in your applications.
- 4 Type **sales** in the Search bar then press **Enter**.
- 5 Press and hold the **Ctrl** key, then select the following dossiers:
 - **Sales Analysis**
 - **Office Royale Sales**
 - **Retail Sales Report**

-
- 6 Right-click one of the selected dossiers and select **Share**, then select **Library Access**.



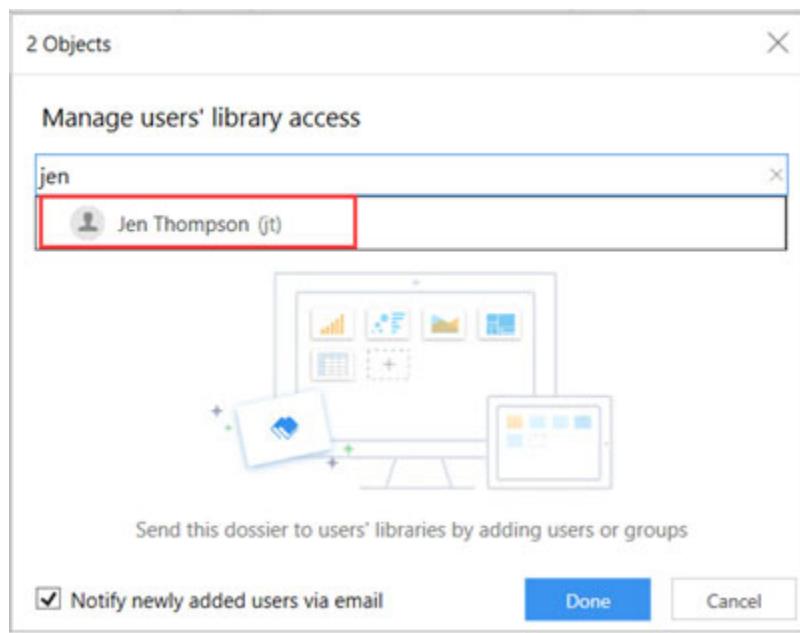
- 7 Type **Sales** in the Search box and select the **Sales** group.
- 8 Click **Done** to make the selected dossiers available in the personal library of all the users in the Sales group.

Make documents available in MicroStrategy Library

In addition to the three dossiers you have already deployed, Jen needs access to a couple of documents in her Library.

- 1 From the navigation pane, select **Documents**.
- 2 In the top search bar type **sales** and press **Enter**.
- 3 Press and hold the **Ctrl** key, then select the following documents:
 - **Company Sales**
 - **Corporate Sales Dashboard**
- 4 Right-click one of the selected documents and select **Share**, then select **Library Access**.

- 5 In the search box type **jen** and click **Jen Thompson**, then click **Done**.

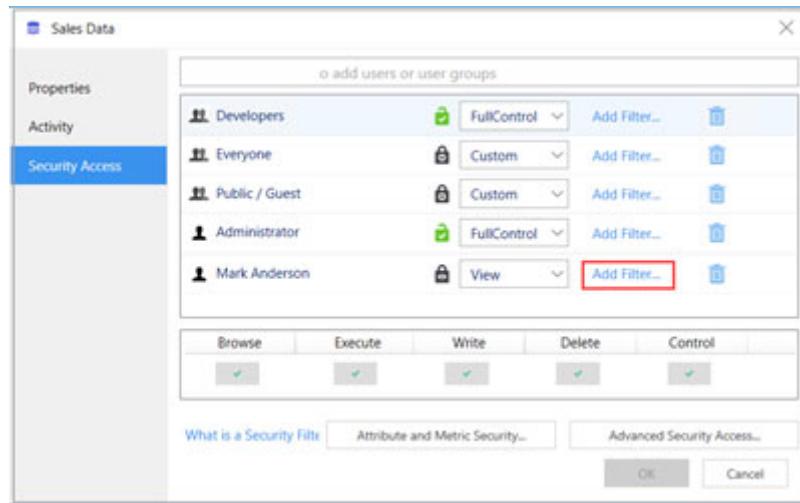


Exercise 4.8: Establishing data security

In this exercise you establish row and column-level data security for your sales analysts by applying security filter and object permissions for the MTDI Sales Data cube created using data import.

Apply security filter based on data access requirements

- 1 From the navigation pane, select **Datasets**, right-click **Sales Data** dataset, then select **Properties**.
- 2 In the Sales Data window, select **Security Access** from the left pane.
- 3 In the Search box type **mark**, then select **Mark Anderson**.
- 4 Select **Add Filter** next to **Mark Anderson**.



- 5 In the **Based on** drop-down, select **Subcategory**.
- 6 In the **Choose elements by** drop-down, select **Selecting in list**.
- 7 From the list in the right pane, select **Boots**, **Dress Shoes**, and **Heels**.
- 8 Click **OK**, then click **Save** to save your security filter selections.

9 In the Search box type **jen**, then select **Jen Thompson**.

User	Access Level	Action Buttons
Developers	FullControl	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)
Everyone	Custom	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)
Public / Guest	Custom	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)
Administrator	FullControl	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)
Jen Thompson	View	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)
Mark Anderson	View	Browse (checkmark), Execute (checkmark), Write (disabled), Delete (disabled), Control (disabled)

10 For **Jen Thompson**, create a security filter to include **Sandals**, **Slippers**, and **Sneakers**.

New Qualification

Subcategory In List Sandals, Slippers, Sneakers

Based on Subcategory

Choose elements by Selecting in list

In List (radio button selected)

Not In List (radio button)

Search View Selected

Clear All

Boots

Dress Shoes

Heels

Sandals

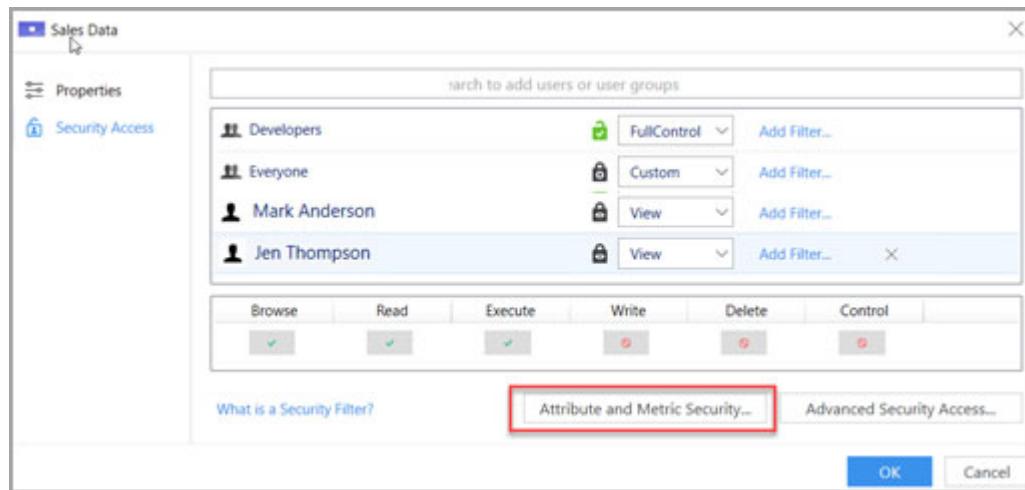
Slippers

Sneakers

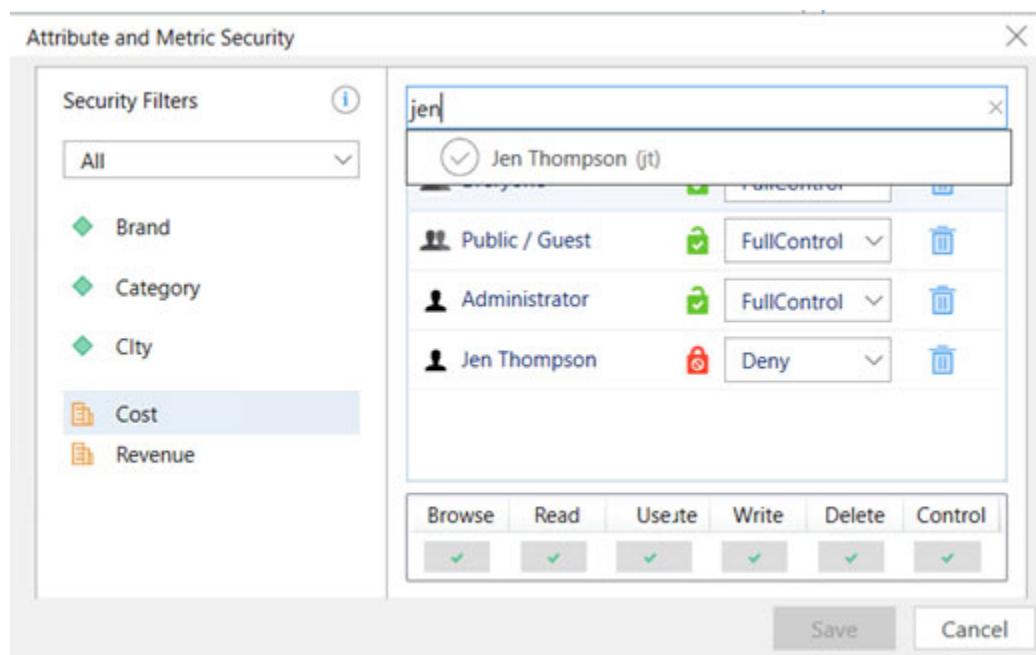
As the final step in your user provisioning, you need to hide the Cost metric from the Sales Data cube for the Jen Thompson user.

Apply metric object permission for Jen Thompson user

1 Select **Attribute and Metric Security**.



- 2 Select the **Cost** metric and in the search box type **jt**, then select **Jen Thompson**.
- 3 In the drop-down for permissions, select **Deny** for this user.



- 4 Click **Save**, then click **OK** to close the Dataset Permissions window.

Exercise 4.9: Verify content privileges and deployment

In this exercise, you verify the content deployed for each user. You first log into MicroStrategy Library as Mark Anderson to view the dossiers deployed for him. You then log out and log back in as Jen Thompson to ensure that the additional documents have been deployed for her. Finally, you log into MicroStrategy Web to author a dossier using the Sales Data cube, ensuring that the security permissions have been granted correctly.

Log in to MicroStrategy Library to check user access rights

1 On your MicroStrategy Landing page, hover on **Library**, then click **Launch**.

2 Log in as Mark Anderson using the following credentials:

- User Name: **ma**
- Password: **ma**

If you are already logged in with a different user, you can log out by selecting Account icon on right corner of your Library menu.

3 Select **Skip** to close the tutorial window and view the three dossiers that were deployed to your library.

4 Click **Retail Sales Report** to start exploring this dossier.

You can use the right and left arrow keys to navigate between the pages of the dossier.

5 Click the **Table of Contents**  icon to view the chapters and pages of the dossier.

You can also filter, bookmark, and share the dossier from your Library.

6 Click the **Library**  icon to the left to go back to your Library.

7 Click the **Account**  icon on the right, then select log out.

8 Log back into Library as Jen Thompson using the following credentials:

- User Name: **jt**
- Password: **jt**

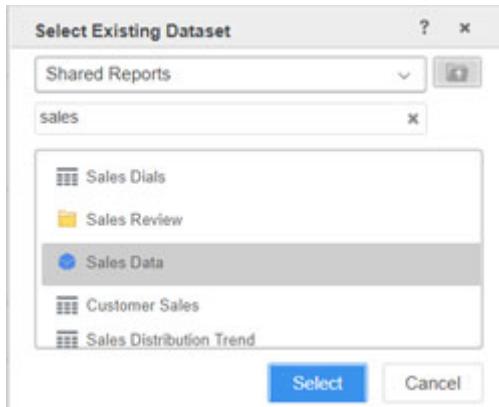
You can view the documents and dossier you deployed earlier for Jen.

Check the dataset permissions you granted to Sales group users

Create a dossier in MicroStrategy Web using the Sales Data cube at the source dataset to check the security permissions.

- 1 On your MicroStrategy Landing page, hover on **MicroStrategy Web**, then click **Launch**.
- 2 Log in with as Mark Anderson using the following credentials:
 - User Name: **ma**
 - Password: **ma**
- 3 From the **View** menu select **Enter Presentation Mode** in the MicroStrategy Tutorial dossier.
- 4 Click **Go to MicroStrategy Web**.
- 5 Click **Create**, then select **New Dossier**.
- 6 In the Datasets panel, select **Existing Dataset**.
- 7 In the search box type **Sales**, then select the **Sales Data** cube.

Make sure you select the cube and not the report with the same name.



- 8 From **Sales Data** dataset, drag **Subcategory** attribute to the **Rows** drop zone.
- 9 Double-click the **Revenue** and **Cost** metrics to add them to the grid visualization.
- 10 Rename Visualization 1 to **Subcategory Sales**.

11 From the Format panel, choose any color you like under Grid Template.

Your dossier should resemble the image below:

The screenshot shows the MicroStrategy Web interface. On the left, there's a sidebar titled 'DATASETS' with a tree view of datasets: Sales Data (In memory), Brand, Category, City, Day, Item, Subcategory, Cost, Revenue, and Row Count - 8 fo...'. The main area displays a grid titled 'Subcategory Sales' with three rows of data:

Subcategory	Revenue	Cost
Boots	483720.49	259910.39
Dress Shoes	521823.00	306214.90
Heels	398230.50	184429.18

Mark is able to view data for the Subcategories you selected in his security filter.

12 Close the dossier without saving it.

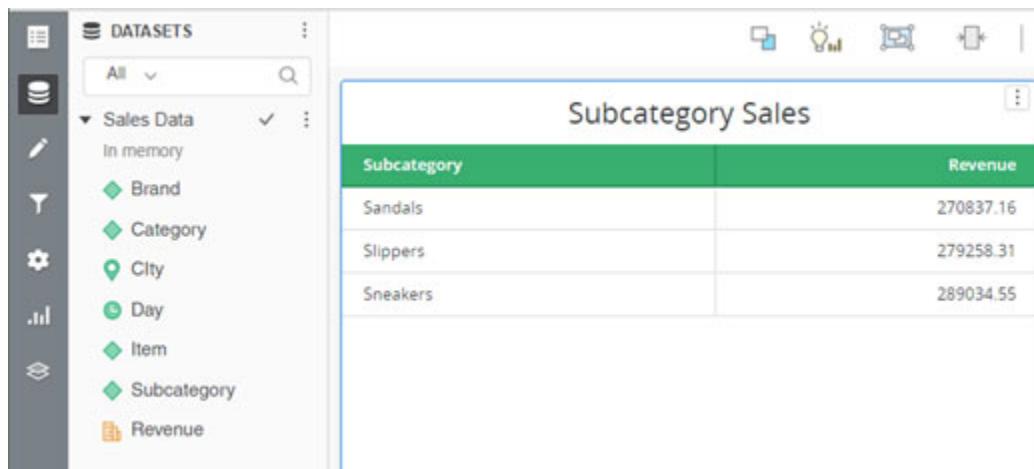
13 From the MicroStrategy Web default page, click the arrow next to Mark Anderson and select **Logout**.

The screenshot shows the MicroStrategy Web homepage. At the top right, there's a user profile for 'Mark Anderson' with a dropdown arrow. Below the header, there's a red 'Create' button and a sidebar with links: 'Recents', 'Shared Reports', 'My Reports', and 'My Objects'. The main content area shows two folder icons: 'Bu...' and 'En...', both owned by Administrator, modified on 8/1/16, and containing several items. To the right, there's a sidebar with links: 'Feedback', 'Help', and a red 'Logout' button.

14 Click **Continue** to go back to the login page.

Creating the same dossier as Jen Thompson

- 1 Log back into MicroStrategy Web as Jen Thompson using the following credentials:
 - User Name: **jt**
 - Password: **jt**
- 2 Select the **MicroStrategy Tutorial** Project.
- 3 From the **View** menu select **Enter Presentation Mode** in the MicroStrategy Tutorial dossier.
- 4 Click **Go to MicroStrategy Web**.
- 5 Click **Create**, then select **New Dossier**.
- 6 In the Datasets panel, select **Existing Dataset**.
- 7 In the search box type **Sales**, then select the **Sales Data** cube.
- 8 From **Sales Data** dataset, drag **Subcategory** attribute to the **Rows** drop zone.
- 9 Double-click the **Revenue** metric.



The screenshot shows the MicroStrategy Web interface. On the left, there's a sidebar with icons for datasets, filters, and other settings. The main area displays a dossier titled "Subcategory Sales". Inside the dossier, there's a table with two columns: "Subcategory" and "Revenue". The data in the table is:

Subcategory	Revenue
Sandals	270837.16
Slippers	279258.31
Sneakers	289034.55

Jen is able to view data for the Subcategories you selected in her security filter.

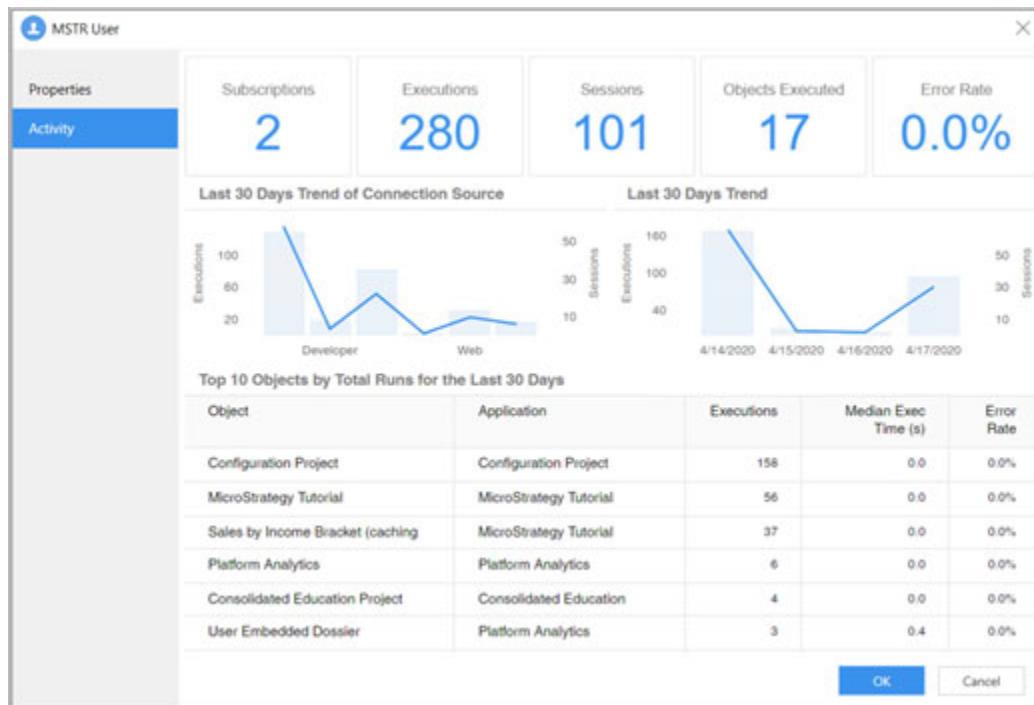
You can see that this dataset does not include the Cost metric since you removed Jen's permission to access this column in the dataset.

Exercise 4.10: Monitoring real-time telemetry data

Platform Analytics is the monitoring tool that captures telemetry data from your MicroStrategy environments. Real-time data from various areas of the MicroStrategy platform including environments, projects, users, sessions, and report executions are collected and stored in a repository. Platform Analytics provides several ways to access, analyze, and act on this telemetry, including out of the box standard dossiers and native telemetry interfaces in MicroStrategy Workstation. In this exercise, you use Workstation to view usage telemetry data at the user level.

Monitor real-time telemetry

- 1 Launch **MicroStrategy Workstation**.
- 2 From the navigation pane, select **Environment**, and connect to your environment.
- 3 From the navigation pane, click **User & Groups**.
- 4 In the Environment panel, select **All Users**.
- 5 Right-click the **MSTR** user, then select **Activity**.



OBJECT MIGRATIONS

Now that you've standardized the environment administration process and set up platform connectivity and security, you can implement a project lifecycle management strategy to create, test, and migrate objects across different environments. With an effective strategy, the Platform Administrator can provide fast, stable, and secure analytics environments for the users, while also ensuring object integrity.

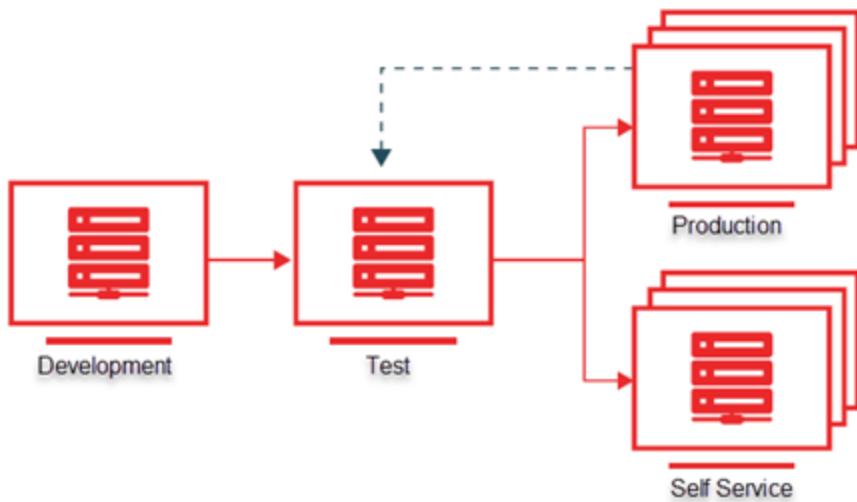
In this chapter, we will review object migrations and analytics environment synchronization.

Managing project lifecycle

As you roll out the MicroStrategy platform, you must create different analytics environments where architects, application designers, and other users can create and test reports, documents, and other objects. As the Platform Administrator, you must develop guidelines for the project lifecycle management strategy. This task involves creating and managing different analytics environments and migrating objects across these environments.

As each business is unique, so is the project lifecycle management strategy. Some of things that the Platform Administrator should take into consideration when developing the guidelines for the project lifecycle management include:

- **Determining the number of analytics environments**—Ideally, when having a governed object development, you should have at least three separate environments—one each for the development, test, and production environments. If users will be doing data discovery and creating a lot of ad hoc reports and other objects, you should have an additional self-service environment.



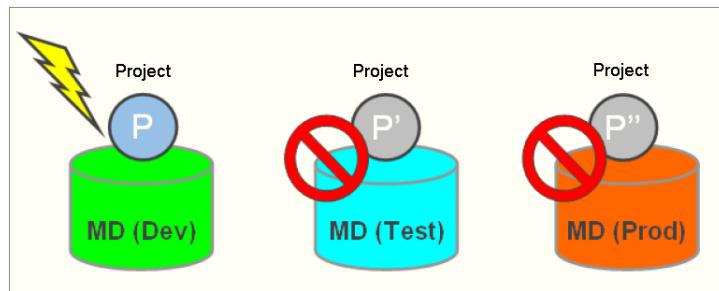
- **Development environment**—Used for creating projects and various application, configuration, and schema objects based on your business requirements.
- **Test environment**—Used for testing the objects that have been created in the development environment. In the test environment, you can connect your MicroStrategy project to a development warehouse for initial testing. However, when conducting user acceptance testing, it is recommended that you test against the data from the production warehouse.

If testing reveals that objects have been created correctly, you migrate them for use in the production environment. However, in case of any issues, you modify the object again in the development environment and then retest it in the testing environment. This cycle continues until your testing indicates that the object has been created correctly and is ready for migration to the production environment.

- **Production environment**—After the objects have been tested and shown to be ready for use in an environment accessible to users, you copy them to the production environment. The production environment is your “live” environment. It consists of the project used by most of the users in

your company. It provides up-to-date reports and tracks various business objectives.

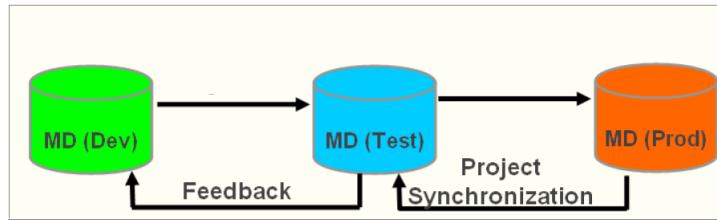
- **Self-service environment**—A separate MicroStrategy project where users can create their reports, documents, and other objects on their own.
- **Creating related projects**—Specify how the platform administration team will create the development, test, and production environments. Ideally, after creating the production project using Developer, you should duplicate it using the MicroStrategy Project Duplication wizard to create development and test projects so that all three projects are “related”, i.e., have the same project schema ID.
 - ❖ You can use tools such as Object Manager and Project Merge Wizard to migrate objects across related projects.
- **Determining the type of changes allowed in each environment**—Specify what kind of changes will be allowed in each environment. Typically, it is recommended that you create or make changes to objects only in development environment; no changes should be made in the test or the production environment.



The only exception to this rule is for changes to the users’ security; these changes can be made in the production system. For example, you can modify the object permissions and access control list (ACL) in the production environment based on the specifications developed in coordination with other Intelligence Center users.

The reason for this exception is that the security settings may be set up differently in the development and production environments. In addition, the development environment may not have all the users, groups, security roles and other security objects that are set up in a production environment. Consequently, it could be risky to assume that the security parameters are defined properly in the development environment and exactly mirror them in the production environment. As a result, it is always important to verify the security settings in the production environment to ensure users have appropriate access to the MicroStrategy application and the warehouse data.

- **Enforcing one-way object migrations**—In coordination with other Intelligence Center teams, set up a process to ensure that only one-way migration of objects is allowed—from the development environment to the test environment, and then to the production environment as shown in the following image:



Exceptions to this rule may be made in certain situations such as the following:

- Upon testing in the test environment, you identify issues with an object. In this scenario, you must resolve the issues in the development environment and re-test it in the test environment. If all issues have been resolved, the object can then be migrated to the production environment.
- An object (such as a document) that has been deleted in the development environment but you now need it again; in this case, you can restore it from the production environment.
- A user creates an object (such as a report or a document) first in his personal folders in the production environment and later that object is needed in the development environment. This could happen in situations where the development environment does not have sufficient data to test documents or reports, or the user creates the object on an ad hoc basis but going forward, it needs to be governed and shared with other users.
- In Self Service environments, you have an application object (such as a report or a document) that can no longer be maintained by a user and requires development environment users to inherit it. In such a case, the object can be moved to the development environment.
- Using Undo Object Manager packages to move application objects to the development environment in case of any issues with migration.
- **Environment synchronization**—You should synchronize objects in the production environment with those in the test environment, and then synchronize the development environment with the test environment.
- **Setting up UAT environment**—Identify the process that the platform administration team will use to get users' acceptance before migrating objects from the test to the production environment.

To facilitate user acceptance, you can either mandate the use of the test environment, or have a separate user acceptance test (UAT) environment, taking MicroStrategy licensing requirements into consideration so that you stay in compliance with your software agreement. In both cases, the environment used for the user acceptance testing should be very similar to the production environment with respect to factors such as hardware, concurrency, and so forth.

In coordination with other Intelligent Center users, you should also develop standards for the data source that will be used for user acceptance testing. Typically, it is recommended that you perform user acceptance testing using the data from the production warehouse.

- **Defining security requirements**—Identify what type of user group structure and security roles will be used in each environment.

The user groups and security roles are typically different in each environment to ensure the separation of duties. The development environment should allow access to only administrators and developers. If needed, business users can be provided view access.

The test environment should allow access to only administrators and test users. The production environment should allow access to only administrators and analysts.



Any additional changes made subsequently in the production environment should be made on an exceptional basis and documented.

- **Recommending object migration tools**—Recommend tools that the platform administration team will use for executing and object migrations. You can use tools such as Object Manager and Project Merge Wizard for object migrations, and Integrity Manager for validating migrations.

Implementing object migrations: Using project releases and change management systems

An important aspect of an effective project lifecycle management strategy for object promotion across different environments should be to migrate the project objects as part of project releases (PR), i.e., development and migration of project objects as bundles. The purpose of project releases is to migrate multiple objects at the same time at a specified schedule rather than promoting individual objects as soon as they are developed. You should create Object Manager packages to move objects as releases from one environment to another.

Use of project releases provides the following advantages:

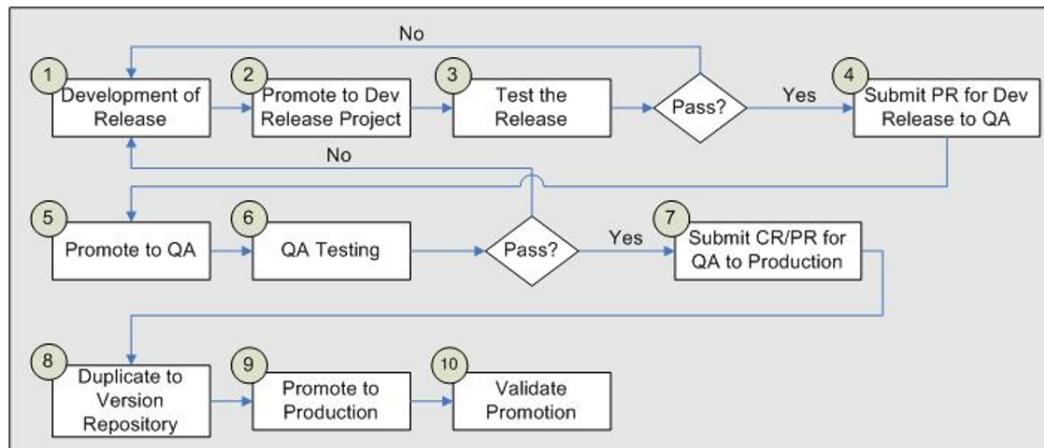
- **Adds structure to the project lifecycle**—As releases exit a phase, related resources are freed to begin work on the next bundle. For example, after the developers have completed their development work for a given release, they are free to start working on the next set of development objects.
- **Facilitates planned migration schedules**—Project release management facilitates planned migration schedules among different environments while making it easier to communicate the project release status to all pertinent stakeholders. Users know what to expect, and can be trained in a timely manner to utilize new project content.

Another important aspect of an effective project lifecycle management strategy is to coordinate with application designers, architects, and system administrators in implementing a change management system. Such a system should allow for:

- Entering/sending change requests (CR)
- Monitoring change requests and communicate with the change request reviewer
- Approving/denying change requests (or not)
- Signing off change requests

The platform administration team will use this system to ensure proper analytics environment governance and for identifying objects dependencies and conflict resolutions.

The following shows a sample of steps that may be involved in object migrations:



Best Practice

Best practices for object migrations

- 1 **Creating environments**—When creating environments, it is a good practice to first create the production project and the metadata. Create the development and test projects by duplicating the production project. Creating the projects in this manner ensures that they have the same project schema ID and are therefore considered as related projects.

Similarly, copy the production metadata, and use that to restore metadata in the development and test environments. After restoring, clean up the development and test projects, including removing any redundant users, folders, and objects.

- 2 **Setting up user access**—The development environment should allow access to only administrators and development users. If needed, business users can be provided limited view access.

The test environment should allow access to only administrators and test users.

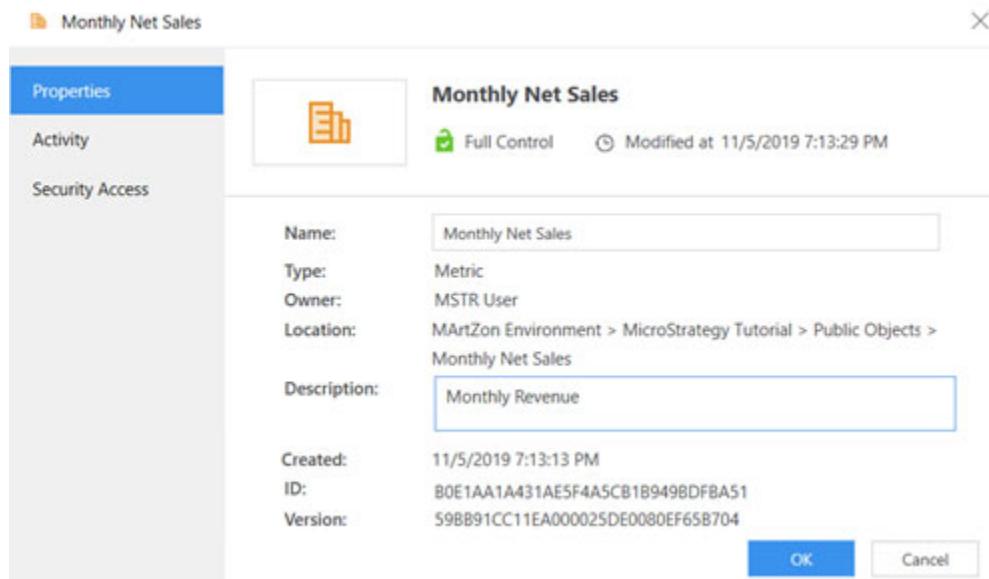
The production environment should allow access to only administrators and analysts.

- 3 **Specifying responsible party for making ACL changes**—Specify which Intelligence Center team will make appropriate changes to a development object's ACLs. In development environment, since each developer has ownership over the object he creates, you can have that developer make appropriate changes to the object ACLs. Alternatively, implement a process to have the developers provide information about the required ACL changes to the platform administration team which can then update the object's ACLs.

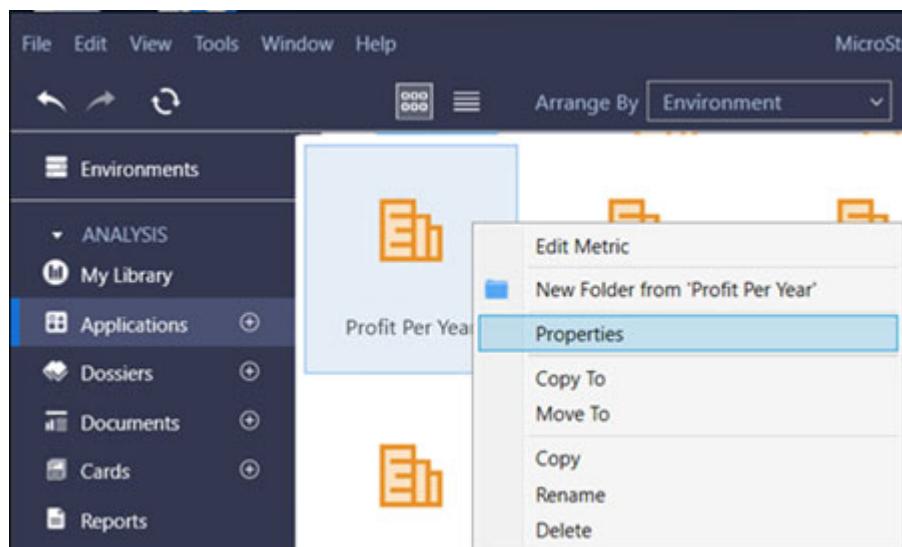
Regardless of who changes a development object's ACLs, as the QA team users (UAT users) test the object against the production data, a process should be set in place to have the platform administration team verify that the object ACL is set up correctly so as to prevent unauthorized data access. This process also ensures that the appropriate Quality Assurance (QA) team members and analysts have the required privileges and permissions to work with the objects in the test and production environments, respectively.

- 4 **Developing naming convention**—In coordination with the Intelligence Center architects, develop an appropriate naming convention for the development objects and ensure that all objects conform to that standard. The naming convention should be consistent and scalable, and take the business context of the object into consideration. You can also include a Description and Long Description with each object in MicroStrategy metadata.

For example, instead of naming a metric as Sales Level Metric, which may be meaningless to most analysts, name it as Monthly Net Sales metric. Similarly, the Description could include the definition of the metric, while the Long Description could include the change history.



To access metric editor using Workstation, launch Workstation, from the navigation pane select **Applications**, then navigate to the folder where your metric is stored. Right-click the metric, then select **Properties**.



5 Reusing existing objects in development—The Platform Administrator should implement formal policies and procedures for each environment to ensure projects in each environment are kept stable, clean and robust. You

should coordinate with the Intelligence Center application designers and architects to ensure that they avoid the creation of redundant objects. If an object already exists, the application designers in the development environment should use it rather than re-creating or duplicating it. Reuse of existing objects prevents creation of unneeded, duplicate objects and keeps your environment clean.

6 Object promotion strategy using designated folders for storing development objects—Assign designated folders where developers can store their objects. For example, it is a good practice to have three different folders in the development project: development, review, and promotion folders.

- **Development**—Each developer should have his own development folder to which only he has the write permissions while other developers have read permissions.
- **Review**—Create review folders to which each developer can move his developed objects upon creation.
- **Promotion**—After review, the developed objects are moved into promotion folders where object manager packages can be created for migrating objects to the test environment.

7 Performing regression testing—The MicroStrategy semantic layer consists of two main groups of objects—schema and application. The SQL engine uses both types of objects to generate SQL for a specific database platform. A change in an object can influence the outcome of the SQL generation, sometimes intentional and sometimes not.

After each update of the semantic layer (i.e., every time new or changed objects are moved into the test and production environments), perform regression testing using tool such as Integrity Manager to make sure that the SQL and data results of existing reports, documents, dossiers, and Intelligent Cubes and reports are not unintentionally changed.

A good practice is to create a standard set of reports that access the important facts and hierarchies. This set can be reused every time the regression tests need to be executed.



The Analytics Architect must sign off on this test set, preferably after each major change of the semantic layer.

Exercise 5.1: Duplicate the MicroStrategy Tutorial project

For the management of your application life cycle, you typically create multiple environments to facilitate development, testing, and production use. You can then move objects from one environment to another such as for testing or for use in production environment.

In this exercise, you will use Object Manager to duplicate the existing MicroStrategy Tutorial project to create a development project (MicroStrategy Tutorial – Dev).

Duplicate the MicroStrategy Tutorial project

- 1 On the Windows machine in your cloud environment, right-click the Windows **Start** button and select **Search**.
- 2 In the **Search Windows** box, search for and click **Object Manager**.
- 3 In the Open Project Source window, click **Cancel**.
- 4 In MicroStrategy Object Manager, on the **Project** menu, select **Duplicate Project**.
- 5 In the Project Duplication Wizard, in the Welcome window, click **Next**.
- 6 In the Project Duplication – Source Project Location window, in the **Available Project Sources** drop-down list, select **MicroStrategy on AWS I-Server**.
- 7 Under Authentication, type the login credentials for your administrator. For this exercise, use the login credentials listed in the Welcome to MicroStrategy on AWS email.
- 8 Click **Next**.
- 9 In the Project Duplication – Source Project Selection window, under **Available Projects**, select the **MicroStrategy Tutorial** project.
- 10 Click **Next**.
- 11 In the Project Duplication – Duplicate Project Location window, in the **Available Project Sources** drop-down list, select **MicroStrategy on AWS I-Server**.

- 12** Under Authentication, select **With the login id and password provided below.**
- 13** Type the login credentials for your administrator. For this exercise, use the login credentials listed in the Welcome to MicroStrategy on AWS email.
- 14** Click **Next.**
- 15** In the Project Duplication - Duplicate Project Creation window, in the Destination project name box, type **MicroStrategy Tutorial - Dev** as the name of the duplicated project. You can overwrite the description in the Destination project description box if you want.
- 16** Click **Next.**
- 17** In the Project Duplication - Select objects to duplicate window, under Project objects, accept the default option, **All objects.**
- 18** Scroll down to the bottom, and under Profiles, select the **Skip empty profile folders** check box.

Excluding empty profile folders can result in faster performance time during project duplication, especially for metadata with a large user population.
- 19** Click **Next.**
- 20** In the Project Duplication – Process Options window, accept the default options and click **Next.**
- 21** In the Project Duplication – Viewing Options window, clear the **View event log concurrently** check box, then click **Next.**
- 22** In the Project Duplication – Log Options window, clear the **Event log**, **Inconsistent object log**, and **Statistics log** check boxes, then click **Next.**

If a message is displayed about overwriting the event log, click **Yes.**
- 23** In the Project Duplication – Summary window, click **Finish**. A window opens and displays the progress of the duplication.

It takes several minutes to duplicate a project.
- 24** When the duplication is complete, a message displays specifying that duplication is finished. Click **OK.**
- 25** Click **Exit** to close the Project Duplication Wizard.

Exercise 5.2: Create an Object Manager package for an Intelligent Cube

The MicroStrategy platform empowers users with the ability to create ad hoc reports, dossiers, and documents. At times, these user-created objects need to be shared with the rest of the enterprise users and their development going forward needs to be governed.

As part of this exercise, you change the default home page for the Tutorial project. You then log into MicroStrategy Web as *ffdemo* user and create an MTDI cube in the MicroStrategy Tutorial project by importing Forecast Data Excel file provided by your instructor. As the users can only save objects in their personal folders in the production environment, save the cube as Forecast Data Cube in the My Reports folder.

This cube is deemed to be useful to other enterprise users and needs to be governed going forward. To do so, you will access the Windows machine in your cloud environment as the *mstr* user.

Next, as the users' personal folders are not displayed by default, you will modify the Object Manager preferences to display the hidden Profiles folder, and then create an update package containing the new cube and save the package in the *MSTR* folder.

Lastly, you will import the Object Manager package in to the MicroStrategy Tutorial - Dev project and move the cube to the Public Objects\Reports folder.

Download the **Forecast Data.xlsx** file

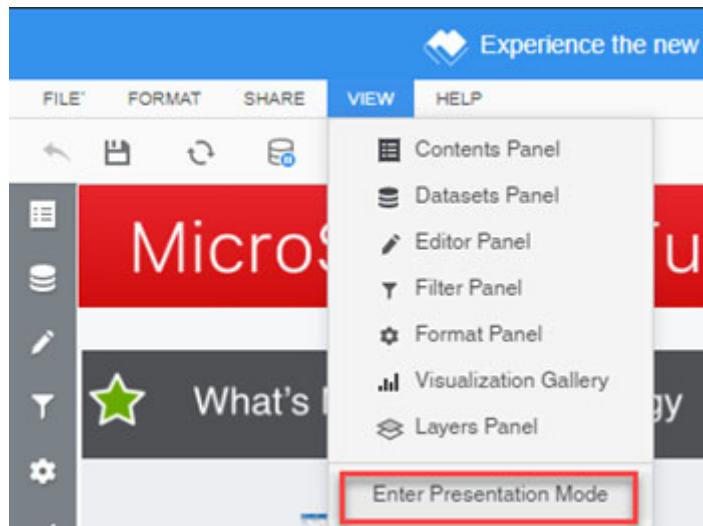
- 1 Download the **Forecast Data.xlsx** file provided by your instructor to your local machine.

Access MicroStrategy Web and import data from a local file

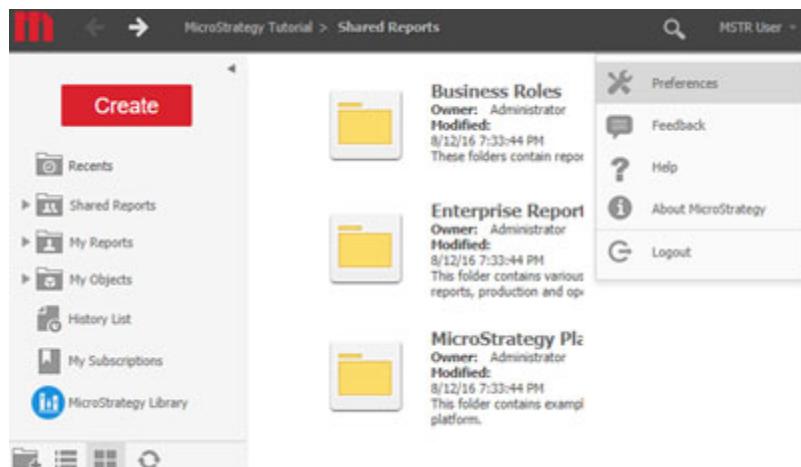
- 2 On the landing page, hover over **MicroStrategy Web** and click **Launch**.

The MicroStrategy Tutorial dossier is launched as your default home page.

3 From the **View** menu, select **Enter Presentation Mode**.



- 4 On the bottom right of your screen, click **Go to MicroStrategy Web**.
- 5 Click the down arrow next to **MSTR User**, then select **Preferences**.



- 6 For Default start page, select **Home**.
- 7 Scroll down to the bottom of the page and click **Apply**.
- 8 From the left pane, select **Project Defaults**.
- 9 Under Default start page, select **Home**.
- 10 Scroll down and click **Apply**.

- 11 From the top navigation menu, select **MicroStrategy Tutorial** to view the MicroStrategy Web home page.
- 12 Click the down arrow next to **MSTR User**, then select **Logout**.
- 13 Click **Continue**, then log in to the **MicroStrategy Tutorial** project as **ffdemo** user. Enter the same password used for the mstr user.
- 14 On the Home page, click **Create**, then select **Add External Data**.
- 15 In the Connect to Your Data window, click **File from Disk**.
- 16 Click **Choose files** and select the **Forecast Data.xlsx** file.
- 17 Click **Open**.
- 18 Click **Finish** and save the cube as **Forecast Data Cube** in the **My Reports** folder.
- 19 Close the Start your analysis window.

An MTDI cube named Forecast Data Cube is created in the Intelligence Server memory.

Access Object Manager

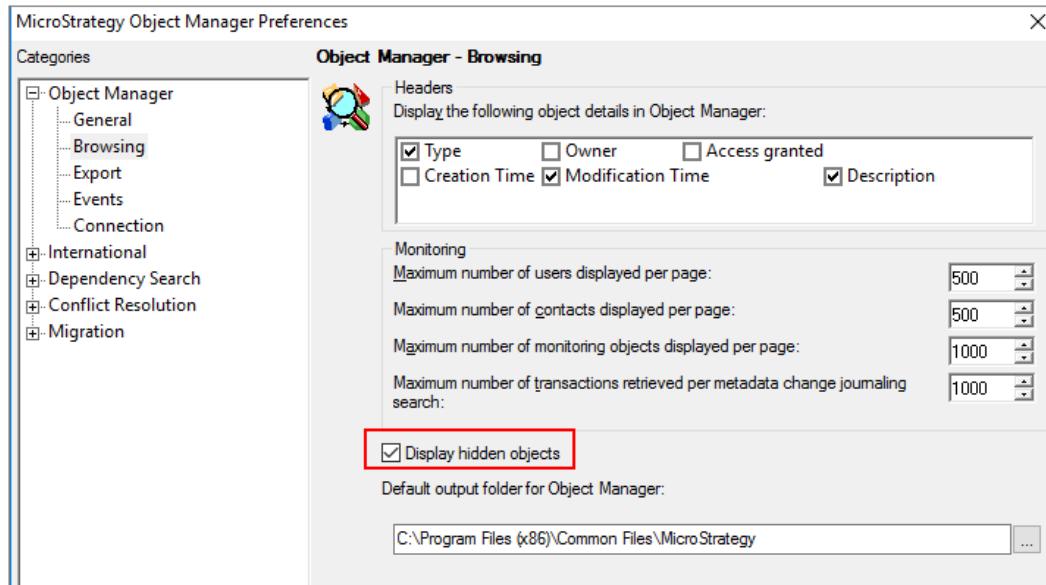
In production environments, users typically can save objects only in their personal folders which are hidden to other users by default. You will now use Object Manager to first display the hidden user folders and then create your update package.

- 1 On the Windows machine in your cloud environment, right-click the Windows **Start** button and select **Search**.
- 2 In the **Search Windows** box, search for and click **Object Manager**.
- 3 In the Open Project Source window, select the check boxes for the **MicroStrategy on AWS I-Server** project sources and click **Open**.
- 4 In the Login window, type the credentials for the **mstr** user listed in the Welcome to MicroStrategy on Cloud email. Click **OK**.

Display the hidden Profiles folder

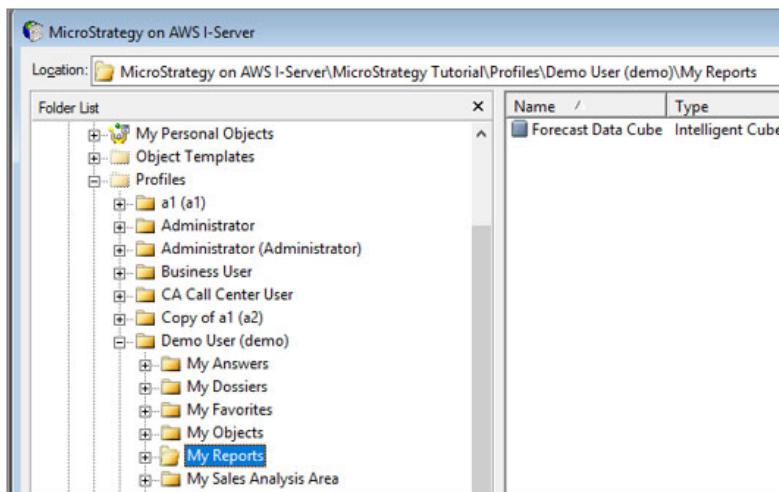
- 5 In Object Manager, on the **Tools** menu, select **Object Manager Preferences**.

- 6 In the MicroStrategy Object Manager Preferences, under **Object Manager**, select **Browsing**. Then select **Display hidden objects** and click **OK**.



You should now see the Profiles folder which was previously hidden.

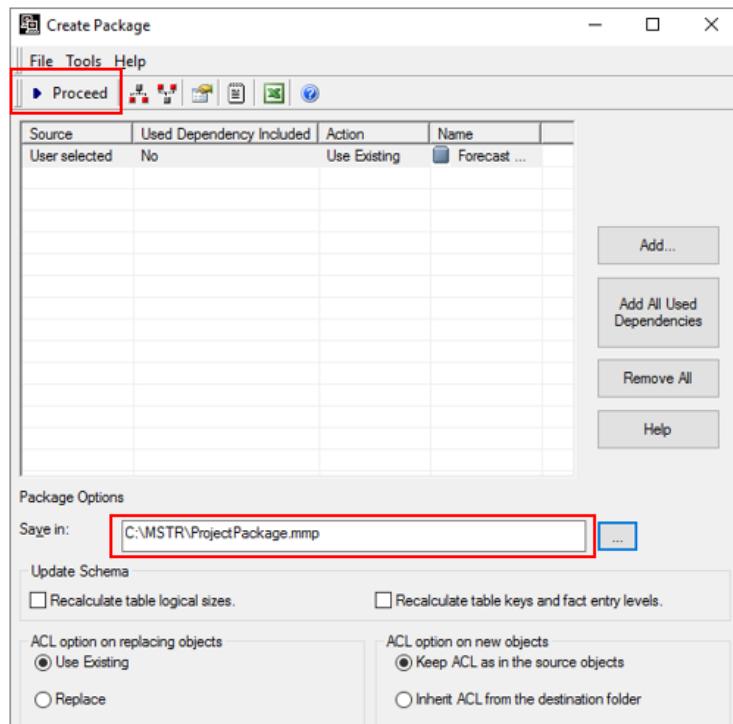
- 7 In the MicroStrategy Tutorial project, under **Profiles**, locate the **ffdemo User** folder, and then select **My Reports**—you will see the Forecast Data cube in that folder.



Create the update package

You will now use Object Manager to create your update package.

- 1 In Object Manager, select the **MicroStrategy Tutorial** project.
- 2 On the **Tools** menu, select **Create Package**.
- 3 Click **Add**.
- 4 In the Add Objects window, in the Available objects drop-down list, navigate to the **\Profiles\ffdemo (ffdemo)\My Reports** folder.
- 5 Select **Forecast Data Cube** and click the **>** button to move the report to the Selected objects pane. Click **OK**.
- 6 In the Create Package window, under Package Options, in the **Save in** box, change the folder where the update package file will be saved to **C:\MSTR** and save the package as **ProjectPackage.mmp**. Click **Proceed**.



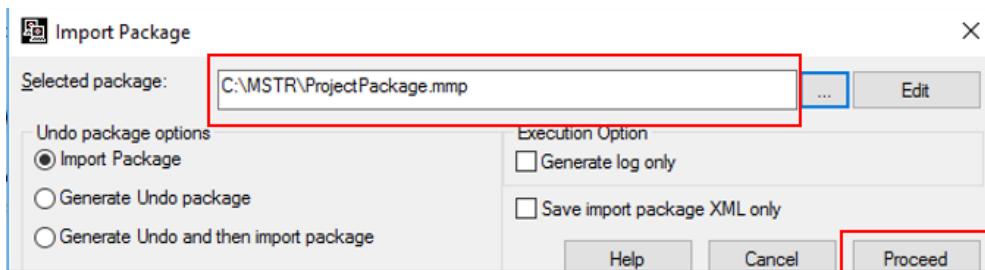
Object Manager starts creating the update package and displays a message about successfully creating the package on completion.

- 7 Click **OK**.

Import the package

You will now use Object Manager to import the package in to the MicroStrategy Tutorial - Dev project.

- 1 In Object Manager, access the **MicroStrategy Tutorial - Dev** project.
- 2 On the **Tools** menu, select **Import Package**.
- 3 Browse to the saved update package file location, **C:\MSTR**, select your package and click **Open**. Then click **Proceed**.



- 4 In the MicroStrategy Object Manager window, click **Yes** to the message regarding proceeding with the importing of the objects.
All objects in the update package are copied to the project. An Operation Successful message displays when the import process is complete. Click **OK**.

Move the cube to make it available to other users

The import process copies Forecast Data Cube to the My Reports folder of the ffdemo user. You will now move the cube to the Public Objects\Reports folder.

- 5 In Object Manager, in the **MicroStrategy Tutorial - Dev** project, access the **Profiles\ffdemo (ffdemo)\My Reports** folder. Right-click the **Forecast Data Cube** and select **Cut**.
- 6 Paste the cube in the **Public Objects\Reports** folder.
The cube is now available to all users in the project.

- 7 Exit Object Manager.

Publish the cube

To enable users to use the imported cube, you must publish it.

- 1 In MicroStrategy Web, log in to the **MicroStrategy Tutorial - Dev** project as **mstr** user.
- 2 In the **Shared Reports** folder, right-click **Forecast Data Cube**, then select **Republish** to publish the cube.
- 3 In the Republish Forecast Data Cube window, browse to the **Forecast Data** Excel file provided by your instructor and select **Refresh**. Click **Done** after the cube is republished and then log out of the **MicroStrategy Tutorial - Dev** project.

PLATFORM MONITORING

An Intelligent Enterprise consistently monitors performance of all their analytics environments to maximize efficiency and utilization of all BI applications. As the Platform Administrator, you must develop monitoring standards and guidelines for MartZon deployment environments. This helps your team operate and monitor MartZon's analytics environments to ensure they are running properly.

In this chapter, we review:

- Using Platform Analytics to monitor your environment and object utilization to identify unused objects in projects in real-time.
- Analyzing system usage.
- Detecting environment issues when the MicroStrategy platform is updated.
- Monitoring licenses and MicroStrategy License Manager.

Guide environment monitoring

Analytics platform environments must be monitored to ensure that performance standards are maintained and data is consistently delivered to users. To monitor environments, you should require your team to track statistics, identify baselines

for the various performance metrics, and quickly diagnose performance anomalies.

To ensure that MicroStrategy projects keep up with demand and process data without errors, develop guidelines to help monitor environments on a daily basis to gain insight on your user activity. A lack of monitoring standards can result in deteriorating environment performance and declining adoption rates. Ensuring your platform maintains high performance enhances user experience and productivity.

As the Platform Administrator, you should develop guidelines for:

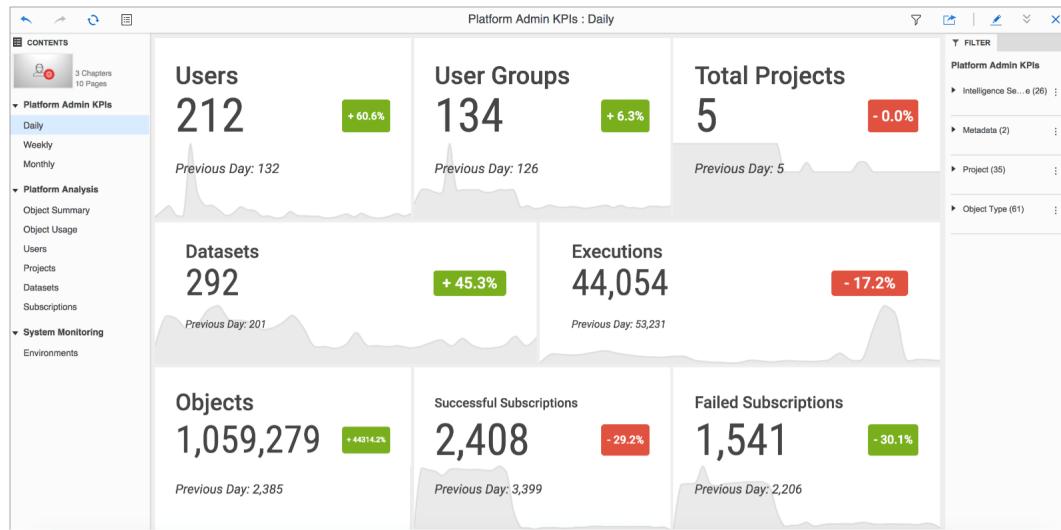
- 1** Checking the status of platform servers (such as Intelligence Server, MicroStrategy Web, MicroStrategy Mobile, and Collaboration Server) to ensure they are up and running.
- 2** Monitoring user connections, database connections, and job statuses using tools such as Job Monitor and other administrative monitors in Developer to check long running session or job bottlenecks.
- 3** Analyzing cache and cube utilization in Platform Analytics.
- 4** Purguing redundant caches and cube files with tools such as Developer to ensure that the Intelligence Server memory is not depleted.
- 5** Auditing the analytics environment configuration using tools such as Developer, MicroStrategy configuration files, and server logs, and tracking changes over time to establish a baseline.
- 6** Configuring analytics environments to log statistics using Platform Analytics for analyzing statistical data on environment usage.
- 7** Coordinating with the Intelligence Center architects and other users regarding issues or areas for improvement.
- 8** Ensuring that the platform environments are stable and governed to ensure that business objectives are met.

9 Assessing important platform administration KPIs such as the ones shown in the following table:

KPI Category	KPI	Administrative action
Environment	<ul style="list-style-type: none">Number of total environmentsNumber of active environments	<ul style="list-style-type: none">If the number of active environments is less than total environments, identify the root cause and decommission the obsolete environments
Project objects	<ul style="list-style-type: none">Number of datasetsNumber of active datasetsNumber of total objects (schema and application)Number of active objects (schema and application)	<ul style="list-style-type: none">If the number of active objects or datasets is less than the total objects or datasets, identify the root cause and work with the Analytics Architect to plan a clean-up strategy
Users and projects	<ul style="list-style-type: none">Number of total usersNumber of active usersNumber of total projectsNumber of active projects	<ul style="list-style-type: none">If the number of active users is less than total users, identify the root cause and disable inactive usersIf the number of active projects is less than total projects, identify the root cause and work with the Analytics Architect and Application Architect to plan the deletion of the inactive projects or merge similar projects to reduce the metadata size. Alternatively, you can unload inactive projects
Subscriptions	<ul style="list-style-type: none">Number of successful subscriptionsNumber of failed subscriptions	<ul style="list-style-type: none">Identify the root cause of subscriptions failures, contact the owners, and either remove the failed subscriptions, or resolve the reason for failureInventory the successful subscriptions and check if those are still current and in use by the recipients. Plan a clean-up strategy for unused subscription

You can use out-of-the-box dossiers that ship with Platform Analytics, or create customized dashboards from telemetry data collected in the Platform

Analytics data warehouse. The following image presents an example of a custom dossier showing daily platform administration KPIs:



Analyzing real-time data: Platform Analytics

Monitoring MicroStrategy in real time enables you, as the Platform Administrator, to deliver an optimal user experience. Without proactive monitoring, organizations can become trapped in a perpetually reactive mode: unable to respond to issues and problems until they are too disruptive to correct and negatively impact the business.

Platform Analytics is a MicroStrategy platform monitoring tool that you can utilize to identify areas for improvement, anticipate problematic behavior, and identify possible adjustments to administrative settings.

You can use Platform Analytics to simultaneously monitor and analyze multiple MartZon MicroStrategy environments, such as your development, testing, and

production environments, as well as your various departmental environments, as displayed in the following image:



Accessing Platform Analytics

As the Platform Administrator, you need to analyze the statistics collected by Platform Analytics to understand how your MicroStrategy platform is being utilized. There are several ways to access, analyze, and act on telemetry data collected in Platform Analytics, including out-of-the-box standard dossiers and native telemetry interfaces in MicroStrategy Workstation. Users can access this data using the following methods:

- **By viewing the Platform Analytics data embedded in Workstation:** Using workstation, authorized users are able to gain access to telemetry data related to their usage patterns. Platform Analytics exposes captured data directly in the user interface of Workstation.
- **By running the out-of-the-box Platform Analytics dossiers:** Platform Analytics ships with a MicroStrategy project that provides out-of-the-box dossiers designed to track various aspects of the MicroStrategy platform.
- **By creating your own dossiers:** Platform Analytics also supports the creation of self-service content (dossiers, reports, and documents) which are based on the out-of-the-box schema and application objects included in the Platform Analytics project.

Platform Analytics project: Areas of analysis

Using the Platform Analytics project you can track the following aspects of the MicroStrategy platform:

- **Environment:** MicroStrategy version and hardware characteristics.
- **System:** Configuration and connection information.
- **Projects:** Object statistics and utilization.
- **Users:** Activity and adoption statistics.
- **Objects:** Project object characteristics and utilization.
- **Cubes:** Dependency, sizing, and status.
- **Subscriptions:** Report and dossier distribution information.
- **Errors:** Rates and categories.
- **Licensing:** User privileges and compliance data.

Using the out-of-the-box dossiers, you can determine how your environment can be improved and perform the administrative tasks required to make the desired optimizations. For example, your analysis of the Object Telemetry dossier might indicate that a specific Intelligent Cube is executed much more often than its scheduled refresh interval. Upon further investigation, you might find that the cube execution time is also significantly higher than other cubes. To improve performance and ensure user satisfaction, you might identify a list of users who most frequently publish the cube and consult with them to develop a refresh schedule that maximizes performance.

Another example is the Cache and Cube Monitoring dossier. Using this dossier you can analyze the number of cubes in your environment, the amount of memory they consume, the distribution of cubes across projects, and other

summary level information. You can drill on each cube to see more details including cube dependencies, hit count, last update, and usage trends.



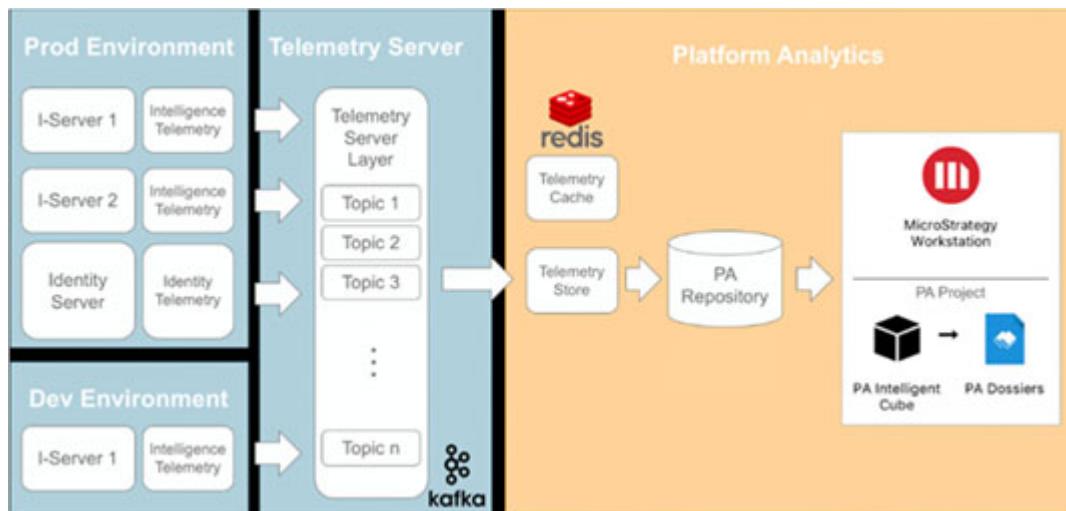
Platform Analytics architecture and services

The Platform Analytics architecture consists of the following components:

- **Intelligence Telemetry**—This component acts as a telemetry producer, sending all the data generated by the Intelligence Server to the Telemetry Server.
- **Telemetry Server**—This component serves as a message broker that receives and temporarily houses all the data sent by the producers.
- **Platform Analytics Store**—This component reads the data that the Intelligence Telemetry producers send to the Telemetry Server layer, transforms this data, and loads it in the Platform Analytics Repository.
- **Telemetry Cache**—This component is used to improve the processing performance of the Platform Analytics Store.
- **Platform Analytics Repository**—This data warehouse stores all the MicroStrategy telemetry processed by the Telemetry Store. This data is then used by the dossiers included with Platform Analytics project.
- **Platform Analytics Project**—This MicroStrategy project contains the out-of-the-box schema and application objects for Platform Analytics, including the standard Platform Analytics dossiers, attributes, metrics, and cubes.

- **Platform Analytics Cube**—This data import cube contains 14 days' worth of Platform Analytics data and is used to feed data to all the standard Platform Analytics dossiers.

The flow of information is displayed in the following image:



The following process describes how MicroStrategy platform information is captured, processed, and distributed by Platform Analytics (PA) in real-time.

- 1 Users log in to MicroStrategy projects, run reports against the Intelligence Server, and perform administrative tasks. Users may also log in to MicroStrategy Badge to access physical and logical assets.
- 2 The Intelligence Server logs statistics to the Telemetry Server layer, which temporarily houses the statistics data.

Tip: When you configure statistics logging, you enable a group of basic statistics such as executions, sessions, prompt answers, subscriptions, and others to be logged. If you need to perform troubleshooting tasks in your environment, you can also temporarily log advanced statistics to track job steps and SQL passes.
- 3 The Platform Analytics Services read messages from the Telemetry Server layer, transform the data, and store it in the Platform Analytics repository.
- 4 In the Platform Analytics project, the Platform Analytics Intelligent Cube retrieves data from the data warehouse at the configured interval (every hour by default). The last 30 days of data are stored in the cube.
- 5 The Platform Analytics project dossiers and Workstation Activity screen retrieve and display data from the Platform Analytics Intelligent Cube.

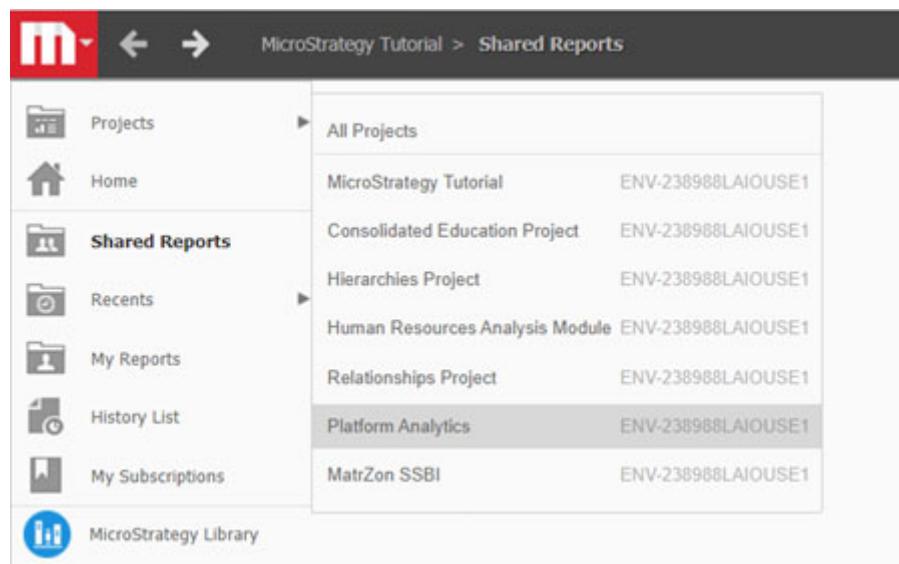
Exercise 6.1: Accessing Platform Analytics dossiers

The Platform Analytics project is already configured in your MicroStrategy on Cloud environment. In this exercise, you access the project and explore some of the predefined dossiers. As the Platform Administrator, you can leverage these dossiers to monitor and optimize the system.

Although your environment was created recently and has a minimal amount of data in the Platform Analytics repository, exploring the predefined dossiers helps you understand how the monitoring tool can help you tune your environment. Additionally, navigating the dossiers might inspire you to create custom dossiers and reports using the attributes and metrics in the project.

Access the Platform Analytics project

- 1 In MicroStrategy Web, click the **arrow** next to the MicroStrategy icon. Then, point to **Projects** and select **Platform Analytics**.



- 2 Open the **Shared Reports\1. Dossiers** folder. This folder contains eight dossiers that can help you analyze environment utilization and performance, and tune the MicroStrategy platform.
- 3 Execute the dossiers and view the individual chapters in each dossier. Notice that each dossier contains several filters that can help you narrow the results.

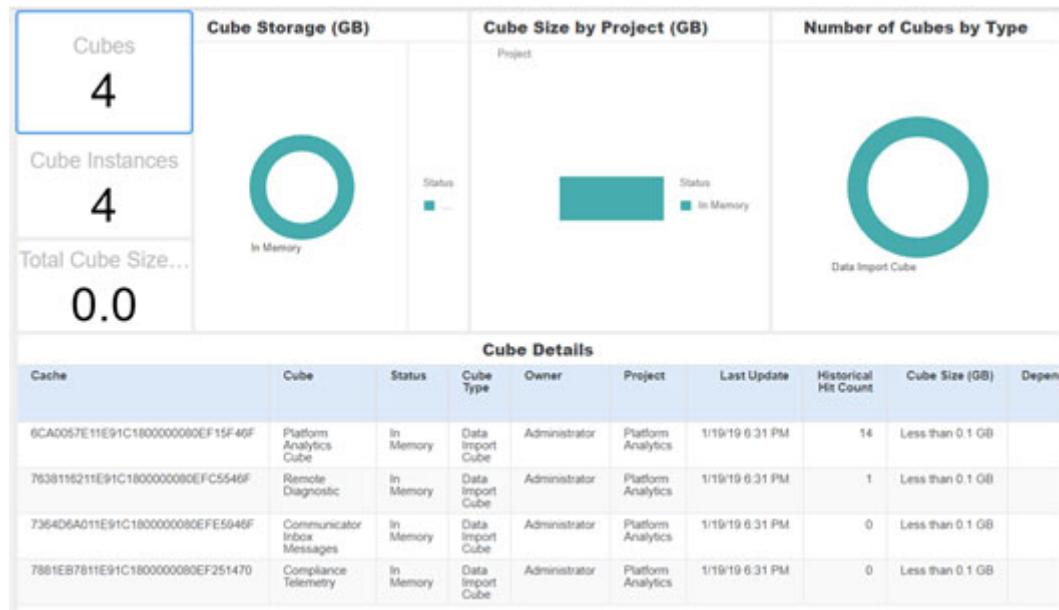
As you view the metrics on each page, think about how you would use the dossier to monitor your own environment.

 Because the environment for this class was created recently, the dossiers may not display any data.

Refresh the Platform Analytics cube

Although Platform Analytics collects environmental statistics in real time, the Intelligent Cube is scheduled to refresh every hour by default and therefore displays new data in the dossiers every hour. You can manually refresh the cube by following the steps in this section.

- 4 Open the **Shared Reports\1. Utilities** folder. This folder contains the Intelligent Cubes on which the dossiers are based.
- 5 Right-click **Platform Analytics Cube** and select **Republish**. The Refresh Platform Analytics Cube window opens.
- 6 Click **Refresh**. The cube refreshes after a few moments. Click **Done**.
- 7 Re-execute the dossiers in **Shared Reports\1. Dossiers** to see if any new data is displayed. For example, the Cube and Cache Monitoring dossier now reflects the number of cube hits that occurred when you executed the dossiers and should look similar to the following image.

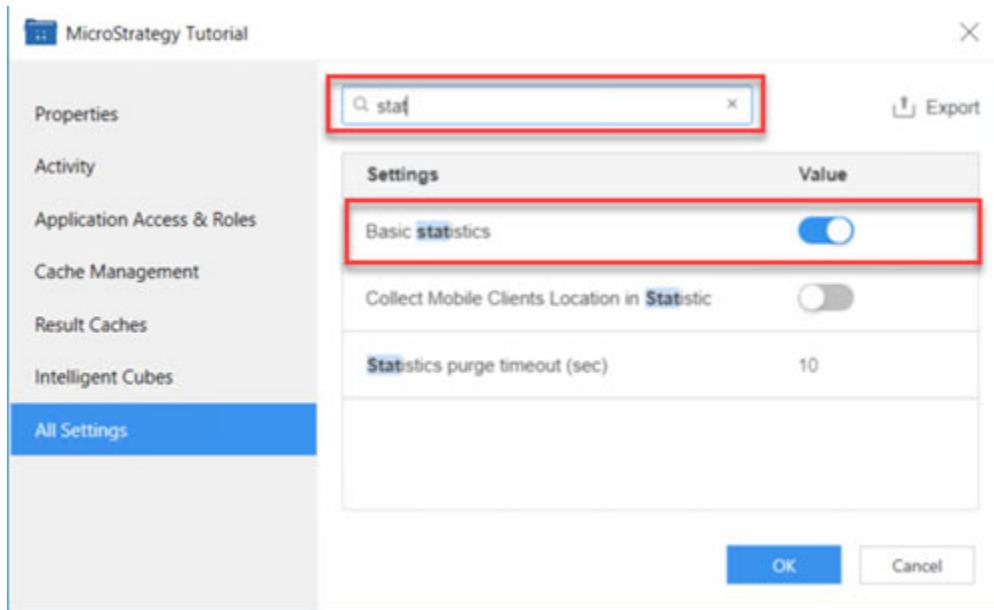


Exercise 6.2: Enable statistics from Workstation for platform monitoring

Platform Analytics is configured by default on MicroStrategy Cloud environments. You enable statistics for each application in your environment that you wish Platform Analytics to collect statistics from.

Enable Statistics using Workstation

- 1 On the Windows machine in your cloud environment, launch Workstation.
- 2 From the navigation pane, select **Environments**.
- 3 Double-click **MartZon Environment** and provide your credentials to connect.
- 4 In the Select Applications window, select **MartZon SSBI** and **MicroStrategy Tutorial** applications., then click **OK**.
- 5 From the navigation pane, select **Applications**.
- 6 Right-click **MicroStrategy Tutorial** application and select **Properties**.
- 7 From the left pane, select **All Settings**, then type **stat** in the top search box.
- 8 Enable **Basic statistics**, then click **OK**.

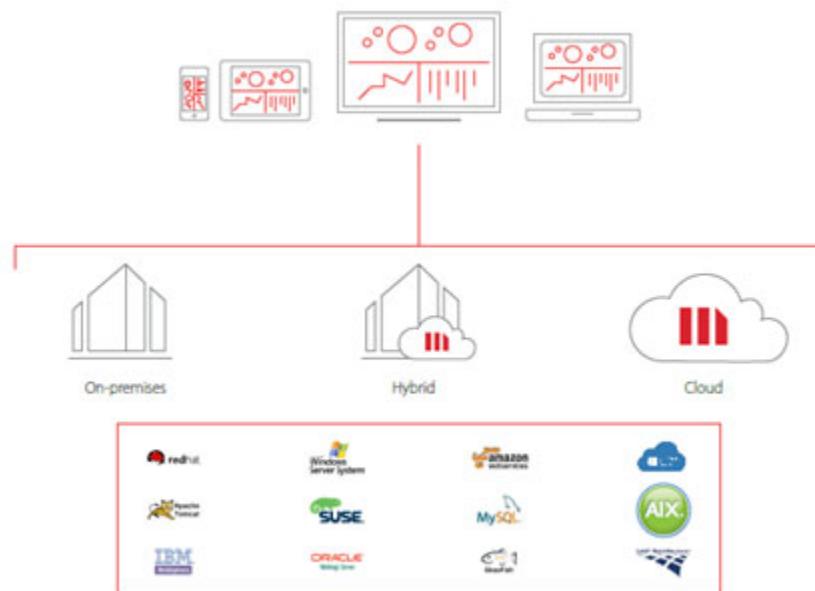


- 9 Repeat the same steps to enable Basic statistics for MartZon SSBI application.

Environment management

MicroStrategy delivers flexible options to deploy analytics across your enterprise with on-premises, cloud, and hybrid options. This enables MartZon to use MicroStrategy Enterprise Cloud to rapidly launch fully configured and ready-to-use enterprise analytics and mobility projects when they need to. As the Platform Administrator, you have the option to deploy and manage the full MicroStrategy platform on dedicated AWS and Azure infrastructures. You can also choose to host your data locally or use cloud storage through Amazon Web Services (AWS) and Microsoft Azure.

MicroStrategy deployment options



Regardless of the environment topology you configure for your organization, you can use Workstation to manage and access all of them in one interface.

Exercise 6.3: Monitoring your environment

In this exercise you connect to your environment using MicroStrategy Workstation. You then view the topology of all nodes and services, and track certificates for the MicroStrategy platform components installed in your environment.

Connect Workstation to your Intelligence Server

- 1 Launch **MicroStrategy Workstation**, if not already open.
- 2 From the navigation pane, select **Environments**.
- 3 Right-click **MartZon Environment** and select **Connect**, if not already connected.
- 4 On the left, select **Monitors** to view the state of all services running in your environment.
- 5 Click the arrow on the right to expand services.



You can monitor the current state of every service on each node, and you can also start or stop individual services.

- 6 If there are any services that did not start in your environment, right-click the service and select **Start**. Then, provide the credentials from your MicroStrategy Cloud email.

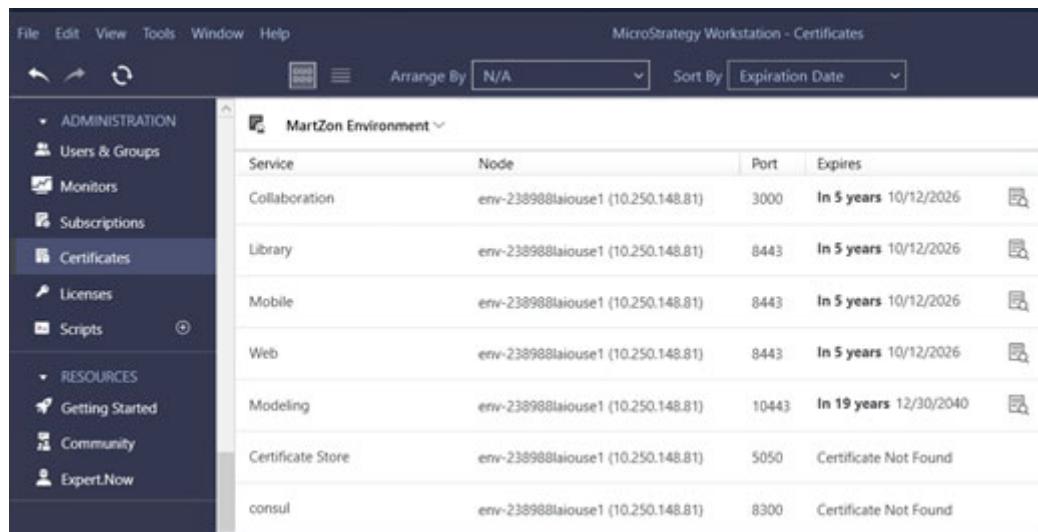


Do not stop any of the running services, as you will not be able to continue with the rest of your exercises.

The screenshot shows the MicroStrategy Workstation interface with the title "MicroStrategy Workstation - Monitors". The left sidebar is titled "Environments" and lists categories like ANALYSIS, APPLICATIONS, DOCUMENTS, CARDS, REPORTS, DATASETS, SCHEMAS, ADMINISTRATION, and RESOURCES. Under ADMINISTRATION, "Monitors" is selected. The main pane shows the "MartZon Environment" with a summary: "MartZon Environment" and "MicroStrategy 2021". Below this is a "Nodes" section. To the right is a list of services with their status: Certificate Store (Started), Collaboration (Green), Export (Green), Identity Telemetry (Red), Intelligence (Green), Library (Green), Mobile (Green), Modeling (Green), Store (Green), Telemetry (Green), Telemetry Cache (Green), Telemetry Consumer (Green), and Telemetry Manager (Green). A header at the top of the main pane says "env-238988laiouse1" and "14 Services | 10.250.148.81" with a "Needs Attention" status.

- 7 On the left, select **Certificates** to view all your environment certificates in one place along with details about where they are stored and when the certificates expire.

- 8 Click the **Preview Certificate** icon to the right of Web Certificate to view the MicroStrategy Web Server certificate details.



Service	Node	Port	Expires	
Collaboration	env-238988laiouse1 (10.250.148.81)	3000	In 5 years 10/12/2026	
Library	env-238988laiouse1 (10.250.148.81)	8443	In 5 years 10/12/2026	
Mobile	env-238988laiouse1 (10.250.148.81)	8443	In 5 years 10/12/2026	
Web	env-238988laiouse1 (10.250.148.81)	8443	In 5 years 10/12/2026	
Modeling	env-238988laiouse1 (10.250.148.81)	10443	In 19 years 12/30/2040	
Certificate Store	env-238988laiouse1 (10.250.148.81)	5050	Certificate Not Found	
consul	env-238988laiouse1 (10.250.148.81)	8300	Certificate Not Found	

- 9 Write down the following information for the Web certificate.

Service ID: _____ Port: _____

Certificate Type: _____ IP Address: _____

Monitoring upgrades: Testing for discrepancies

The Platform Administrator is responsible for upgrading the analytics environments. Whenever the MicroStrategy platform or database environment are updated, analysts and application designers may encounter erroneous behavior in existing datasets. To ensure that datasets continue to function as the project evolves and updates are applied, develop standards in collaboration with the Intelligence Center architects, application designers, and system administrators for performing and testing upgrades. These guidelines should:

- Recommend tools for performing upgrades. For example, you can perform upgrades using MicroStrategy Installation Wizard or automate them using response files.

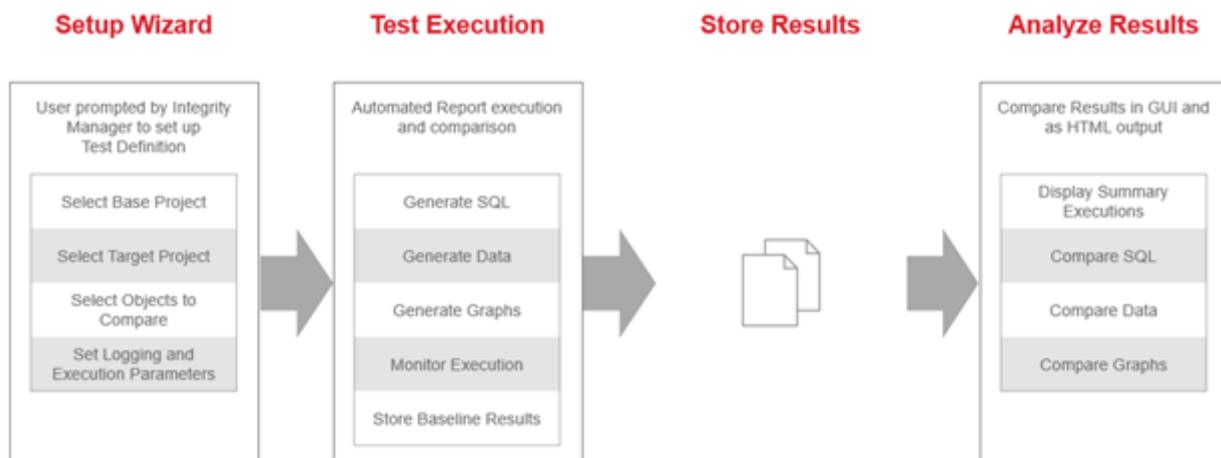


The platform administration team may also need to start/stop analytics environment servers (such as Intelligence Server and Web application server) after applying configuration updates to ensure the changes are applied.

- Recommend tools for testing upgrades. The following section discusses MicroStrategy Integrity Manager and how it can be used for testing upgrades to ensure it was successful.

Testing environment discrepancies due to upgrades: Integrity Manager

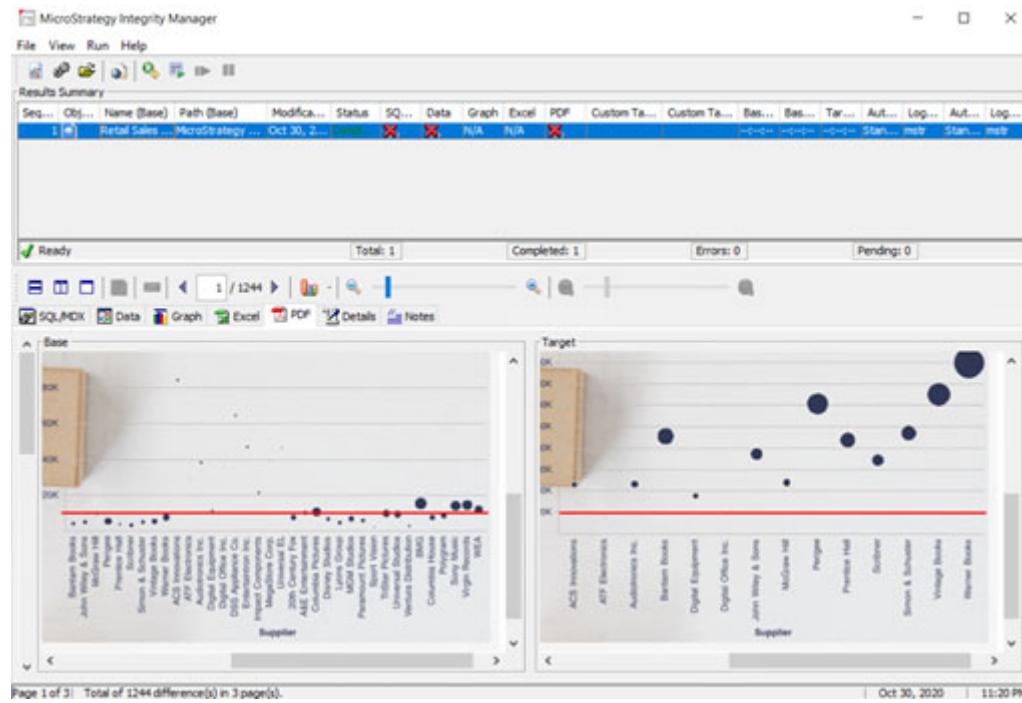
When the platform administration team upgrades projects to the latest version of MicroStrategy, the data delivered to business users must remain consistent and reliable. To identify inconsistencies in datasets, Integrity Manager provides tests to quickly and efficiently compare multiple reports and documents from different environments, and uncovers discrepancies in result sets and output formats. The following image shows the Integrity Manager workflow for selecting a test, executing it, and analyzing the results.



Integrity Manager can be employed to review the impact of upgrades on the generated SQL and data results. A project versus project test displays the report results in upgraded project and compares those results against the same reports in the original project.

The Integrity Manager Wizard walks you through the process of setting up integrity tests. You specify what kind of integrity test to run, what reports or documents to test, and the execution and output settings. Then you can execute the test immediately, or save the test for later use and re-use. In the example below, the PDF export for the dossier displayed in the Base pane on the left

returns more data than the same dossier displayed in the Target pane on the right.



To ensure that datasets retain functionality and display properties when the environment is upgraded, develop a testing strategy that leverages Integrity Manager. Your guidelines should include steps to test datasets, as well as a strategy to analyze the results of those tests. For example, your standards might include the following steps to guide application designers in comparing datasets in distinct environments:

- 1 Determine whether the report is going to be tested against another project, a baseline, or against itself with modified settings.
- 2 Run the Integrity Manager test and identify discrepancies in report results, SQL, or display.

Application designers should follow your guidelines to compare reports in distinct environments and to gauge the possible impacts of an upcoming MicroStrategy platform upgrade.

Exercise 6.4: Test the integrity of reports across projects

You can use Integrity Manager to verify the integrity of objects that exist in two projects.

In this exercise, you create a project versus project test to verify the integrity of reports that exist in both the MicroStrategy Tutorial project and in the MicroStrategy Tutorial - Dev project. After modifying a couple of reports in one of the projects, you run the test in Integrity Manager and compare the report results in SQL/MDX, (grid) Data, and PDF outputs. You will also display the HTML-formatted SQL file. You do not need to save your test configuration.

Test the integrity of reports across projects

Modify reports

- 1 In Developer, log in to the MicroStrategy Tutorial project using the login credentials listed in the MicroStrategy Cloud email.
- 2 Navigate to the **Public Objects\Reports\Subject Areas\Human Resources Analysis** folder.
- 3 Open and edit the **Employee Headcount by Country** report. Add the **Region** attribute to the template.
- 4 **Save** and close the report.
- 5 Edit the **Length of Employment** report. Remove the **Hire Date** attribute from the report. Save and close the report.
- 6 Click the **Minimize** icon to minimize MicroStrategy Developer.

Create a Project versus Project Test in Integrity Manager

- 7 On the Windows desktop in your cloud environment, right-click the Windows **Start** button and select **Search**.
- 8 In the **Search Windows machine**, search for and click **Integrity Manager**.
- 9 When prompted, in the User Account Control window, click **Yes** to open Integrity Manager.
- 10 From the **File** menu, select **Create Test**.
- 11 In the MicroStrategy Integrity Manager Wizard, click **Project versus Project**.
- 12 Click **Next**.
- 13 On the Enter Base Project Information page, provide the following information:

- In the **Server Name** box, type the name of your Intelligence Server. You can find the Intelligence Server name in the hosts file.
- In the **Server Port** box, use the default port number which is **34952**.
- In the **Authentication Mode** list, select **Standard Authentication**.
- In the **Login id** and **Password** boxes, type the login credentials listed in the MicroStrategy Cloud email.
- In the **Project** drop-down list, select the **MicroStrategy Tutorial** project.

14 Click **Next**.

15 On the Enter Target Project Information page, provide the following information:

- In the **Server Name** box, type the name of your Intelligence Server. You can find the Intelligence Server name in the hosts file.
- In the **Server Port** box, use the default port number which is **34952**.
- In the **Authentication Mode** list, select **Standard Authentication**.
- In the **Login id** and **Password** boxes, type your administrator credentials.
- In the **Project** drop-down list, select the **MicroStrategy Tutorial - Dev** project.

16 Click **Next**.

17 In the Select Objects from the Base Project to be included in the Test window, navigate to the **Public Objects\Reports\Subject Areas**. Select the **Human Resources Analysis** folder.

18 Click **Next**.

19 In the Select Prompt Settings window, accept the default options.

20 Click **Next**.

21 In the Select Execution Settings window, accept the default options.

22 Click **Next**.

23 In the Select Processing Options window, select the **SQL/MDX, Data**, and **PDF** check boxes for reports and clear any other check boxes.

For each option that you select, you can analyze your selected reports or documents in that format while reviewing the test results.

- 24** Click **Next**.
- 25** Review the Integrity Manager Wizard test settings, then click **Run** to execute the test immediately, without saving the test.
- 26** On the Results Summary page, compare the results for the two reports (Employee Profitability Analysis and Length of Employment) that you have modified. The analysis of both reports shows different results in all of the Outputs because of the presence of the Region attribute in one of the reports (Employee Profitability Analysis) and the absence of the Hire Date attribute in the Length of Employment report.
- 27** On the toolbar, click **HTML Output**. Results are displayed in the HTML format.

Monitoring licenses: Using MicroStrategy License Manager

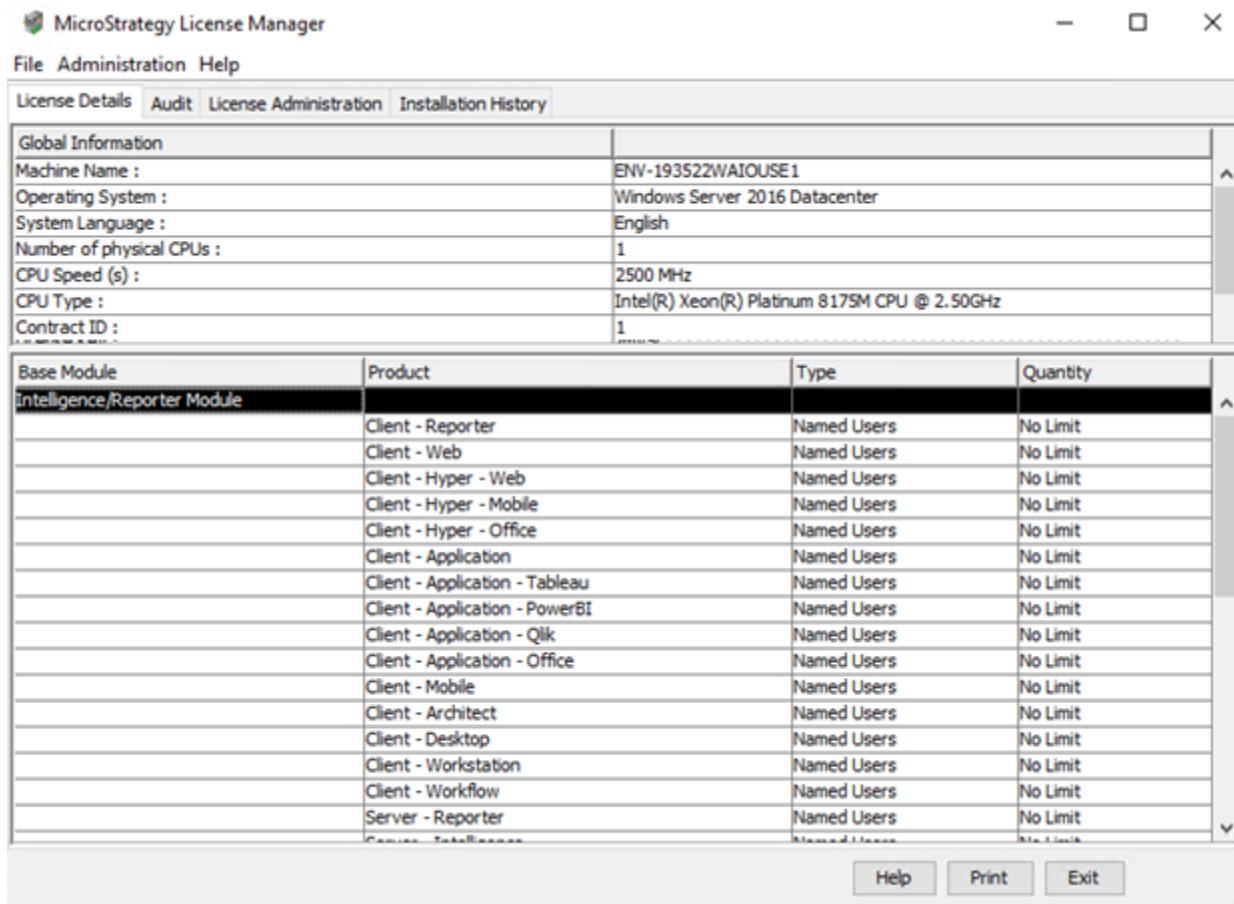
License compliance is an important factor in a successful Intelligent Enterprise. The Platform Administrator should proactively manage the licenses to ensure that MartZon has received all the required MicroStrategy products in accordance with its software agreement while also staying in compliance with the agreement.

Managing your licenses also helps you plan for future license needs based on system usage, while allowing you to take full advantage of the MicroStrategy features and functionality. For example, you might have a CPU-based Intelligence Server license for four CPUs, but only be using two CPUs. An audit of your licenses can alert you to this issue and you can then modify your setup so that you use all four of your licensed CPUs.

License Manager

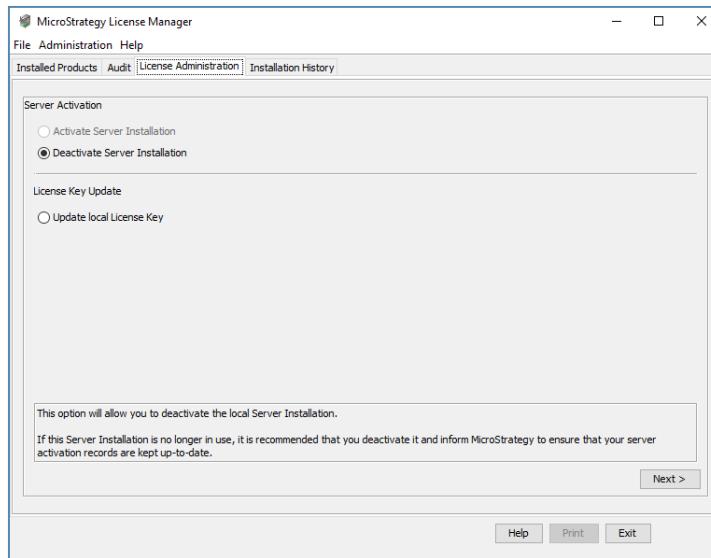
As the Platform Administrator, you can use License Manager for managing your licenses, staying in compliance with your software agreement, and planning for future licensing needs based on your system usage. This tool enables you to:

- View system information and the version, expiration date, and edition of the MicroStrategy products installed on the machine.

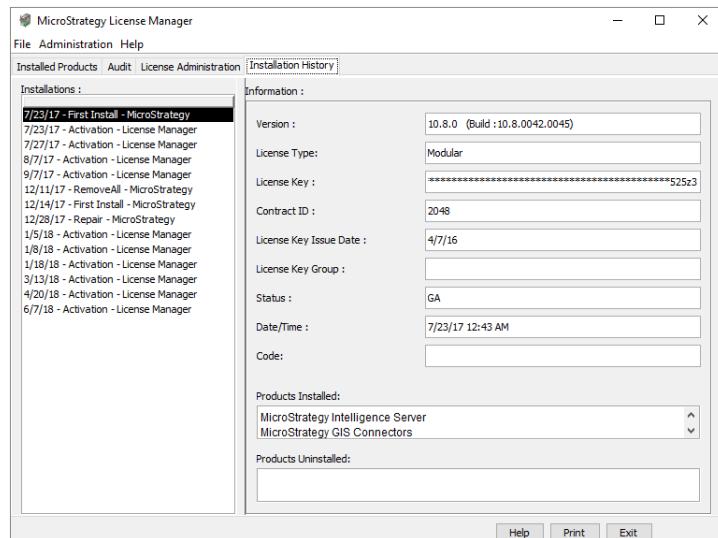


- View the number of product licenses in use by a specified user group. You can also display the enabled or disabled licenses used by the particular user group for the selected products. From this information, you can determine whether you have the required number of licenses. You can also print a report or create and view an HTML file with this information.
- Activate, deactivate, and update server installations. For example:
 - If the products already installed on the machine are also licensed in the new key, you can upgrade license keys using MicroStrategy License Manager.
 - If the new key doesn't provide licenses for products already installed on the machine, or if you need to install additional products using the new key, you must do this using the MicroStrategy Installer.

You can also update license keys to use more processors or install MicroStrategy products on a faster processor machine.



- View details of MicroStrategy installations and uninstallations on a machine on the Installations History tab, including the information applied hot fixes.



The Installations History tab shows:

- First Install
- Upgrade Install
- Repair
- Removal

- Uninstall (Remove All)
- Activation
- Deactivation

Managing and verifying licenses

As the Platform Administrator, you need to ensure that you are managing your licenses effectively based on the two main types of licenses:

- Named User licenses — the number of users with access to specific functionality are restricted
- CPU licenses — the number and speed of the CPUs used by MicroStrategy server products are restricted



Refer to your MicroStrategy contract and any accompanying contract documentation for descriptions of your MicroStrategy license types.

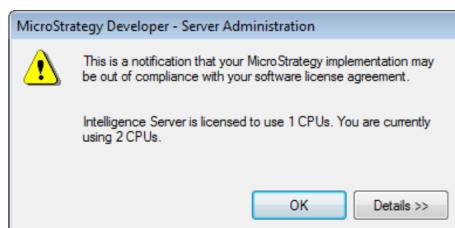
You can use License Manager to ensure that your system is in compliance with your licenses. You can check for and manage the following licensing issues:

- More copies of a MicroStrategy product are installed and used than the number of licenses.
- More users are using the system than the number of licenses.
- More CPUs are being used with Intelligence Server (or MicroStrategy Web or Mobile Server) than the number of licenses.

The following sections discuss the two types of licenses in more detail.

Managing CPU licenses

When you purchase CPU licenses, the Intelligence Server monitors the number of CPUs being used in your implementation and compares it to the number of licenses that you have. Exceeding the licenses will cause you to be out of compliance with your license agreement.



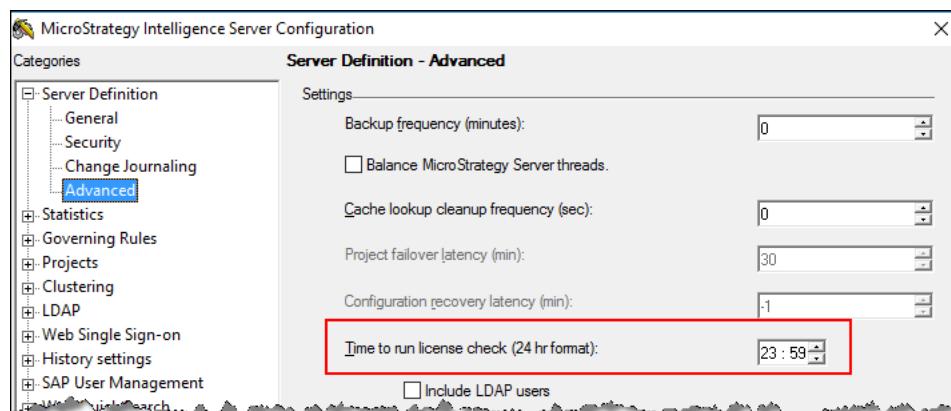
For example, you may have MicroStrategy Web installed on two dual-processor machines (four CPUs) while your license is for only two CPUs, causing you to be out-of-compliance with your software agreement.

To resolve this problem, you can either use License Manager to reduce the number of CPUs being used on a given machine, or you can obtain additional licenses from MicroStrategy.

To verify your CPU licenses, Intelligence Server automatically scans the network to count the number of CPUs in use by all Intelligence Servers in the network (subnet). If the number of CPU licenses has been exceeded, an error message is displayed when a user logs in to a MicroStrategy product.

By default, CPU license compliance checking is done automatically at the following times:

- Intelligence Server start up (this constitutes a network check)
- Intelligence Server shut down (this constitutes a network check)
- Intelligence Server joins a cluster (this constitutes a cluster check)
- Intelligence Server leaves a cluster (this constitutes a cluster check)
- When you manually check using License Manager (this constitutes as cluster and network checks)
- Once-a-day check as per the Time to run license check setting in Intelligence Server Configuration Editor (default time is 23:59) (this constitutes as cluster and network checks)



Managing Named User licenses

In a Named User licensing scheme, the privileges given to users and groups determine what licenses are assigned to users and groups. Intelligence Server monitors the number of users in your MicroStrategy environment with each privilege, and compares that to the number of available licenses.

For example, the **Web use filter editor** privilege is a **Web Professional** privilege. If you assign this privilege to User1, then Intelligence Server grants a Web Professional license to User1. If you only have one Web Professional license in your environment and you assign any other Web Professional privilege, such as **Web use prompt editor**, to another user, you will be out of compliance with your software agreement.

 The **administrator** user that is created with the metadata repository is not considered in the licensed user count. Similarly, users without any product-based privileges (such as Web Analyst or Web Reporter) are listed in License Manager in the group **Users without license association**, and are not counted against any MicroStrategy licenses.

To fix the out of compliance problem, you can either change the user privileges to match the number of licenses you have, or you can obtain additional licenses from MicroStrategy. License Manager can determine which users are causing the metadata to exceed your licenses and which privileges for those users are causing each user to be classified as a particular license type.

License files

The MicroStrategy platform stores licensing-related information (such as the license key) primarily in the following three files: mstr.hist, mstr.bhv, and activate.xml. These files are generated during the installation, modification, or upgrade of the MicroStrategy platform, and are accessed at various times during the operation and maintenance of the MicroStrategy platform. For example, the files are generated during the license key update using License Manager and Intelligence Server startup. You will get errors if any of these files are corrupted or damaged or if License Manager does not have read and write permissions on these files.

 The user account running must have administrative privileges to allow License Manager to modify the three license-related files related.

- **mstr.hist**—This file contains the history of MicroStrategy installations (including any repairs) and uninstallations, and is used for Intelligence Server

activation. It also contains information such as the license key and the details of the license updates.

```
[General]
LatestIndex=15
LatestQRAIndex=8
LatestLicenseManagerIndex=15
[Install1]
ContractId=2048
License Key Group=31
LicenseKeyIssueDate=2016-4-7
Build=10.8.0042.0045
Release=10.8.0
CDKey=B6B5wWzvxt52FvW8xfbBb
Source=MicroStrategy
Type=First Install
Time=12:43:43
Date=2018-07-23
State=GA
LicenseType=Modular
Install=MicroStrategy Intelligence Server
Install=MicroStrategy GIS Connectors
Install=MicroStrategy Portlets
Install=MicroStrategy Web Services (J2EE)
```

By default, mstr.hist is located at:

Windows:

C:\Program Files (x86)\Common Files\MicroStrategy\

UNIX/Linux:

<MSTR_LOG_PATH> which is the directory specified as the Log directory during MicroStrategy installation

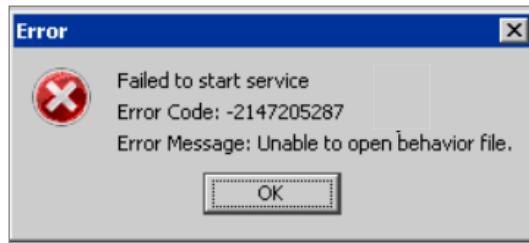
The MicroStrategy platform may not function properly if this file is corrupted or damaged. The following image shows a sample error message that displays when trying to activate Intelligence Server on a machine with damaged mstr.hist file:



Depending on your environment, you can restore the mstr.hist file by performing a repair or a fresh install of the MicroStrategy platform.

- **mstr.bhv**—This behavior file encodes the information from the license key in a binary format and determines the behavior of the various MicroStrategy components on the machine.

A damaged behavior file can cause errors when the administrator uses License Manager to apply a new license key. If the file is damaged, the components will not function correctly. For example, it can prevent Intelligence Server from starting, result in an error message as shown in the following image:



By default, mstr.bhv is located at:

Windows:

C:\Program Files (x86)\Common Files\MicroStrategy\

UNIX/Linux:

<MSTR_HOME_PATH> which is the directory specified as the Home directory during MicroStrategy installation



You can restore the mstr.bhv file from a valid mstr.bhv backup file, or performing a repair or a fresh install of the MicroStrategy platform.

- **activate.xml**—This activation XML file contains information about your MicroStrategy installation. It is automatically generated during installation, modification, or upgrade of Intelligence Server. During server activation, this XML file is uploaded to MicroStrategy either automatically by the installation routine or through License Manager or by manually uploading this activation file through a web browser via the secure web site, <https://licensing.microstrategy.com>.

By default, activate.xml is located at:

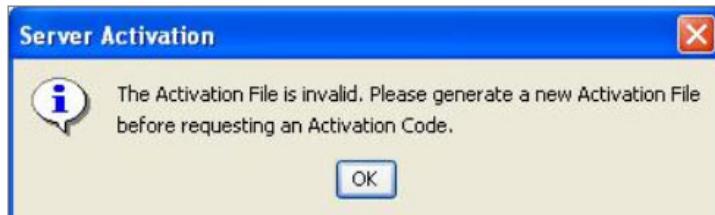
Windows:

C:\Program Files (x86)\Common Files\MicroStrategy\

UNIX/Linux:

<MSTR_HOME_PATH> which is the directory specified as the Home directory during MicroStrategy installation

If the file is damaged, you will receive an error and will not be able to activate your server installation



 You can use License Manager to regenerate the Activate.xml file.

Updating processor capacity: Using Service Manager

When using CPU licenses, you must provide the number of processors to be used by Intelligence Server on that machine as part of the installation process. If deploying Intelligence Server on a machine that has more CPUs than licensed, the platform administration team will need to update the processor capacity using CPU or processor affinity.

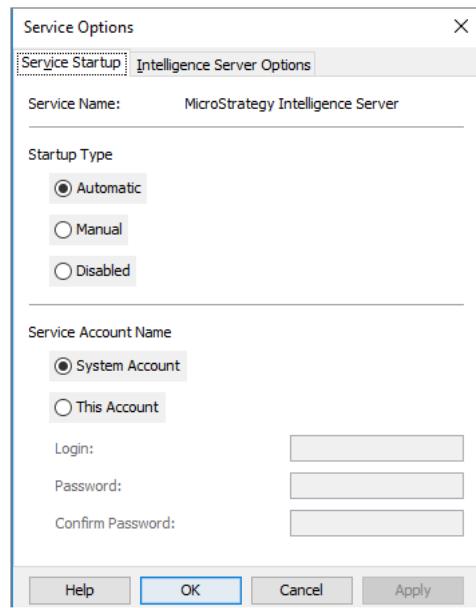
CPU affinity is the ability to deploy Intelligence Server on a specific, selected CPUs (a subset of the total number of physical CPUs) on a given machine. You can apply CPU affinity, which is stored as a binary bit mask, using MicroStrategy Service Manager. This tool enables you to configure and manage the services running on the Intelligence Server machine. It can also limit the number of CPUs used by the Intelligence Server.

 The steps below are not to be performed as an in-class exercise. They are for reference only.

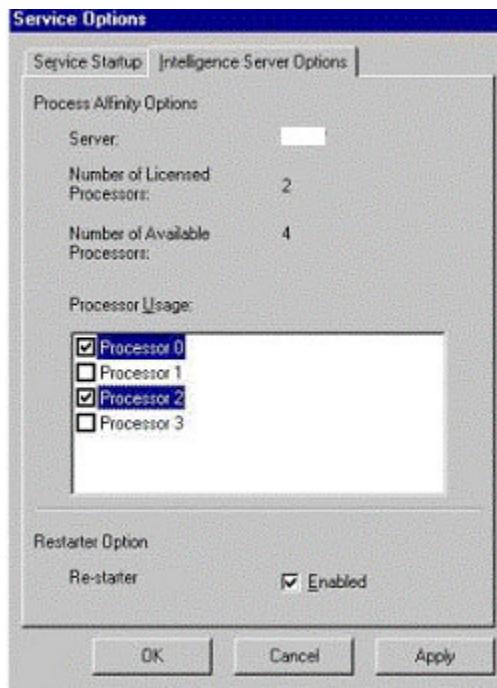
To change CPU affinity settings in Service Manager:

- 1 On the Intelligence Server machine, launch **Service Manager**.
- 2 From the **Service** drop-down list, select **MicroStrategy Intelligence Server**.

- 3 Click **Options**. Then, in the Service Options window, select the **Intelligence Server Options** tab.



- 4 In the Processor Usage area, select which processors Intelligence Server should use. In the example image below, the machine has four CPUs while the license only allows two CPUs. The administrator can specify which two CPUs Intelligence Server should use:



When you are finished, click **OK**. The Service Options window closes and the CPU affinity is updated.

License Manager best practices

Best Practice

- 1** You should manually trigger a license check using License Manager when:
 - Updating the compliance status of Intelligence Server, especially if you have Named User (NU) license as the NU compliance status is not updated by an Intelligence Server restart. For CPU compliance status, an Intelligence Server restart is sufficient
 - Obtaining License Manager Report before sending it to Tech Support for an out-of-compliance case, otherwise the information in the report will be inconsistent
 - Performing a compliance check of your licenses
 - The automated check has not been occurring. This usually happens when the Intelligence Server is shut down every night and is not running at 23:59, which is the default time automated compliance check time.
- 2** If using CPU licenses, you need to ensure that the machine specifications are compatible with your license agreement. CPU-based license key includes specifications for the clock speed which is checked at installation time. It is a hard stop—if the machine has more clock speed than what the license key provides, the installation will not proceed.

Exercise 6.4: Perform a License Manager audit

In this exercise, you will perform a License Manager audit on the Windows machine to display the number of licenses assigned for each product in the Everyone group.

You can access License Manager on the Windows machine by clicking the **Start** button, point to **MicroStrategy Tools**, and select **License Manager**.

The login credentials and other information you will need for accessing your environment are included in the MicroStrategy Cloud email.

Based on your audit, how many active licenses are associated with Architect and Web?

- Number of active Client Architect licenses_____
- Number of active Client Web licenses_____

Exercise 6.4 Solutions: Perform a License Manager audit

In this exercise, you will perform a License Manager audit on the Windows machine to display the number of licenses assigned for each product in the Everyone group.

You can access License Manager on the Windows machine by clicking the **Start** button, point to **MicroStrategy Tools**, and select **License Manager**.

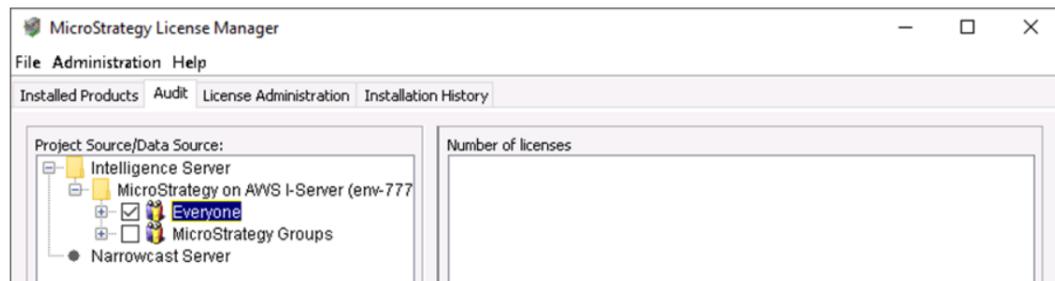
The login credentials and other information you will need for accessing your environment are included in the *MicroStrategy Cloud email*.

Based on your audit, how many active licenses are associated with Architect and Web?

- Number of active Client Architect licenses_____
- Number of active Client Web licenses_____

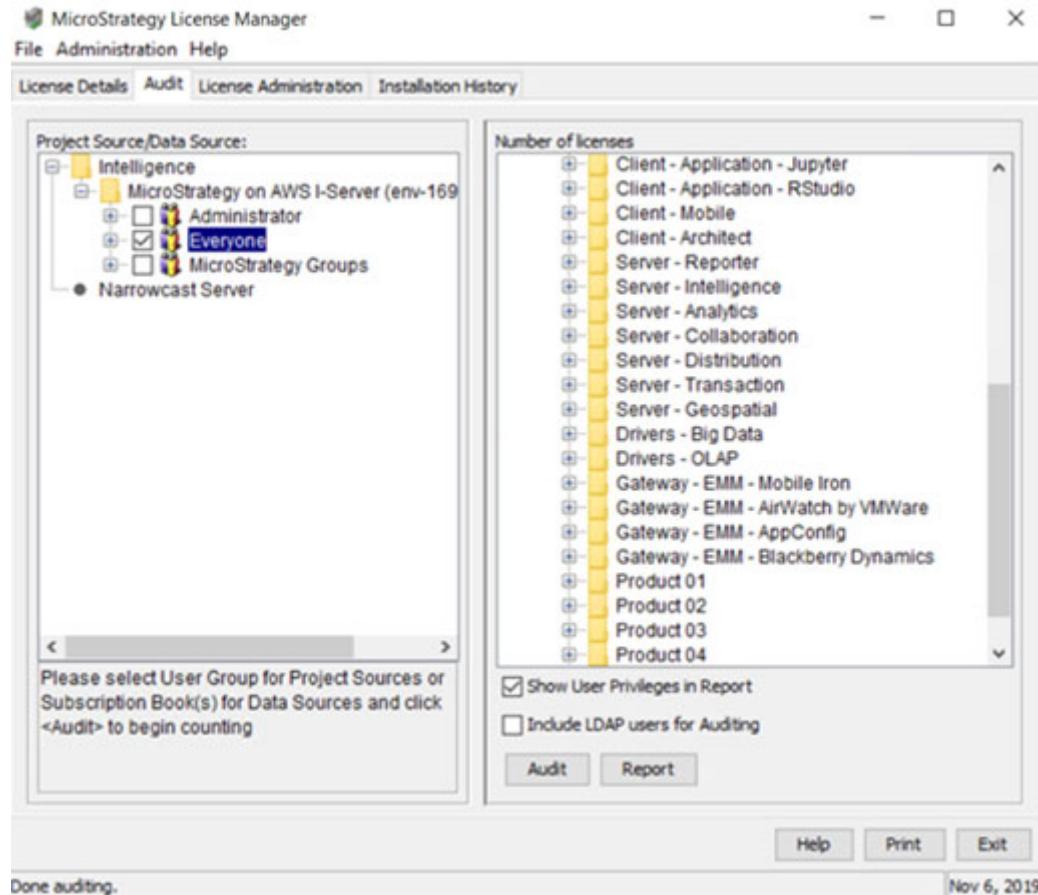
Access License Manager on the Windows machine

- 1 On the Windows machine, click **Start**, point to **MicroStrategy Tools**, and select **License Manager**.
- 2 When prompted, in the User Account Control window, click **Yes** to open License Manager.
- 3 On the Audit tab of MicroStrategy License Manager, under Intelligence Server, click the **MicroStrategy on AWS I-Server** project source, and in the **Login id** and **Password** boxes, type the login credentials listed in the MicroStrategy Cloud email. Click **Connect**, then click **OK** in the message window regarding establishing the connection.
- 4 Select the **Everyone** group.



5 Select Show User Privileges in Report. Then, click Audit.

In the Number of licenses pane, you can view the number of licenses (active and disabled) associated with each product.



6 How many active licenses are associated with Architect and Web?

- Number of active Client - Architect licenses: **22**
- Number of active Client - Web licenses: **22**

 The number of licenses in your environment may be different than that shown in the preceding image.

Exercise 6.5: Generate a License Manager report

You want to ensure that MartZon received all MicroStrategy products in accordance with its licensing agreement.

In this exercise, you will use the License Manager on the Linux machine to generate a report containing information about your MicroStrategy installation.

You can access License Manager on the Linux machine using Putty (or VNC) and then navigate to the **/opt/mstr/MicroStrategy/bin** directory and launch **mstrlicmgr**.

After generating the report, answer the specified questions about your installation.

1 Intelligence Server machine

- Number of CPUs _____
- Processor speed _____
- Type of CPU (processor) _____
- Operating system type and version _____

2 What MicroStrategy product types are installed on the Linux machine? _____

3 How many users of the following product types does your license key provide?

- Server - Intelligence _____
- Server - Geospatial _____
- Server - Collaboration _____
- Drivers - Big Data _____
- Client - Architect _____
- Client - Web _____
- Client - Mobile _____

- Client - Hyper - Voice_____
- 4** How many users have you set for those product types?_____

Exercise 6.5 Solutions: Generate a License Manager report

You want to ensure that MartZon received all MicroStrategy products in accordance with its licensing agreement.

In this exercise, you will use the License Manager on the Linux machine in your environment to generate a report containing information about your MicroStrategy installation.

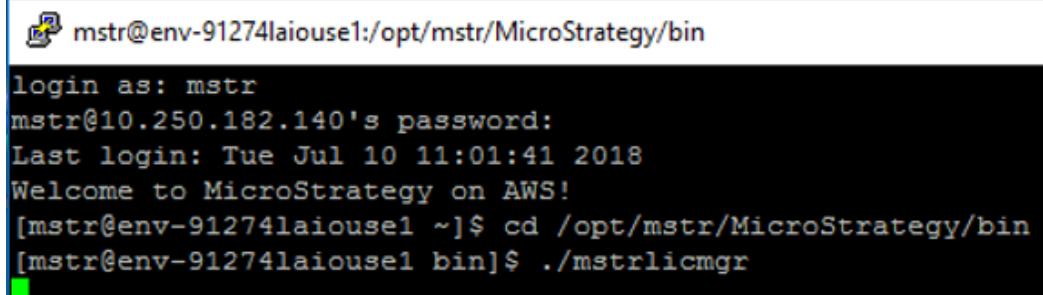
You can access License Manager on the Linux machine using Putty (or VNC) and then navigate to the **/opt/mstr/MicroStrategy/bin** directory and launch **mstrlicmgr**.

After generating the report, answer the specified questions about your installation.

Launch License Manager from the bin subdirectory

As you had already changed the directory path to the bin subdirectory in Putty as part of an earlier exercise, you can launch License Manager by typing the command for it.

- 1 Maximize the Putty window.
- 2 On the console, type **./mstrlicmgr**. License Manager opens.



```
mstr@env-91274laiouse1:/opt/mstr/MicroStrategy/bin
login as: mstr
mstr@10.250.182.140's password:
Last login: Tue Jul 10 11:01:41 2018
Welcome to MicroStrategy on AWS!
[mstr@env-91274laiouse1 ~]$ cd /opt/mstr/MicroStrategy/bin
[mstr@env-91274laiouse1 bin]$ ./mstrlicmgr
```



You may need to maximize the Xming window to see the License Manager window.

- 3 When prompted, in the User Account Control window, click **Yes** to open License Manager.
- 4 Select the **Audit** tab.

- 5 In the Project Source/Data Source pane, expand **Intelligence**, then click the **mstr_metadata** project source.
- 6 In the **Login id** and **Password** boxes, type the login credentials listed in the MicroStrategy Cloud email.
- 7 Click **Connect** and then click **OK** in the message window regarding establishing the connection.
- 8 Select the **Everyone** group.
- 9 In the right pane, select **Show User Privileges in Report**, then click **Report**.
- 10 In the Save window, type **LicMgrRpt** as the file name and click **Save**.

The report is generated.

Micro Strategy License Manager Report																																																						
<u>Local Machine Information</u>																																																						
<u>System Information</u>																																																						
<table><tr><td>Machine Name</td><td>env-98165iaious1</td></tr><tr><td>CPU Speed</td><td>2699 MHz</td></tr><tr><td>CPU Type(s)</td><td>Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz</td></tr><tr><td>No. of physical CPUs</td><td>1</td></tr><tr><td>Operating System</td><td>Linux 3.10.0-862.el7.x86_64</td></tr><tr><td>OS Service Pack</td><td>Please refer to the install.log file at /opt/mstr/MicroStrategy/install</td></tr><tr><td>System Language</td><td>English</td></tr></table>					Machine Name	env-98165iaious1	CPU Speed	2699 MHz	CPU Type(s)	Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz	No. of physical CPUs	1	Operating System	Linux 3.10.0-862.el7.x86_64	OS Service Pack	Please refer to the install.log file at /opt/mstr/MicroStrategy/install	System Language	English																																				
Machine Name	env-98165iaious1																																																					
CPU Speed	2699 MHz																																																					
CPU Type(s)	Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz																																																					
No. of physical CPUs	1																																																					
Operating System	Linux 3.10.0-862.el7.x86_64																																																					
OS Service Pack	Please refer to the install.log file at /opt/mstr/MicroStrategy/install																																																					
System Language	English																																																					
<u>License Key Information</u>																																																						
<table><tr><td>Contract ID</td><td>1</td></tr><tr><td>License Key</td><td>VKh3B*****52WzY</td></tr><tr><td>License Key Group</td><td>1</td></tr><tr><td>License Key Issue Date</td><td>3/8/2016</td></tr><tr><td>License Type</td><td>Modular</td></tr></table>					Contract ID	1	License Key	VKh3B*****52WzY	License Key Group	1	License Key Issue Date	3/8/2016	License Type	Modular																																								
Contract ID	1																																																					
License Key	VKh3B*****52WzY																																																					
License Key Group	1																																																					
License Key Issue Date	3/8/2016																																																					
License Type	Modular																																																					
<u>Software Activation Information</u>																																																						
<table><tr><td>Activation State</td><td>Not Applicable</td></tr><tr><td>Host ID</td><td>bae92d318b8fec0</td></tr></table>					Activation State	Not Applicable	Host ID	bae92d318b8fec0																																														
Activation State	Not Applicable																																																					
Host ID	bae92d318b8fec0																																																					
<u>Installed Products</u>																																																						
<table><thead><tr><th>Products</th><th>Edition</th><th>Version</th><th>Expiration</th><th>Evaluation Period</th></tr></thead><tbody><tr><td>Intelligence Server Module</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>OLAP Services Option</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>Report Services Option</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>Distribution Services Option</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>Transaction Services Option</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>MultiSource Option</td><td>Universal</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr><tr><td>System Manager</td><td>-</td><td>10.11.0051.0056 (10.11.0)</td><td>Not Applicable</td><td>-</td></tr><tr><td>Command Manager</td><td>-</td><td>10.11.0051.0056 (10.11.0)</td><td>Not Applicable</td><td>-</td></tr><tr><td>Command Manager (OFM Edition)</td><td>-</td><td>10.11.0051.0056 (10.11.0)</td><td>-</td><td>-</td></tr></tbody></table>					Products	Edition	Version	Expiration	Evaluation Period	Intelligence Server Module	Universal	10.11.0051.0056 (10.11.0)	-	-	OLAP Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-	Report Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-	Distribution Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-	Transaction Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-	MultiSource Option	Universal	10.11.0051.0056 (10.11.0)	-	-	System Manager	-	10.11.0051.0056 (10.11.0)	Not Applicable	-	Command Manager	-	10.11.0051.0056 (10.11.0)	Not Applicable	-	Command Manager (OFM Edition)	-	10.11.0051.0056 (10.11.0)	-	-
Products	Edition	Version	Expiration	Evaluation Period																																																		
Intelligence Server Module	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
OLAP Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
Report Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
Distribution Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
Transaction Services Option	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
MultiSource Option	Universal	10.11.0051.0056 (10.11.0)	-	-																																																		
System Manager	-	10.11.0051.0056 (10.11.0)	Not Applicable	-																																																		
Command Manager	-	10.11.0051.0056 (10.11.0)	Not Applicable	-																																																		
Command Manager (OFM Edition)	-	10.11.0051.0056 (10.11.0)	-	-																																																		

Based on this report, answer the following questions:

1 Intelligence Server machine

- ❑ Number of CPUs _____
- ❑ Processor speed _____

- Type of CPU (processor) _____
- Operating system type and version _____

Solution: Information provided in the **System Information** section of the report.

- 2 What MicroStrategy product types are installed on the Linux machine?_____

Solution: Information provided in the **Installed Products** section of the report.

- 3 How many users of the following product types does your license key provide?

- Server - Intelligence_____
- Server - Geospatial_____
- Server - Collaboration_____
- Drivers - Big Data_____
- Client - Architect_____
- Client - Web_____
- Client - Mobile_____
- Client - Hyper - Voice_____

Solution: Information provided in the **Audit Summary** section of the report.

- 4 How many users have you set for those product types?_____

Solution: Information provided in the **Auditing Summary** section of the report.

Exercise 6.6: Configure the license check time

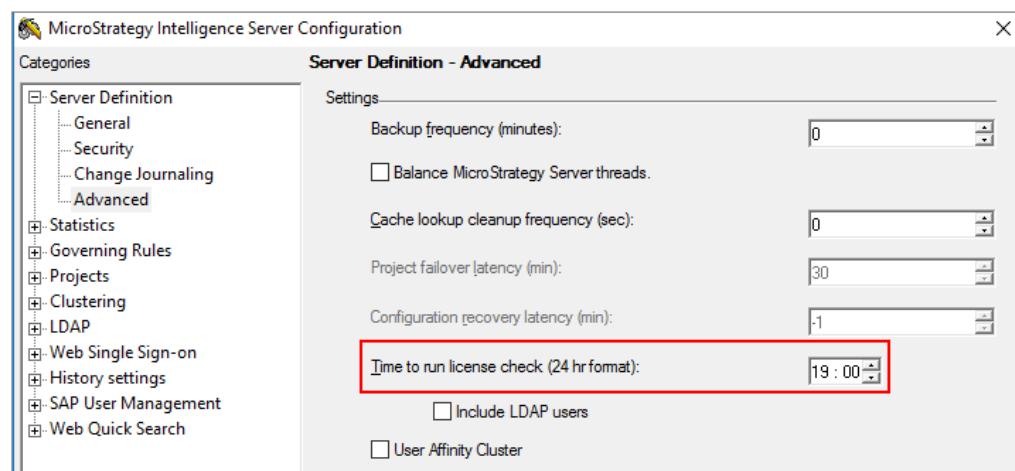
To verify the Named User licenses, Intelligence Server scans the metadata repository daily for the number of users fitting each Named User license type. If the number of licenses for a given type has been exceeded, an error message is displayed when a user logs in to a MicroStrategy product.

In this exercise, you will modify the time when Intelligence Server scans the metadata repository daily for the number of users fitting each Named User license type. The default check time is 23:59; you will change it to 19:00 License Manager—the time when the system is not in use.

Access Developer to configure the license check time

You can configure license check time using Developer on your Windows machine.

- 1 In Developer, right-click **MicroStrategy on AWS I-Server** and select **Configure MicroStrategy Intelligence Server**.
- 2 In the MicroStrategy Intelligence Server Configuration Editor, on the left under Server Definition, select **Advanced**.
- 3 In the Time to run license check (24 hr format) box, change the time to **19:00** and click **OK**.



Monitoring product usage: Compliance Telemetry Dossier

In addition to License Manager, the Compliance Telemetry Dossier in Platform Analytics provides you an easy way to monitor system and product usage to ensure that your organization remains in compliance with licensing agreements.

Platform Analytics has the added benefit of analyzing license telemetry using the power of MicroStrategy dossiers. Additionally, Platform Analytics can help you analyze compliance results across multiple environments.

The Compliance Telemetry Dossier can be accessed from the Platform Analytics project. This dossier is also embedded in the License area of the MicroStrategy Workstation.

Exercise 6.7: Use the Compliance Telemetry Dossier

The Compliance Telemetry dossier in Platform Analytics provides MartZon administrators with insight into what products and quantities are being used based on privilege checks, as well as how existing privileges map to MicroStrategy products packaging.

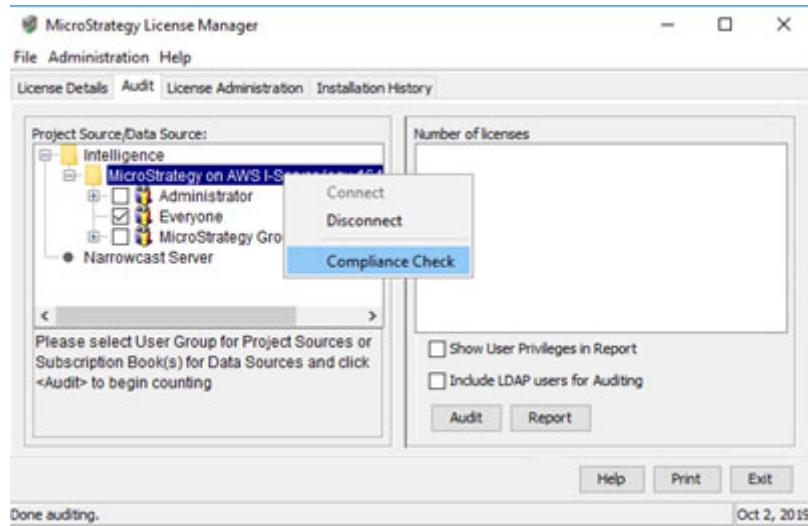
Intelligence Server performs an automatic Named User compliance check once a day according to the schedule set in the server configuration. When this schedule is triggered, it updates the data in Platform Analytics. However, since your cloud environment is typically provisioned on the same day as your class, this schedule may not have triggered yet. To ensure that Platform Analytics contains compliance telemetry data, you need to trigger the compliance check using License Manager.

In this exercise, first trigger the Named User compliance check in License Manager, then log in to Platform Analytics to view the Compliance Telemetry Dossier.

Manually trigger the compliance check using License Manager

- 1 On the remote Windows machine of your cloud environment, click Windows **Start** button, point to **MicroStrategy Tools**, and select **License Manager**.
- 2 When prompted, in the User Account Control window, click **Yes**.
- 3 Select the **Audit** tab.
- 4 From the left pane, expand the **Intelligence** folder.
- 5 Expand **MicroSmstrategy on AWS I-Server** project source.
- 6 In the **Login id** and **Password** boxes, type the login credentials listed in the MicroStrategy Cloud email.
- 7 Click **Connect** and then click **OK** in the message window regarding establishing the connection.
- 8 Select **Everyone**, if not already selected.

9 Right-click **MicroStrategy on AWS I-Server**, then select **Compliance Check**.



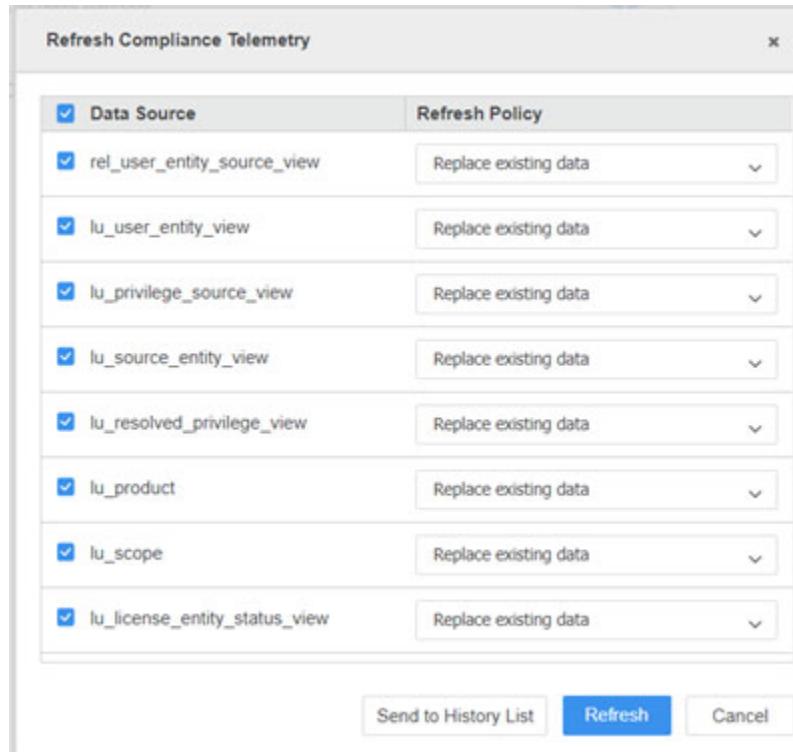
10 Click **OK**, and exit License Manager.

Access the Compliance Telemetry Dossier in MicroStrategy Web

Explore the Compliance Telemetry Dossier in Platform Analytics using MicroStrategy Web.

- 1 Log in to MicroStrategy Web, and select the **Platform Analytics** project.
- 2 Navigate to the **Shared Reports\ 2.Utilites** folder.

3 Right-click the **Compliance Telemetry** cube and select **Republish**.



4 Click **Refresh** to update the Platform Analytics Cube.

5 Click **Done**, after the cube has been published.

6 Navigate to the **Shared Reports\ 1. Dossiers** folder.

7 Open the **Compliance Telemetry** dossier.

License auditing with the Compliance Telemetry dossier

The Compliance Telemetry dossier contains the following pages:

- **License Overview**—Provides a summary of environment, account, and product and license information. Pre-formatted thresholds applied to the Compliance column highlight out-of-compliance usage. Reporter and

Intelligence are represented in independent areas to help quickly pinpoint issues along with the number of accounts enabled for the current license.

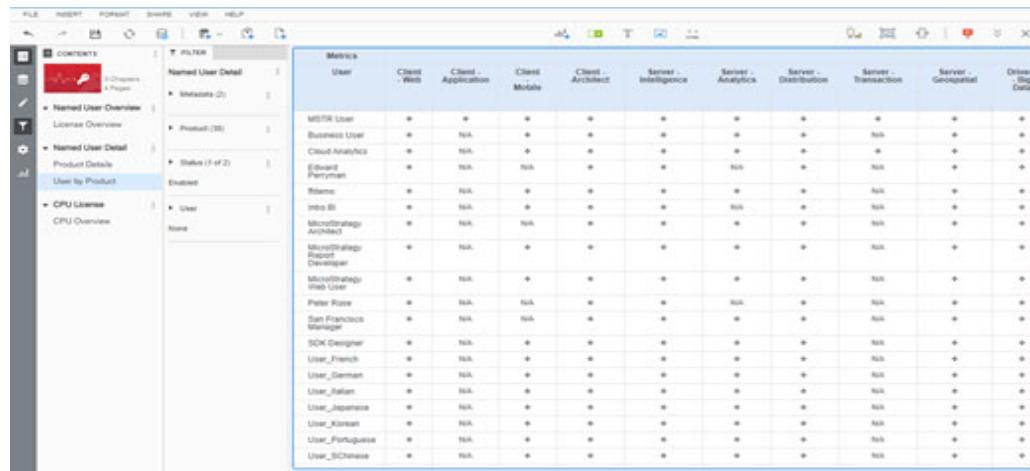
Product	Licenses Purchased	Licenses Available	Licenses Used	Compliance Status
Client - Reporter	Unlimited	Unlimited	0	Unlimited
Client - Web	Unlimited	Unlimited	22	Unlimited
Client - Application	Unlimited	Unlimited	1	Unlimited
Client - Mobile	Unlimited	Unlimited	18	Unlimited
Client - Architect	Unlimited	Unlimited	22	Unlimited
Server - Reporter	Unlimited	Unlimited	0	Unlimited
Server - Intelligence	Unlimited	Unlimited	26	Unlimited
Server - Analytics	Unlimited	Unlimited	19	Unlimited
Server - Collaboration	Unlimited	Unlimited	22	Unlimited
Server - Distribution	Unlimited	Unlimited	26	Unlimited
Server - Transaction	Unlimited	Unlimited	2	Unlimited
Server - Geospatial	Unlimited	26	26	0
Client - Hyper - Web	Unlimited	Unlimited	0	Unlimited
Drivers - Big Data	Unlimited	26	26	0
Drivers - OLAP	Unlimited	26	26	0
Gateway - EMM - Mobile Iron	Unlimited	18	18	0

- **Product Details**—Provides more in-depth analysis of license usage at the product level, as well as detailed information on each user and their associated privileges.

Product	Licenses Used	Licensed Users
Client - Reporter	0	0
Client - Web	22	22
Client - Application	1	1
Client - Mobile	18	18
Client - Architect	22	22
Server - Reporter	0	0
Server - Intelligence	26	26
Server - Analytics	19	19
Server - Collaboration	22	22

User Details		Privileges			
Product	User	User	Privilege	Product	Source
Client - Web	MSTR User	MSTR User	No privilege exposed	Server - Geospatial	MSTR User
	Business User			Drivers - Big Data	MSTR User
	Cloud Analytics			Drivers - OLAP	MSTR User
	Edward Perryman			Gateway - EMM - Mobile Iron	MSTR User
	f1demo			Gateway - EMM - AppWatch by VMWare	MSTR User
	Info BI			Gateway - EMM - AppConfig	MSTR User
	MicroStrategy Architect			Gateway - EMM - BlackBerry Dynamics	MSTR User
	MicroStrategy Report Developer			Create Application	MSTR User
	MicroStrategy Web User			Server - Intelligence	MSTR User
	Peter Rose				
	San Francisco Manager				
	SDK Designer				
	User_French				

- **User by Product**—The User by Product page consists of a Product-Privilege matrix in reference to the current MicroStrategy Product Packaging. Each client license requires a corresponding server license, so a client privilege automatically consumes a server license, with the exception of Reporter privileges. The privileges associated with the Reporter product are listed in their own column in the matrix.



The screenshot shows the 'User by Product' matrix in the MicroStrategy Platform Administrator. The matrix has 'Metrics' as the first column and 'User' as the second column. The rows list various users, and the columns represent different product packages: Client - Web, Client - Application, Client - Mobile, Client - Architect, Server - Intelligence, Server - Analytics, Server - Distribution, Server - Transaction, Server - Geospatial, and Drivers - Big Data. Most users have a 'Client - Web' license, while some like 'Business User' and 'Cloud Analytics' have 'Server - Distribution' licenses. The 'Server - Transaction' column is mostly empty except for a few users like 'Edward Johnson' and 'Ritamay'. The 'Drivers - Big Data' column also contains mostly empty cells with a few '+' symbols.

Metrics	User	Client - Web	Client - Application	Client - Mobile	Client - Architect	Server - Intelligence	Server - Analytics	Server - Distribution	Server - Transaction	Server - Geospatial	Drivers - Big Data
	MSTR User	+	+	+	+	+	+	+	+	+	+
	Business User	N/A	+	+	+	+	+	+	N/A	+	+
	Cloud Analytics	N/A	+	+	+	+	+	+	+	+	+
	Edward Johnson	N/A	N/A	+	+	N/A	+	N/A	N/A	+	+
	Ritamay	N/A	+	+	+	+	+	N/A	N/A	+	+
	Intro BI	N/A	N/A	+	+	N/A	+	N/A	N/A	+	+
	MicroStrategy Architect	N/A	N/A	+	+	+	+	+	N/A	+	+
	MicroStrategy Report Developer	N/A	+	+	+	+	+	N/A	N/A	+	+
	MicroStrategy Sales User	N/A	+	+	+	+	+	+	N/A	+	+
	Peter Rose	N/A	N/A	+	+	N/A	+	N/A	N/A	+	+
	San Francisco Manager	N/A	N/A	N/A	+	+	+	N/A	N/A	+	+
	SDK Designer	N/A	+	+	+	+	+	N/A	+	+	+
	User_French	N/A	+	+	+	+	+	N/A	+	+	+
	User_German	N/A	+	+	+	+	+	N/A	+	+	+
	User_Italian	N/A	+	+	+	+	+	N/A	+	+	+
	User_Japanese	N/A	+	+	+	+	+	N/A	+	+	+
	User_Korean	N/A	+	+	+	+	+	N/A	+	+	+
	User_Portuguese	N/A	+	+	+	+	+	N/A	+	+	+
	User_Schinese	N/A	+	+	+	+	+	N/A	+	+	+

- **CPU License**—The CPU License page lists the number of CPUs assigned to a purchased license.

An environment can be configured to use either a CPU-based license or a Named User-based license. Since your cloud environment has been configured for Named User-based license during installation, this page does not show any data.

TROUBLESHOOTING

As the Platform Administrator, you need to provide technical leadership and support to guide your platform administration team in troubleshooting the analytics environment issues. Your team can use various logs that are generated during the course of environment operation for troubleshooting issues.

In this chapter, we will review:

- Supporting the analytics environments: troubleshooting
- Resolving issues: tools for troubleshooting

Supporting the analytics environments: troubleshooting

Given the importance of analytics environments in day-to-day business operations, Intelligent Enterprises should focus on supporting analytics environments and creating better user support experiences. As such, the Platform Administrator should create guidelines for troubleshooting MicroStrategy Web server, Intelligence Server, and other components in an analytics environment. By creating troubleshooting guidelines recorded in a troubleshooting guide, the

Platform Administrator provides a consistent methodology to zero in on root causes and search for resolution options.

Creating troubleshooting guidelines

A common troubleshooting process is outlined below:

- 1 Gather detailed steps to reproduce the issue. Steps to reproduce should be specific step-by-step instructions to see the issue.
- 2 Perform each step to confirm if the issue can be reproduced.
- 3 Record the steps that reproduce the issue 100% of the time or note if the issue is intermittent. Every time an option is clicked, a folder is navigated, or a task is performed, then a step should be noted.
- 4 In case of MicroStrategy Web or MicroStrategy Mobile-related issues, confirm if the issue occurs in the four-tier environment only or does it also occur in a three-tier environment.
 If an issue happens only in the four-tier environment then Intelligence Server is not likely to be the source of problem.
- 5 Collect the appropriate log files (such as DSSErrors.log in case of the Intelligence Server-related issues and web server logs in case of the MicroStrategy Web server-related issues) if further troubleshooting is necessary, and review for underlying errors when the issue is reproduced.
- 6 Collaborate with the System Administrator and Intelligence Center architects to resolve the issues to ensure an accurate and timely issue resolution. For critical issues such as Intelligence Server abnormal shutdown (crash), or Intelligence Server unresponsiveness (hang), the platform administration team should do the following:
 - Work with System Administrator to bring the Intelligence Server up and running as soon as possible.
 - Log a case with MicroStrategy Technical Support for root cause analysis using a case management system.

For non-critical issues, do the following:

- Log a case with MicroStrategy Technical Support for root cause analysis using a case management system.
- Follow up on the case until a resolution is reached and communicate the resolution to the case owner.

Make sure your troubleshooting guide includes the following:

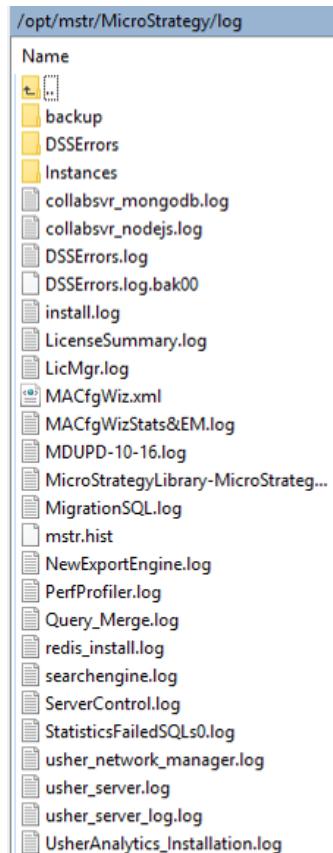
Best Practice

- Reproducing the issue with exact step-by-step instructions is critical in understanding the issue's behavior, which helps to formulate resolution options.
- Your team should request screen shots when applicable. Additionally, they can also help capture, investigate, and back up the server logs using tools such as MicroStrategy Diagnostics Configuration Tool and the operating system file system to ensure an accurate and timely issue resolution.

Using predefined log files

The Platform Administrator should ensure that his team is familiar with the log files that can be used for resolving problems. The MicroStrategy platform provides a number of log files that can be used for troubleshooting. By default, these log files are located in the following location:

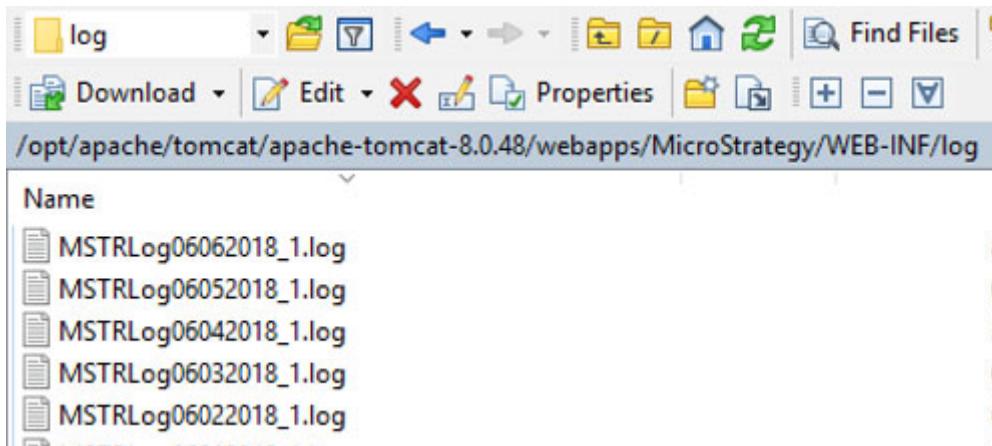
- Windows:** C:\Program Files (x86)\Common Files\MicroStrategy\Log
- Linux/Unix:** /opt/mstr/MicroStrategy/log



A brief explanation of some of the log files that the platform administration team can use for troubleshooting is provided below:

- **install.log**—If the install of the MicroStrategy platform does not complete successfully, you can review the install.log file to identify the reason for any errors.
- **MACfgWizStats&EMg**—If there are any issues related to the use of Configuration Wizard, such as when creating metadata repository or Enterprise Manager statistics repository, you can review the information logged in the MACfgWizStats&EM file to identify the reason for any errors.
- **MAEntMgr.xml**—If the Enterprise Manager statistics data load failed for any reason, you can review the MAEntMgr.xml file to identify the reason for any errors.
- **Web logs**—For any issues related to the communication between the MicroStrategy Web server and the Intelligence Server, you can analyze the web log files generated in the WEB-INF\log subfolder of the MicroStrategy Web installation folder. The log files are located at:
 - **Windows:** C:\Program Files (x86)\MicroStrategy\Web ASPx\WEB-INF\Log
 - **Linux/Unix:** <MicroStrategy Web installation path>/s/webapps/MicroStrategy/WEB-INF/log

The following image shows sample web logs when using MicroStrategy Web Universal with Tomcat on the Linux box of a MicroStrategy on AWS machine.



- **DSErrors.log**—DSErrors.log is one of the most important trace files that should be used when troubleshooting issues related to Intelligence Server.

In addition to the various predefined log files, you can also enable logging within MicroStrategy products such as Command Manager and using tools such as MicroStrategy Diagnostics and Performance Logging tool.

Analyzing log files

All messages in the log files have the same format. Each entry has the following parts:

PID:[thread][date::time][module name][trace type]message

The following table summarize each part of a message:

Section	Definition
PID	Numeric ID of the process that performed the action
thread	Numeric ID of the thread that performed the action
date::time	Date and time at which the action happened
module name	Name of the MicroStrategy component that performed the action
trace type	Type of the log file entry
message	Message about the action

Exercise 7.1: Analyze a log file

In this exercise, you will review the listed messages from a sample log file. These messages were logged upon executing a report named Length of Employment. The report was initially not cached.

After reviewing, explain what is happening in each of the labeled message.

- Message A:

```
286: [THR:480] [06/27/2018::12:24:23:860] [DSS  
ReportServer] [Report Source Tracing]Creating  
Report(Definition) with Flags=0x1000180(OSrcCch  
UptoSrcCch)
```

- Message B:

```
286: [THR:480] [06/27/2018::12:24:23:860] [DSS  
ReportServer] [Report Source Tracing] where  
Definition = Object(Name="Length of Employment"  
Type=3(Report Definition) ID=  
D1AE564911D5C4D04C200E8820504F4F Proj=  
B19DEDCC11D4E0EFC000EB9495D0F44F Ver=  
493C8E3447909F1FBF75C48E11AB7DEB)
```

- Message C:

```
286: [THR:480] [06/27/2018::12:24:24:931] [DSS  
ReportServer] [Report Source Tracing]Created  
ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

- Message D:

```
286: [THR:480] [06/27/2018::12:24:24:931] [DSS  
ReportServer] [Report Source Tracing]Executing  
ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch)) with Actions=
```

```
0x8300003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw  
LclCch UptLclCch), Flags=0x1000180(OSrcCch  
UptoSrcCch)
```

- **Message E:**

```
286:[THR:480][06/27/2018::12:24:25:181][DSS  
ReportServer][Report Source Tracing]Finding in  
cache: ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

- **Message F:**

```
286:[THR:480][06/27/2018::12:24:25:342][DSS  
ReportServer][Report Source Tracing]Not found in  
cache: ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

- **Message G:**

```
286:[THR:314][06/27/2018::12:24:25:432][DSS  
ReportServer][Report Source Tracing]No prompts in  
ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

- **Message H:**

```
286:[THR:492][06/27/2018::12:24:26:634][DSS  
ReportServer][Report Source Tracing]Executing  
ReportInstance(Job=2 Name="Length of Employment"  
ExecFlags=0x1000184(OSrcCch UptoSrcCch) ExecActn=  
0x1000184(ExeSQL RslvCB LclCch)) with Actions=  
0x300003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw
```

LclCch UptLclCch), Flags=0x1000184 (OSrcCch
UptoSrcCch)

Exercise 7.1 Solutions: Analyze a log file

In this exercise, you will review the listed messages from a sample log file. These messages were logged upon executing a report named Length of Employment. The report was initially not cached.

After reviewing, explain what is happening in each of the labeled message.

- Message A:

```
286: [THR:480] [06/27/2018::12:24:23:860] [DSS ReportServer] [Report Source Tracing] Creating Report (Definition) with Flags=0x1000180 (OSrcCch UptoSrcCch)
```

Answer: Intelligence Server creates a report definition.

- Message B:

```
286: [THR:480] [06/27/2018::12:24:23:860] [DSS ReportServer] [Report Source Tracing] where Definition = Object(Name="Length of Employment" Type=3 (Report Definition) ID=D1AE564911D5C4D04C200E8820504F4F Proj=B19DEDCC11D4E0EFC000EB9495D0F44F Ver=493C8E3447909F1FBF75C48E11AB7DEB)
```

Answer: Intelligence Server loads the report definition object named Length of Employment from the metadata.

- Message C:

```
286: [THR:480] [06/27/2018::12:24:24:931] [DSS ReportServer] [Report Source Tracing] Created ReportInstance (Name="Length of Employment" ExecFlags=0x1000180 (OSrcCch UptoSrcCch) ExecActn=0x1000180 (RslvCB LclCch) )
```

Answer: Intelligence Server creates a report instance named Length of Employment.

- Message D:

```
286: [THR:480] [06/27/2018::12:24:24:931] [DSS ReportServer] [Report Source Tracing] Executing ReportInstance (Name="Length of Employment" ExecFlags=0x1000180 (OSrcCch UptoSrcCch) ExecActn=0x1000180 (RslvCB LclCch)) with Actions=0x8300003f (Rslv GenSQL ExeSQL Alrt XTab EvalVw
```

```
LclCch UptLclCch), Flags=0x1000180(OSrcCch  
UptoSrcCch)
```

Answer: Intelligence Server begins executing the report instance.

- Message E:

```
286:[THR:480][06/27/2018::12:24:25:181][DSS  
ReportServer][Report Source Tracing]Finding in  
cache: ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

Answer: Intelligence Server checks to see whether the report exists in the report cache.

- Message F:

```
286:[THR:480][06/27/2018::12:24:25:342][DSS  
ReportServer][Report Source Tracing]Not found in  
cache: ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

Answer: Intelligence Server did not find the report in the cache.

- Message G:

```
286:[THR:314][06/27/2018::12:24:25:432][DSS  
ReportServer][Report Source Tracing]No prompts in  
ReportInstance(Name="Length of Employment"  
ExecFlags=0x1000180(OSrcCch UptoSrcCch) ExecActn=  
0x1000180(RslvCB LclCch))
```

Answer: Intelligence Server checks for prompts and finds none in the report.

- Message H:

```
286:[THR:492][06/27/2018::12:24:26:634][DSS  
ReportServer][Report Source Tracing]Executing  
ReportInstance(Job=2 Name="Length of Employment"  
ExecFlags=0x1000184(OSrcCch UptoSrcCch) ExecActn=  
0x1000184(ExeSQL RslvCB LclCch)) with Actions=  
0x300003f(Rslv GenSQL ExeSQL Alrt XTab EvalVw  
LclCch UptLclCch), Flags=0x1000184(OSrcCch  
UptoSrcCch)
```

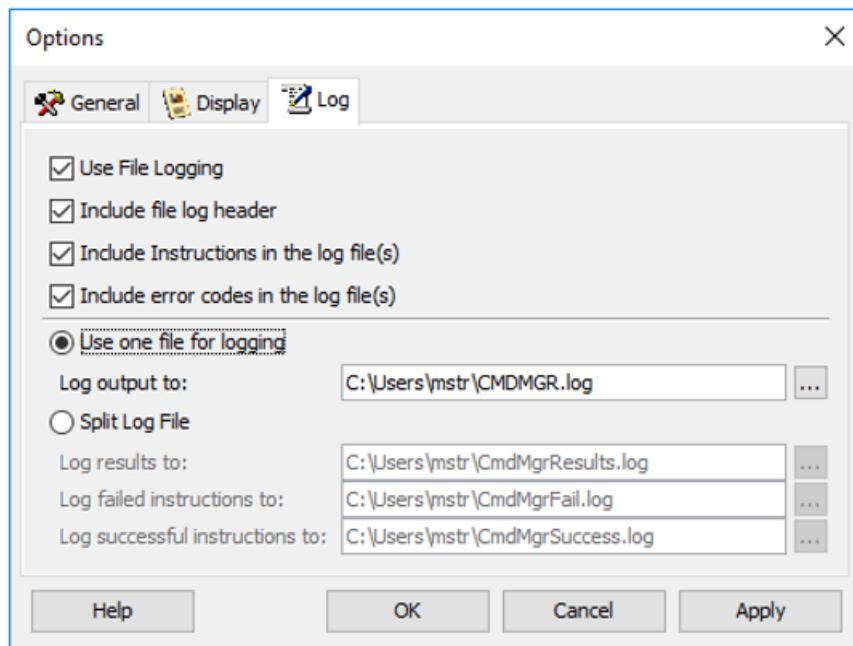
Answer: Intelligence Server executes the report and updates the caches.

Exercise 7.2: Enable logging in Command Manager

The MicroStrategy platform provides you the ability to enable logging through various interfaces. In this exercise, you will enable logging in Command Manager.

Access Command Manager script to enable logging

- 1 On the Windows machine in your cloud environment, right-click the Windows **Start** button and select **Search**.
- 2 In the **Search Windows machine**, search for and click **Command Manager**.
- 3 Select **Connect to a Project Source** and log in to the **MicroStrategy on AWS I-Server** project source using the login credentials provided in the MicroStrategy Cloud email.
- 4 On the **Tools** menu, select **Options**. Then click the **Log** tab.
- 5 Select **Use File Logging**.
- 6 Command Manager gives you the option to split the logs between Results, Failed Instructions, and Successful Instructions. For this exercise, you will send all logs to one single file. To do so, in the Log output box, specify **C:\Users\mstr\sCMDMGR.log**, if not already specified.



7 To finish enabling logging, click **OK**.

Resolving issues: tools for troubleshooting

The Platform Administrator needs to establish guidelines for tools that can be used for troubleshooting. In the MicroStrategy analytics environment, a variety of tools are available to assist with troubleshooting, including the following:

- MicroStrategy Diagnostics and Performance Logging tool
- DSSErrors.log file
- Core dump and stack trace
- MicroStrategy Web server diagnostics and statistics



You can also use the diagnostics and statistics feature for MicroStrategy Mobile Server. It is similar in functionality to the MicroStrategy Web diagnostics and statistics. For details, refer to the MicroStrategy Mobile Architect course.

- DB Query tool

The following sections discuss each of these tools.

Configuring what is logged: MicroStrategy Diagnostics and Performance Logging tool

Intelligence Server has the capability to log a lot of information since it performs most of the functions in a MicroStrategy analytics environment. You can log information for many aspects of Intelligence Server along with the operating system features and functions. By default, logging is set to a minimum as excessive logging can degrade the system's performance. However, the Platform Administrator should establish guidelines that his team can use detecting problems in the system for which logging is not enabled by default.



By default, logging to the DSSErrors.log file is enabled.

The MicroStrategy platform enables you to configure logging for various components by using a Java-based, multi-platform Diagnostics and Performance Logging tool. The tool, which can be used in the Windows and the Linux/Unix platforms, comes with a default logging configuration already set up but you can customize logging to gather information on the system components and performance counters pertinent to your environment. You can save logged messages to a default log file or a new log file.

Exercise 7.3: Enable report SQL tracing

In this exercise, you will launch and explore the Diagnostics and Performance Logging tool on a Linux machine.

After accessing the tool, you will log the report SQL generated by Intelligence Server to retrieve MicroStrategy object definitions from the metadata as well as data from the warehouse. Tracing the report SQL is one of the commonly used logging options in the analytics environments. You will log the SQL to a new log file called SQLTrace.

After enabling the logging for SQL tracing, you need to restart the Intelligence Server for the log settings to take effect. As restarting the Intelligence Server can take a few minutes, for this exercise, you do not need to restart it.

Access Linux machine using Putty

In a previous exercise, you launched Xming and Putty to connect to your Linux environment from the Windows box in your cloud environment. You will now access Putty and then launch Diagnostics and Performance Logging tool.

- 1 On your Windows desktop, access the Linux environment using Putty.

 If you were logged out, in the Putty Configuration window, double-click **MySession**, and then log in to your SSH session using the user name and password provided in the Welcome to MicroStrategy on AWS email.

Access the bin subdirectory in the home directory on your Linux machine

By default, in Linux environments, all MicroStrategy tools such as Diagnostics and Performance Logging tool are installed in the bin subdirectory of your home directory. As a result, you will first change the directory path to the bin subdirectory and from there, launch Connectivity Wizard.

- 2 On the console, type **cd /opt/mstr/MicroStrategy/bin** and press **Enter**.

- 3 On the console, type **./mstrdiag** and press **Enter**.

 You may need to maximize the Xming window to see the Diagnostics and Performance Logging tool window.

- 4 The Diagnostics and Performance Logging tool contains two tabs—Diagnostics Configuration and Performance Configuration.

The Diagnostics Configuration tab enables you to log diagnostics for specific system components for which you can log diagnostics messages. Examples include such as the Analytical Engine, Authentication Server and Datamart Executor.

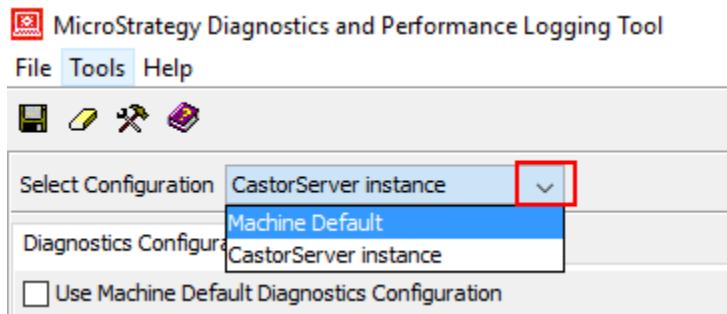
For each component, you can select a dispatcher which is the trace type or system activity about which a message is written to a given log file. You can choose from the following dispatchers:

- **Error**—This dispatcher logs the final message before an error occurs, which can be important information to help detect the system component and action that caused or preceded the error.
- **Fatal**— This dispatcher logs the final message before a fatal error occurs, which can be important information to help detect the system component and action that caused or preceded the server fatality.
- **Info**— This dispatcher logs every operation and manipulation that occurs on the system.

The component/dispatcher combinations you choose depend on your environment, your system, and your users' activities.

The Performance Configuration tab enables you to fine-tune performance diagnostics logging by determining the specific operating system categories and counters for which you want to log diagnostics. For example, you can log information to determine the amount of time it takes the CPU to operate a given system function.

Return to the Diagnostics Configuration tab, if not there already, and click the down arrow beside the **Select Configuration** drop-down list.



You will see two options:

- **Machine Default**—Displays components and counters for the machine on which the user launched the Diagnostics and Performance Logging tool. For example, if the machine has Developer as the only MicroStrategy application, it will only display Developer-related components.

- **CastorServer instance**—Displays components and counters for the Intelligence Server instance only.

5 Select **CastorServer instance**.

When you select **CastorServer instance**, an additional check box is available:

- On the Diagnostics Configuration tab, this check box is named **Use Machine Default Diagnostics Configuration**.
- On the Performance Configuration tab, this check box is named **Use Machine Default Performance Configuration**!

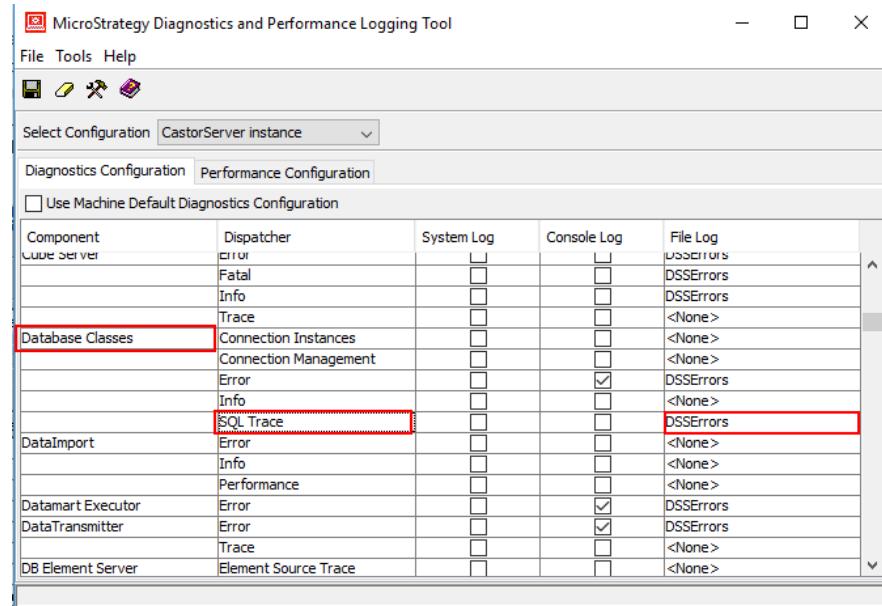
Selecting this check box on either of the two tabs enables you to configure the Intelligence Server instance with the same settings as the Machine Default for this machine. The MicroStrategy platform logs whatever information is configured for Machine Default at runtime even if you have changed and saved information on either tab.

For this exercise, leave the check boxes on both tabs unselected.

Enable report SQL trace

MicroStrategy Engine uses SQL to retrieve information about the project definition (such as schema, metrics, templates, filters, and reports) by querying the metadata. SQL is also used to retrieve information that is shown in reports, prompts, and element browsing and Enterprise Manager activities such as Statistics Purge and Data Load. Logging these queries is necessary to troubleshoot SQL related issues. You will now enable SQL tracing on the MicroStrategy platform for troubleshooting purposes.

- 6 On the Diagnostics Configuration tab, for the **Database Classes** component, locate the **SQL Trace** dispatcher.



- 7 You can log to any of the other existing files, or log in a new file. For this exercise, you will log information to a new file as you will be turning off the tracing after resolving the issue and want to keep this file separate from other existing log files.

In the File Log drop-down list, select **<New>**. The Log Destination Editor displays. Each time a new log file is created with the New option, a new entry is added to the list of available options under the Select Log Destination drop-down list.

- 8 In the File name box, enter **SQLTrace** (or any other name of your choice) as the name for your new log file.

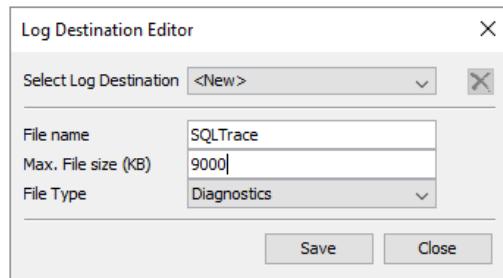
In the Max File Size (KB) field, enter the maximum size for your log file. The default value is 2048 KB but as SQL trace can log a lot of information, for this exercise, you will increase the value to **9000** KB. Information is always appended to a log file. When a file reaches its maximum size, the existing file is backed up (with a .bak extension) and a new file is created.

In the File Type drop-down list, you can specify whether you want this log file to record diagnostics data (useful for troubleshooting) or performance data (useful for system tuning).

- If you select **Diagnostics** as the file type, your new log file becomes available as a selection from the File Log column on the Diagnostics Configuration tab, and can record information for any of the component/dispatcher combinations.

- If you select **Performance** as the file type, your new log file becomes available as a selection under File name from the right-hand side of the Performance Configuration tab, and can record information on performance counters.

For this exercise, accept **Diagnostics** as the file type. Click **Save** and then **Close**.



- 9 On the Diagnostics and Performance Logging tool menu bar, click **Save** and then exit the tool.

Any changes you make to the settings in the Diagnostics and Performance Logging tool only apply to MicroStrategy products installed on the computer on which you are working. You cannot make changes to diagnostics settings for products on a remote computer.

Verify the creation of new log file

- 10 By default, all log files on a Linux machine in your environment are saved to the /opt/mstr/MicroStrategy/logs folder.

 By default, all log files on a Windows machine are saved to the C:\Program Files\Common Files\MicroStrategy\Log folder.

 Enabling SQL tracing for the Database Classes component can adversely impact performance due to intensive hard-disk access and should only be used for troubleshooting purposes.

After enabling the logging for SQL tracing, you need to restart the Intelligence Server for the log settings to take effect. As restarting the Intelligence Server can take a few minutes, for this exercise, you do not need to restart it.

Exercise 7.4: Review of frequently used diagnostics

In the previous exercise, you used the Diagnostics and Performance Logging tool to enable SQL Trace for the Database Classes component.

In this exercise, we will review some of the other diagnostic components that can be used for troubleshooting purposes.

Based on your prior knowledge of the MicroStrategy platform, answer the following questions:

- 1** Enabling SQL tracing for the Database Classes component logs SQL submitted by Intelligence Server against which of the following databases:
 - a Warehouse
 - b Metadata
 - c Enterprise Manager Statistics repository
 - d All of the above

- 2** Which diagnostic component will you enable to capture the SQL submitted against the metadata only?

- 3** Which diagnostic component will you enable for troubleshooting report caches and History List issues?

- 4** What diagnostics component can you use to determine how long it took for the report SQL to error out, assuming you do not have this information from manual testing or other means.

Exercise 7.4 Solutions: Review of frequently used diagnostics

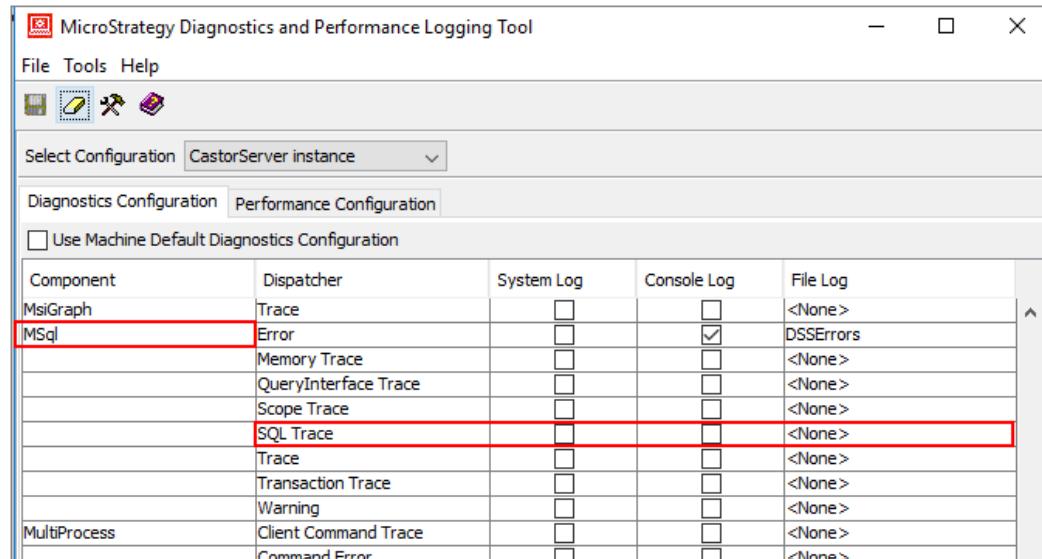
In the previous exercise, you used the Diagnostics and Performance Logging tool to enable SQL Trace for the Database Classes component.

In this exercise, we will review some of the other diagnostic components that can be used for troubleshooting purposes.

Based on your prior knowledge of the MicroStrategy platform, answer the following questions:

- 1 Enabling SQL tracing for the Database Classes component logs SQL submitted by Intelligence Server against which of the following databases:
 - a Warehouse
 - b Metadata
 - c Enterprise Manager Statistics repository
 - d **All of the above**
- 2 Which diagnostic component will you enable to capture the SQL submitted against the metadata only?

Answer: MSql > SQL Trace



- 3 Which diagnostic component will you enable for troubleshooting report caches and History List issues?

Answer:

- For report caching-related issues, enable Report Server > Cache Trace
 - For History List-related issues, enable Kernel > User Trace
- 4 What diagnostics component can you use to determine how long it took for the report SQL to error out, assuming you do not have this information from manual testing or other means.
-

Answer:

You can obtain this information by enabling the following two components:

- a Database Classes > SQL Trace
- b Database Classes > Error (by default this is already enabled and logs to DSSErrors.log)

You can use the SQL Trace to determine the exact time when the SQL was submitted and the DSSErrors.log file to identify the exact time it took for the report SQL to error out.

Consider the following example:

SQL Trace:

2017-10-18 17:11:36.406+08:00 [HOST:ULPSPJ][PID:904][THR:1124][Database Classes][SQL Trace] Executing SQL. Connection ID: 2. SQL: select * from dwtemp.Administrato_2_170533

DSSErrors.log:

2017-10-18 17:11:46.515+08:00 [HOST:ULPSPJ][PID:904][THR:1124][Database Classes][Error] Execute Query failed.

*Error type: Odbc error. Odbc operation attempted: SQLExecDirect. [HYT00:0: on SQLHANDLE] [NCR][ODBC Teradata Driver] **Query timeout expired** Connection String: DSN=TD_Data;UID=dwwas;AUTHENTICATION=;. SQL Statement: select * from dwtemp.Administrato_2_170533.*

Based on the review of the SQL Trace and the DSSErrors.log file, you can see that the error occurred in 10 seconds (=11:36.406+08:00 - 11:46.515+08:00).

In addition, you can see from the error message in the DSSErrors.log file that 10 seconds was a query timeout setting.

Exercise 7.5: Troubleshoot Diagnostics and Performance Logging tool issues

In this exercise, you will launch the Diagnostics and Performance Logging tool on the Windows box.

Next, you will attempt to locate the Database Classes diagnostics component for the CastorServer instance, and then answer the following questions:

- Did you get any error message when attempting to open the Diagnostics and Performance Logging tool?

- Are you able to select the CastorServer instance option under the Select Configuration drop-down list on the Diagnostics tab? Why or why not?

- Do you see an option for the Database Classes diagnostics component? Why or why not?

Exercise 7.5 Solutions: Troubleshoot Diagnostics and Performance Logging tool issues

In this exercise, you will launch the Diagnostics and Performance Logging tool on the Windows box.

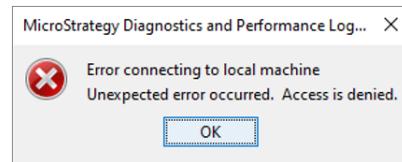
Access Diagnostics and Performance Logging tool

You can launch the Diagnostics and Performance Logging tool using the Diagnostics option under MicroStrategy Tools on the Start button.

- 1 On the Windows box of your cloud environment, click the **Start** icon, and under **MicroStrategy Tools**, click **Diagnostics Configuration**.

Did you get any error when attempting to open the Diagnostics and Performance Logging tool?

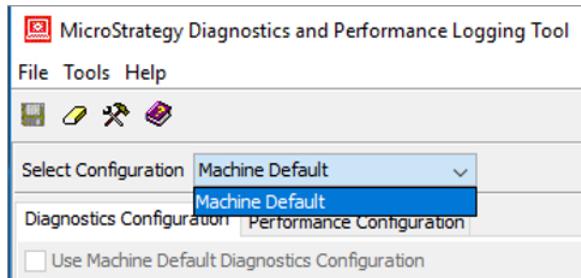
Yes, the following error message displays:



The error occurs because you need to launch the Diagnostics and Performance Logging tool as an administrator. To do so, on the Windows box of your cloud environment, click the **Start** icon, and under **MicroStrategy Tools**, right-click **Diagnostics Configuration**, point to **More**, and select **Run as administrator**.

- 2 If displayed, in the User Account Control window, click **Yes**.
- 3 On the Diagnostics tab, in the **Select Configuration** drop-down list, are you able to select the **CastorServer instance** option under the **Select Configuration** drop-down list on the Diagnostics tab? Why or why not?

No, the CastorServer instance option is not available under the Select Configuration drop-down list.



This is because you are accessing the Diagnostics and Performance Logging tool on the Windows box. The CastorServer instance option refers to the components and counters for the Intelligence Server instance but the Intelligence Server is installed on the Linux machine of your cloud environment.

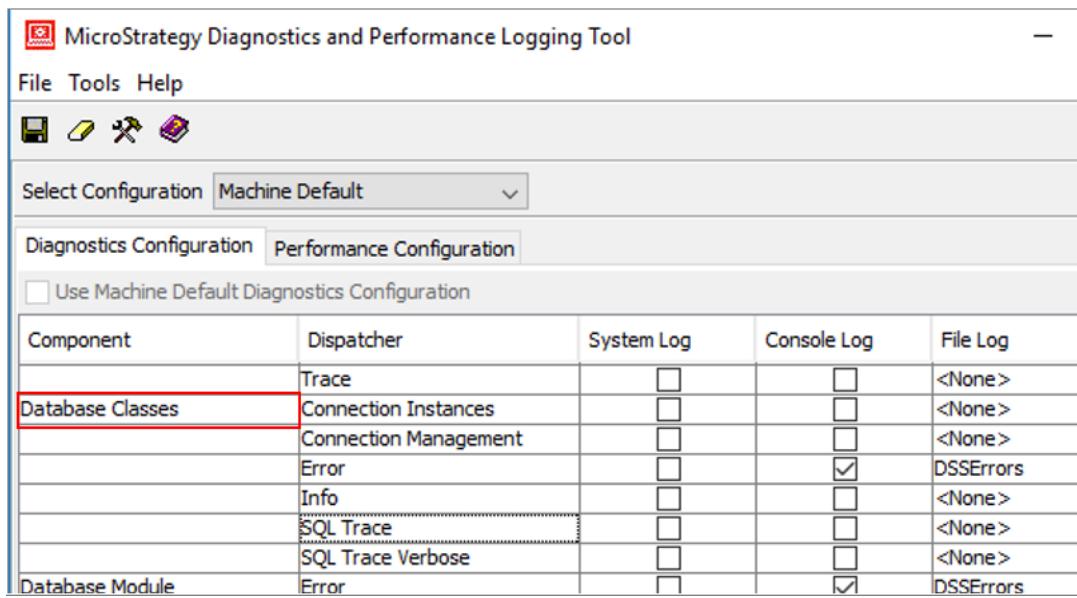
The Windows machine in your cloud environment does not have the Intelligence Server component installed on it. The *CastorServer instance* set of traces applies only to the Intelligence Server, and will be visible if you access the Diagnostics and Performance Logging tool on the Linux box.

- 4 While still on the Diagnostics tab, scroll down and locate the **Database Classes** component.

Do you see an option for the Database Classes diagnostics component? Why or why not?

Yes, you will see an option for the Database Classes diagnostics component even though Intelligence Server is not installed on the Windows machine. The Database Classes diagnostics component option displays because Developer

is installed on the Windows machine. Like Intelligence Server, Developer can also generate SQL to retrieve object definitions from the metadata.



- 5 Exit the Diagnostics and Performance Logging tool without enabling any tracing.

Troubleshooting Intelligence Server issues: Analyzing DSSErrors.log file

Another useful that the platform administration team can use in troubleshooting the Intelligence Server-related issues is the DSSErrors.log file.

When Intelligence Server encounters an error, it throws an exception. Some of these exceptions are considered fatal, causing Intelligence Server to shut down. These exceptions are logged in the DSSErrors.log file often as unknown exceptions.



Fatal exception messages by themselves may not be sufficient for accurate diagnosis.

By default, the DSSErrors.log is created on the Intelligence Server machine in the following location:

- **Windows:** C:\Program Files (x86)\Common Files\MicroStrategy\Log

- **Linux/Unix:** The default location of the DSSErrors.log depends on the Log Path specified during installation. The Log Path can be located by looking in the install.log file as shown in the following image:

```
*****
MicroStrategy Installer for Unix/Linux
*****
MicroStrategy Suite information:
Home Path: /opt/mstr/MicroStrategy
Install Path: /opt/mstr/MicroStrategy/install
Log Path: /opt/mstr/MicroStrategy/log
Installation size: 12 GB
Operation started on 04/04/2018 16:34:03:932 UTC
```

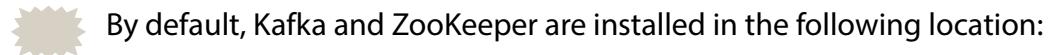


On the Linux machine in your cloud environment, the file is located at:

/opt/mstr/MicroStrategy/log

Intelligence Server can write directly to the DSSErrors.log or use MicroStrategy Messaging Services. Messaging Services is a component that is coupled with the Intelligence Server during installations and upgrades. It uses the following services:

- Apache Kafka
- Apache ZooKeeper
- MicroStrategy Intelligence Server Log Consumer



By default, Kafka and ZooKeeper are installed in the following location:

- <Intelligence Server installation path>\MicroStrategy\Messaging Services\Kafka\
- Intelligence Server Log Consumer is installed in the following location:
- <Intelligence Server installation path>\MicroStrategy\Intelligence Server\KafkaConsumer

By default, when you create a server definition, logging to DSSErrors.log using Messaging Services which is configured out-of-the-box and runs automatically after the installation is completed. Intelligence Server logs using Messaging Services Server only when Messaging Services feature is enabled and Kafka Server can be connected successfully. Logs are sent to local disk if Messaging Services is disabled or the Kafka Server is down or unreachable because of network issues.

Server State Dump

An important component of DSErrors.log file is Server State Dump (SSD). When an Intelligence Server shuts down unexpectedly, it logs SSD which is a collection of information related to the state of Intelligence Server at the time of an unexpected shutdown of Intelligence Server.

SSD provides insight into what was going on in Intelligence Server when the shutdown occurred. This information can be used in diagnosing the cause of the shutdown and avert subsequent problems. An analysis of the logged information in the SSD is also helpful in the tuning of Intelligence Server for memory usage.

This extra logging for SSD is triggered under a limited number of conditions described below so as not to hinder performance due to extra logging:

- When Intelligence Server starts up
- When an unknown exception condition occurs during execution of a Intelligence Server process
- When Intelligence Server shuts down due to exhaustion of the memory resources
- When the Intelligence Server Configuration Editor settings are changed
- When the Project Configuration Editor settings are changed
- When Intelligence Server reaches the 'throttling' state. Throttling occurs when Intelligence Server memory usage exceeds the governing limit specified for the Maximum Intelligence Server use of total memory (%) setting in the MicroStrategy Intelligence Server Configuration Editor. When MicroStrategy Intelligence Server reaches throttling state, it denies all requests from a MicroStrategy Web client (or a client built with the MicroStrategy Web API) until the memory usage drops below the limit.

Core dump and stack trace

Core dump file

When the Intelligence Server crashes, a core dump file is generated by the operating system for the Intelligence Server (MSTRSvr) process. A core dump is a snapshot of the Intelligence Server in memory at the time of the crash.

By reading the core dump file, MicroStrategy Technical Support can understand the actions being taken immediately prior to the crash, and may be able to obtain

information such as the report or username which caused the crash. They can also get the call stack from the core dump file. A call stack provides information on the functions or methods being used at the time the SSD was written.

 Core file generation is controlled by the operating system, not by MicroStrategy processes, so difficulties in generating or obtaining core files must be followed up with your system administrator.

The core dump file is created in the location specified by the Linux operating system administrator. By default, it is created under the IntelligenceServer subdirectory under the home directory.

“Core dump” is a generic term that captures the state of the process (in this case, the Intelligence Server). With core dumps, it is important to distinguish between a crash mode dump and a hang mode dump:

- A crash mode dump is a dump of information when the Intelligence Server hits a failure point (a crash).
- A hang mode dump is a snapshot of the Intelligence Server state information that is deliberately invoked at 2 or 3 different points in time (usually 5 minutes apart).

When the Intelligence Server encounters an abnormal shutdown (such as a crash), the operating system generates a core file (called a crash mode dump) that is used to perform a root cause analysis of the crash.

When Intelligence Server encounters unresponsiveness (such as a hang), you can do the following:

- Windows: Collect hang mode dumps
- Linux: Collect pstacks (detailed below)

Stack trace report

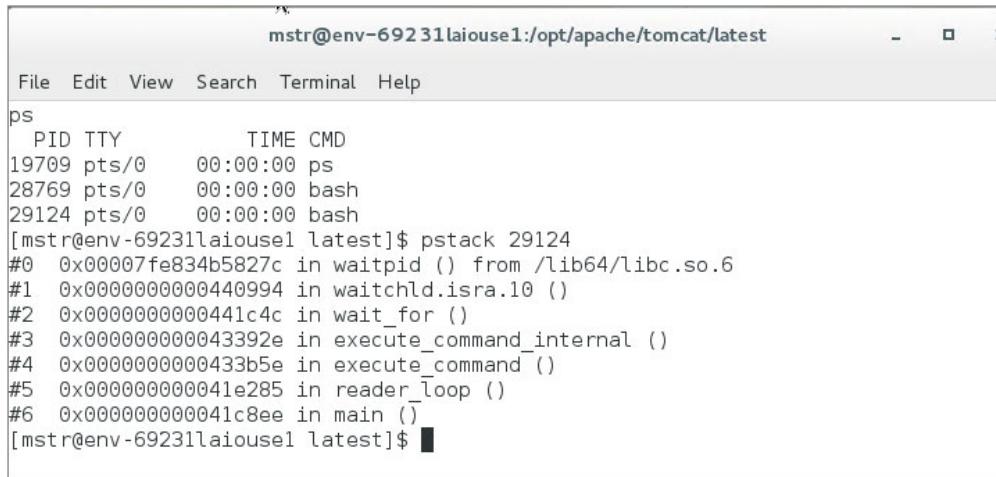
A stack trace is a report that provides information about application functions or methods. It provides valuable information for debugging application issues.

When troubleshooting Intelligence Server issues, you may need to log a stack trace of a running process. In Linux, you can use the pstack command to print a stack trace of a running process, once attached to a process. The command is:

```
pstack pid
```

pstack attaches to the active process named by the pid on the command line, and prints out an execution stack trace. If the process is part of a thread group, then pstack will print out a stack trace for each of the threads in the group.

pstack PID Output



The screenshot shows a terminal window titled "mstr@env-69231laiouse1:/opt/apache/tomcat/latest". The window contains a menu bar with File, Edit, View, Search, Terminal, and Help. Below the menu is a command-line interface. The user runs the "ps" command to list processes, then runs "pstack 29124" to generate a stack trace for the process with ID 29124. The stack trace shows the call stack for the thread associated with PID 29124, listing various library functions and internal routines.

```
mstr@env-69231laiouse1:/opt/apache/tomcat/latest
File Edit View Search Terminal Help
ps
  PID TTY      TIME CMD
19709 pts/0    00:00:00 ps
28769 pts/0    00:00:00 bash
29124 pts/0    00:00:00 bash
[mstr@env-69231laiouse1 latest]$ pstack 29124
#0 0x00007fe834b5827c in waitpid () from /lib64/libc.so.6
#1 0x0000000000440994 in waitchld.isra.10 ()
#2 0x0000000000441c4c in wait_for ()
#3 0x000000000043392e in execute_command_internal ()
#4 0x0000000000433b5e in execute_command ()
#5 0x000000000041e285 in reader_loop ()
#6 0x000000000041c8ee in main ()
[mstr@env-69231laiouse1 latest]$
```

Exercise 7.6: Log a stack trace

You are experiencing slow performance in your MicroStrategy environment. The MicroStrategy Technical Support team has asked you to send them a stack trace of the MSTRSvr process.

To generate the stack trace, you will use the top utility to find out the process ID (PID) of the MSTRSvr process, and then use pstack command to generate the stack trace associated with the MSTRSvr process. Since you need to send the file to the Support team, you will output the stack trace to a text file named MyPstack.txt.

 As the main purpose of this exercise is to show you how to generate the stack trace, instead of the MSTRSvr process, you can use any other process as well.

Use Putty to access the Linux machine

As you used Putty in an earlier exercise, you can just maximize and access the Putty window. Since you will be outputting the stack trace to a text file, for the

ease of locating the text file, you will first change the current directory to the /home/mstr (home directory).

1 Access the Putty window.



Log in again in case you were logged out

2 To change the current directory to the home directory, on the console, type:

cd /home/mstr

Identify the PID

3 To identify the PID of MSTRSVR process, you can execute the top utility. To do so, on the console, type:

top

It displays processes that are consuming the most CPU and memory resources at the time when the command is executed.

Tasks: 250 total, 1 running, 249 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 1.2 us, 0.5 sy, 0.0 ni, 98.0 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st										
KiB Mem : 13395672 total, 1323488 free, 8769916 used, 3302268 buff/cache										
KiB Swap: 0 total, 0 free, 0 used. 4192248 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
4749	mstr	20	0	2432396	662136	140944	S	2.0	4.9	0:50.28 MSTRSvr
1192	root	20	0	127404	3340	2404	S	0.7	0.0	0:03.69 monit
4642	mstr	20	0	5812976	108664	12940	S	0.7	0.8	0:31.71 java
4220	mstr	20	0	5007060	570272	15096	S	0.3	4.3	0:47.59 java
4521	root	24	4	745024	39344	5480	S	0.3	0.3	0:27.28 aws
4660	mstr	20	0	3749124	681780	15652	S	0.3	5.1	1:15.88 java
4668	mstr	20	0	9099300	2.695g	15220	S	0.3	21.1	2:25.37 java
9899	mstr	20	0	157852	2416	1556	R	0.3	0.0	0:00.02 top
1	root	20	0	193640	6752	3980	S	0.0	0.1	0:05.70 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.08 ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.24 timer/0:0

Locate the PID of the MSTRSvr process. For example, in the preceding image, the PID is 4749.



As the displays keeps updating, you can press **Control-Z** to return to the prompt, and then scroll up or down to locate the PID of the MSTRSvr process.

Generate the stack trace

To generate the stack trace, you will use the pstack command.

4 On the console, type:

pstack 4749 > MyPstack.txt

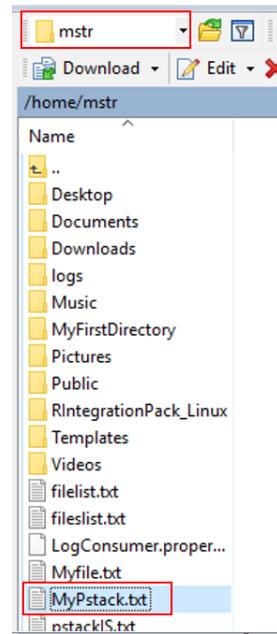
The stack trace is generated and is output to a text file in the home directory.

Locate the log file

You will now locate the log file in the home directory using WinSCP.

- 5 Access WinSCP and navigate to **/opt/home/mstr/**. You should see your **MyPstack.txt** log file; if not, click **Refresh** - it should display the file.

If you need to download it, you can right-click the file and then click **Download**.



Troubleshooting MicroStrategy Web server issues: Diagnostics and Statistics features

User actions in MicroStrategy Web may generate errors, warnings, or messages. The best-in-class Platform Administrator should require his team to frequently reviews logs to proactively fix potential issues and troubleshoot user-reported problems. In the sections below, we will review setting up and analyzing diagnostics and statistics for the MicroStrategy Web server.

Exercise 7.7: Set up diagnostics

The MicroStrategy Web diagnostics feature enables you to capture errors, warnings, and messages in a log file that can then be analyzed for troubleshooting purposes. You can configure both the type of information that MicroStrategy Web logs and the location of the log file.

To establish diagnostics, you can define what information is logged by the MicroStrategy Web server, as well as where it is logged. You can use the MicroStrategy Web Administrator page to enable diagnostics. As the Platform Administrator, you should set up diagnostics for your platform administration team.

Set up internal diagnostics

- 1 Access the MicroStrategy Web Administrator page and click **Configuration**.
- 2 You can choose between two diagnostic setups: Internal and Custom. Keep **Internal** selected. Custom is typically used to load a logger.properties file provided by MicroStrategy Technical Support.
- 3 As you launch the Intelligent Enterprise, you want to log all levels of information to keep track of utilization and errors. In the **Levels** drop down, select **Messages**. This will log all errors, warnings, and messages.
- 4 Keep the **Maximum output file size** at 10,000,000 bytes. When the maximum file size is reached, a new log file is started.
- 5 Keep the **Number of file outputs** at 100. This is the maximum number of log files that will be created.
- 6 Leave the **Flash profiler** drop-down disabled. The Flash profiler helps troubleshoot Flash documents, which the MartZon enterprise does not use.

Enable XML-API logs

For your team to better identify and debug API issues, the Platform Administrator, can enable XML-API logs. The logs provide:

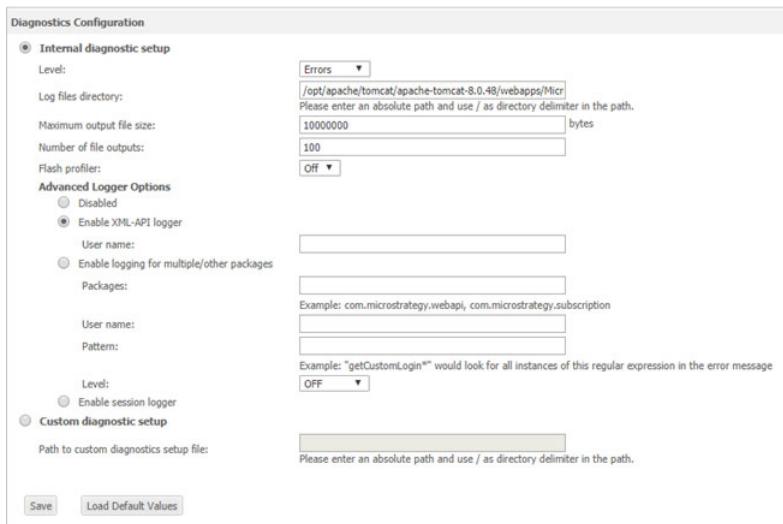
- The name of the class and method being accessed.
- Custom messages used in the code to specify the reason why the message was logged.

- All argument values sent to the method that logged the message.

In the steps below, you will use the Advanced Logger option to enable XML-API logging. Once enabled, your team can determine the specific user and package that is causing errors. This makes it easier to debug API issues as each log contains unique information about a specific user, date, and package.

- 1 From the MicroStrategy Web Administrator page, select **Configuration**.
- 2 Select **Enable XML-API logger** under Advanced Logger Options.
- 3 Leave **Username** blank. Only add users when you want to specify that the XML-API logger only saves messages for that user. If you do not specify a user, messages are logged for all users.
- 4 Click **Save**. The xml logs are now generated in the MicroStrategy Web deployment path \WEB-INF\log.

The diagnostics configuration should look similar to the image below:



Analyze system and server performance: Web statistics

The Platform Administrator should work with other Intelligence Center architects and log MicroStrategy Web statistics when the system is not working properly or when critical or enterprise applications are not performing to expectations.

You can enable statistics to interpret and analyze system and server performance and disable them when you no longer need to monitor performance. For example, you can obtain information about the time taken by the MicroStrategy

Web server and Intelligence Server to complete an operation, how much data is received and sent, the waiting time to receive some data from the Intelligence Server, and so on.

Best Practice

If the statistics are being saved in the file, the file size grows quickly, which is unnecessary. You should not log statistics unless the system is not working properly and you want to analyze the data for system tuning or troubleshooting.

Exercise 7.8: Set up web statistics

In this exercise, you will configure your Web server's statistics to ensure you and your team have access to the appropriate information to monitor analytics environments.

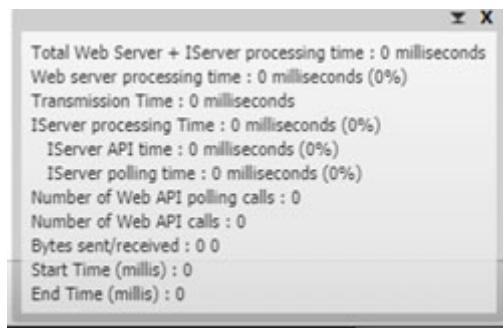
Configure web statistics

- 1 On the Web Server Administrator page, under Diagnostics, click **Statistics**.
- 2 For **Mode**, select **Screen**. This option displays in the lower left corner of all MicroStrategy Web pages.

The other options are:

- **File:** Statistics are written to the file specified in the **source code control** box. Specify the absolute path to the file with / as the directory delimiter. For example, if the file is DemoStats and it is stored on the /C/ drive of the Web server, enter C:/DemoStats.
- **OFF:** Statistics are not displayed on screen nor written to a file. By default, the mode is set to this option.
- **Screen and file:** Statistics are both displayed on screen and written to the file specified in the **Statistics file** box.

- 3 Click **Save**.



Notice the pop-up on your screen displays web page statistics.

- 4 To revert to the default settings, click **Load Default Value**.

Viewing server logs

The information collected in server logs can be difficult to understand if you read it directly from the log file itself. As an alternative to scanning a log that contains all information collected for your system activity, you can filter and view logged information using the View Logs page. This allows you to more easily locate and troubleshoot application errors in the system.

Time	User name	User IP Address	Level	Class	Method	Message
05/03/2018 13:11:51:013			SEVERE	CDSXMLServerSessionImpl	CreateSessionEx	(Login failure)
05/03/2018 13:11:46:929			SEVERE	CDSXMLServerSessionImpl	CloseSession	MoSessionManager::IsUserLoggedIn(): user session is invalid when trying to add new commands in. (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:51:096			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:43:192			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:12:054			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:30:46:113			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:30:36:821			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:51:000			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:02:005			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:28:54:185			SEVERE	CDSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:28:46:525			SEVERE	CDSXMLServerSessionImpl	CloseSession	MoSessionManager::IsUserLoggedIn(): user session is invalid when trying to add new commands in. (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 01:52:27:152			SEVERE	GenericWebAppController	errorAfterRedirect	The URL you have selected for re-direction is invalid. Please verify that the URL syntax is correct, and note that it must be relative and not absolute. (Servlet execution threw an exception)
05/02/2018 01:52:27:101			SEVERE	GenericWebAppController	processRequest	null (java.lang.StackOverflowError)
05/02/2018 01:52:27:080			SEVERE	GenericWebAppController	processRequest	null (java.lang.StackOverflowError)
05/01/2018 21:24:31:942			SEVERE	CDSXMLServerSessionImpl	getWindowsNTSID	Unable to find DLL which supports NT authentication.

Based on the analysis of the server logs, the Platform Administrator should work with other Intelligence Center architects to resolve any system issues.

 The steps below outline how to view the MicroStrategy Web server log file. These steps are provided for reference, they are not intended to be performed in class.

View the MicroStrategy Web server log file

- 1 On the MicroStrategy Web Administrator page, under Diagnostics, click **View logs**.
- 2 In the Display area, select the check boxes for the log information you want to display on the View logs page:
 - **Errors:** Logged errors are displayed on the screen. This is selected by default.
 - **Warnings:** Logged warnings are displayed on the screen.
 - **Messages:** Logged messages are displayed on the screen. In the **From** area, specify the start date of logged information to display on the screen.

- 3 In the **To** area, specify the end date of logged information to display on the screen.
- 4 To display logged information, click **Refresh**. The log file information is displayed at the bottom of the page, containing:
 - **Time**: The time and date when the event log was created
 - **User name**: The name of the user who logged in
 - **User IP Address**: The IP address of the user who logged in
 - **Level**: Enables you to specify whether to log Error, Warning, or Message
 - **Class and Method**: Code references for the issue
 - **Message**: The text of the logged message
- 5 To sort a column, click the **Sort** icon in the column's header.

Troubleshooting data source issues: using DB Query tool

MicroStrategy DB (Database) Query tool is an application that uses ODBC and Java Database Connectivity (JDBC) to access data in any database for which there is an ODBC driver or JDBC driver, respectively. This tool can be used to quickly test and troubleshoot connectivity to data sources as well as create and execute SQL statements on your data sources for various purposes.

Exercise 7.9: Create a Freeform SQL report

In this exercise, you will create a Freeform SQL report which errors out on execution due to a syntax error in your SQL. Save this report as PA_FreeForm in a new folder named PlatAdmin under the Public Objects\Reports folder.

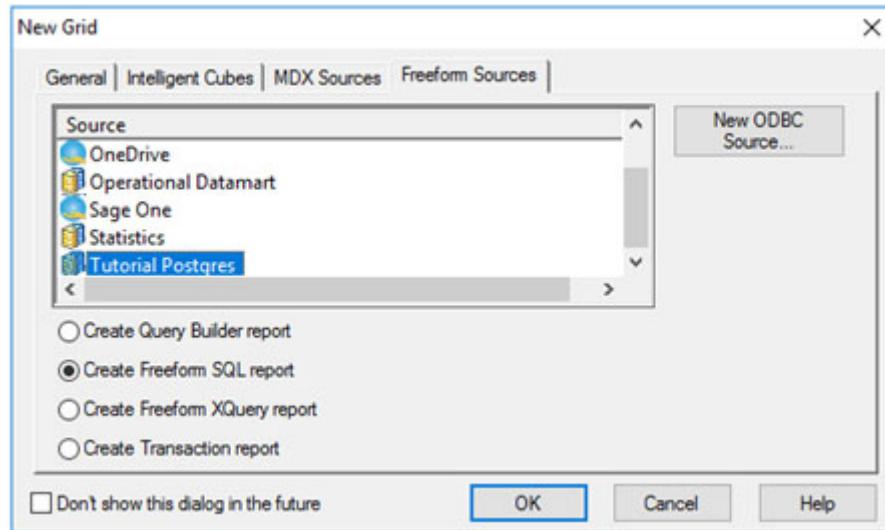
In the next exercise, you will troubleshoot and resolve the issue using the DB Query tool.

Create a folder

- 1 In Developer, in the MicroStrategy Tutorial project, navigate to the **Public Objects\Reports** folder.
- 2 In the Object Viewer, right-click an empty area, point to **New**, and select **Folder**. Then, type **PlatAdmin** as the folder name.

Create the FF SQL report

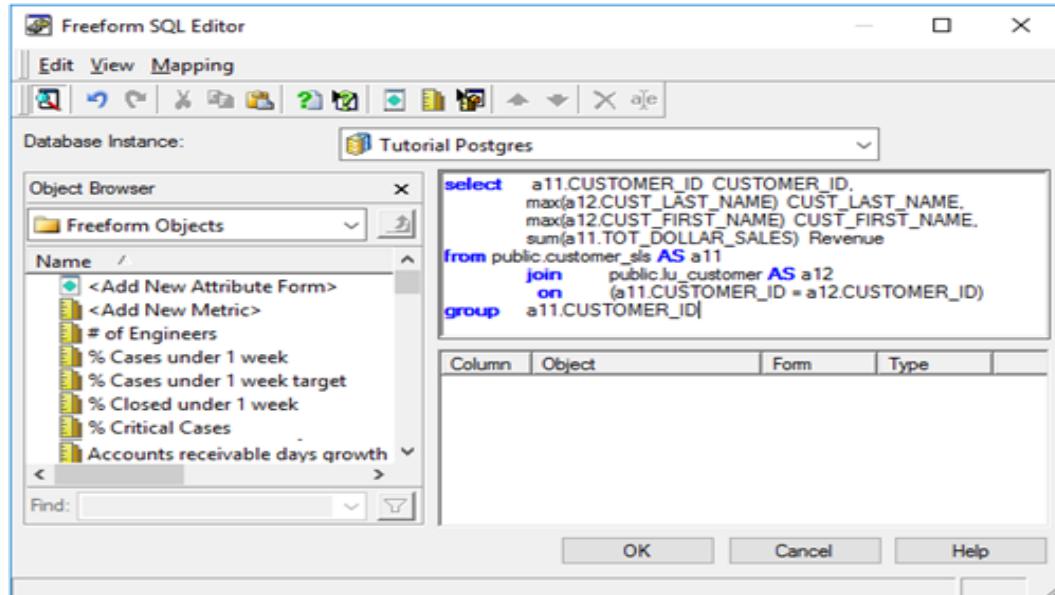
- 1 In the PlatAdmin folder, right-click an empty area in the Object Viewer, point to **New**, and select **Report**.
- 2 In the New Grid window, click the **Freeform Sources** tab.
- 3 Select **Create Freeform SQL report**.
- 4 In the Source list, select **Tutorial Postgres**.

5 Click OK.**Enter the SQL statement**

- 6** Copy the following SQL labeled and paste it into the SQL Statement pane of the Freeform SQL Editor:

```
select a11.CUSTOMER_ID CUSTOMER_ID,  
       max(a12.CUST_LAST_NAME) CUST_LAST_NAME,  
       max(a12.CUST_FIRST_NAME) CUST_FIRST_NAME,  
       sum(a11.TOT_DOLLAR_SALES) Revenue  
  from public.customer_sls AS a11  
 join public.lu_customer AS a12  
    on (a11.CUSTOMER_ID = a12.CUSTOMER_ID)
```

```
group a11.CUSTOMER_ID
```



Map the columns

- 7 In the Freeform SQL Editor, in the Object Browser, in the Freeform Objects folder, double-click **<Add New Attribute Form>**.
- 8 In the Mapping pane, for Column 1, in the Object box, type **Customer** as the attribute name. Leave the Form as **ID**.
- 9 In the Object Browser, in the Freeform Objects folder, double-click **<Add New Attribute Form>**.
- 10 In the Mapping pane, for Column 2, in the Object box, type **Customer** as the attribute name. In the Form drop-down box, type **Last Name**. Ensure Type is set to **Text**.
- 11 Double-click **<Add New Attribute Form>**.
- 12 In the Mapping pane, for Column 3, in the Object box, type **Customer** as the attribute name. In the Form drop-down box, type **First Name**. Ensure Type is set to **Text**.
- 13 In the Object Browser, in the Freeform Objects folder, double-click **<Add New Metric>**.

- 14** In the Mapping pane, for Column 4, in the Object box, type **Revenue** as the metric name. Ensure Type is set to **Number**. The completed column mapping should look like the following:

The screenshot shows the Mapped Columns pane in the Freeform SQL report editor. The Database Instance is set to Tutorial Postgres. The SQL query is:

```
select a11.CUSTOMER_ID CUSTOMER_ID,
       max(a12.CUST_LAST_NAME) CUST_LAST_NAME,
       max(a12.CUST_FIRST_NAME) CUST_FIRST_NAME,
       sum(a11.TOT_DOLLAR_SALES) Revenue
  from public.customer_sls AS a11
  join public.lu_customer AS a12
    on (a11.CUSTOMER_ID = a12.CUSTOMER_ID)
 group by a11.CUSTOMER_ID
```

The Mapped Columns table shows the following mappings:

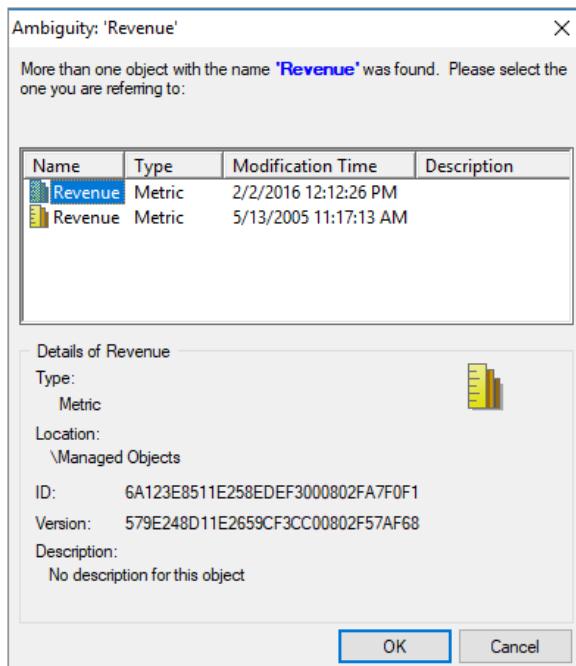
Column	Object	Form	Type
1	Customer	ID	Number
2	Customer	Last ...	Text
3	Customer	First N...	Text
4	Revenue		Number

Tip: Your column order needs to match the above image as you must map the columns in the same order as they appear in the SQL statement. You can drag and drop the columns to change the order or right-click the columns and use the Move Up or Move Down options.

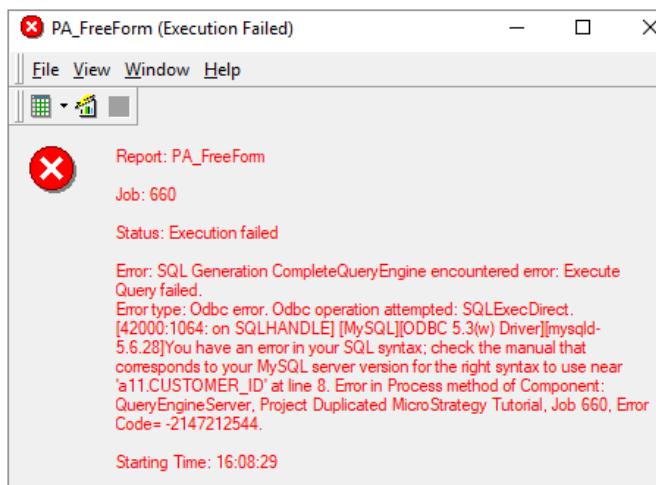
Save the Freeform SQL report

- 15** Close the editor and when prompted, click **Yes**.

- 16** If prompted for Customer in the Ambiguity window, select the first **Customer** attribute and click **OK**. Similarly, If prompted for Revenue, in the Ambiguity window, select the first **Revenue** attribute and click **OK**.



- 17** Right-click **Customer**, point to **Attribute Forms**, and ensure **ID**, **First Name**, and **Last Name** forms are selected.
- 18** Save your report as **PA_FreeForm** in the **Public Objects\Reports\PlatAdmin** folder. Then, run the report. The following error displays indicating an issue with your SQL syntax:



- 19** Return to the **Design View** of the report.

Exercise 7.10: Troubleshoot an issue using the DB Query tool

In the previous exercise, you created a Freeform SQL report that errored out on execution due to a syntax error in your SQL.

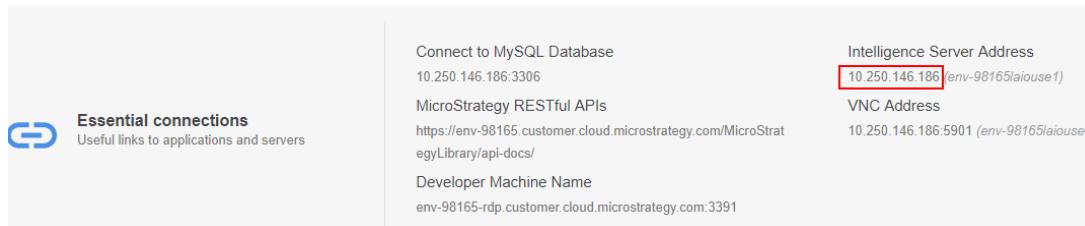
In this exercise, you will use the DB Query tool on the Windows machine to identify the source of error and then resolve the issue.

Access DB Query tool

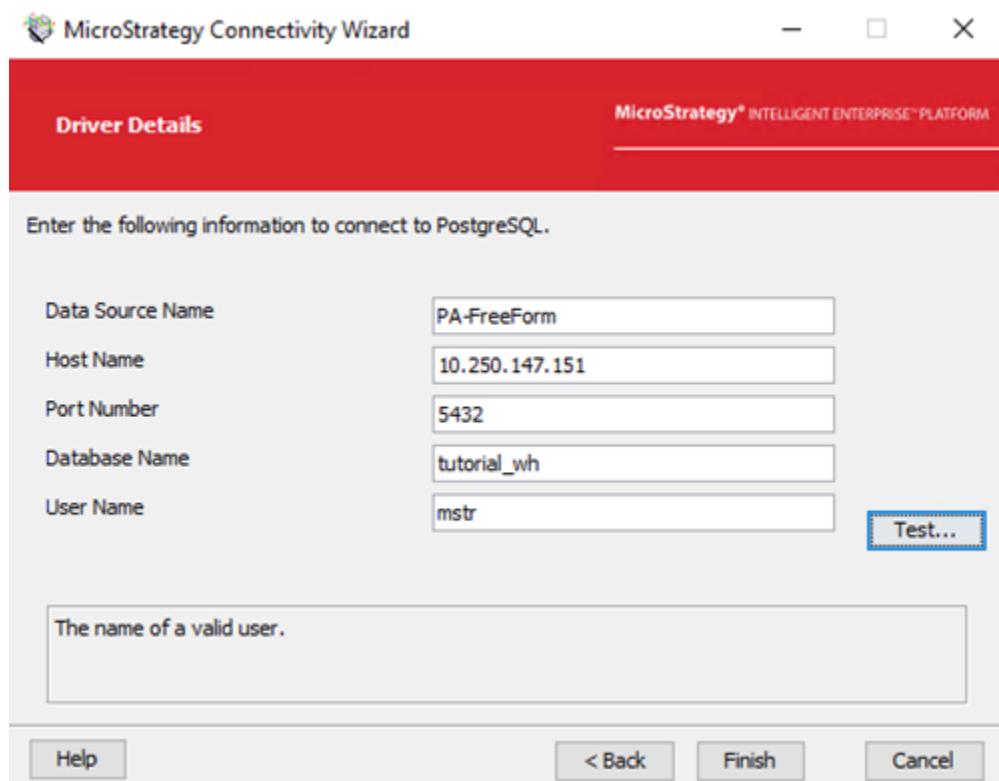
- 1 On your Windows Start menu, point to **MicroStrategy Tools**, and select **DB Query Tool**.
- 2 In the Connection drop-down list, select **New DSN**.
- 3 If displayed, in the User Account Control window, select **Yes**.
- 4 In the MicroStrategy Connectivity Wizard window, select **MicroStrategy ODBC Driver for PostgreSQL**, then click next.

5 In the Driver Details box, do the following:

- In the **Data Source Name** box, type **PA-FreeForm**.
- In the **Host Name** box, type the IP address of the Intelligence Server (listed on the landing page).

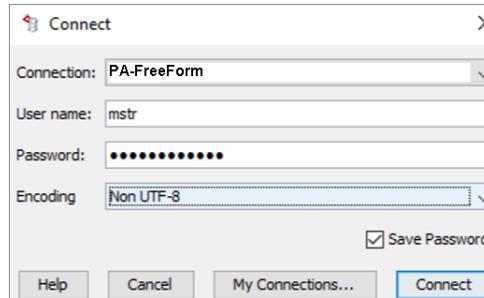


- In the Port box, leave the default value of **5432**.
- In the Database Name box, select **tutorial_wh**.
- In the User box, type **mstr**.
- In the Password box, type the password listed in the Welcome to MicroStrategy on cloud email.



6 Click **Test**. If the credentials were entered correctly, you see a message indicating that the connection was successful.

- 7 Click **Finish**, then click **OK** in the window indicating that the DSN was created successfully.
- 8 In the Connect window, in the User name box, type **mstr**. In the Password box, type the password listed in the Welcome to MicroStrategy on AWS email. Click **Connect**.

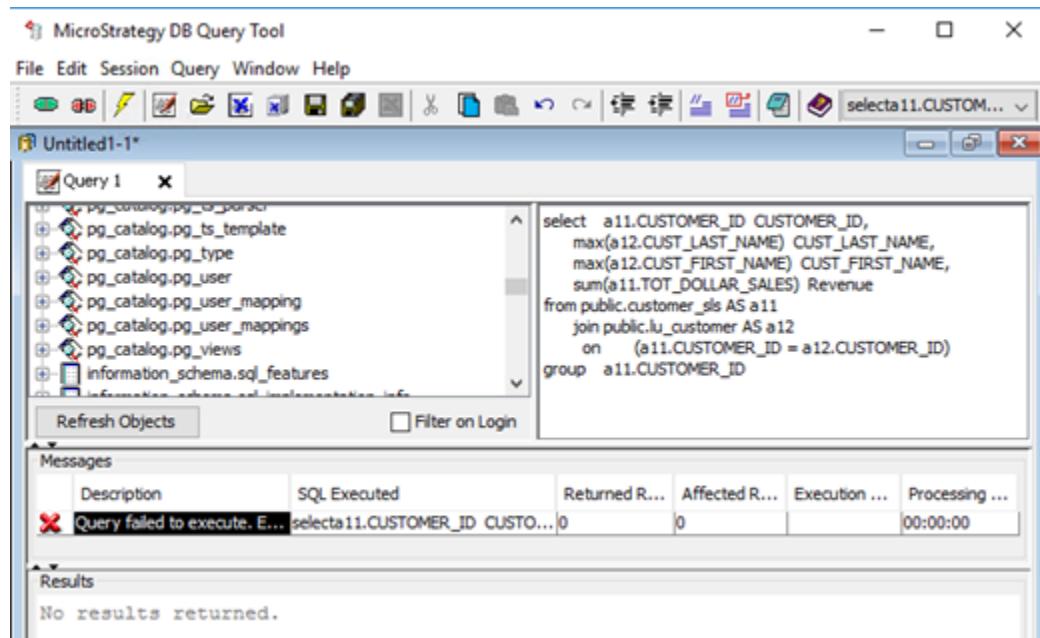


Reproduce the issue in DB Query tool

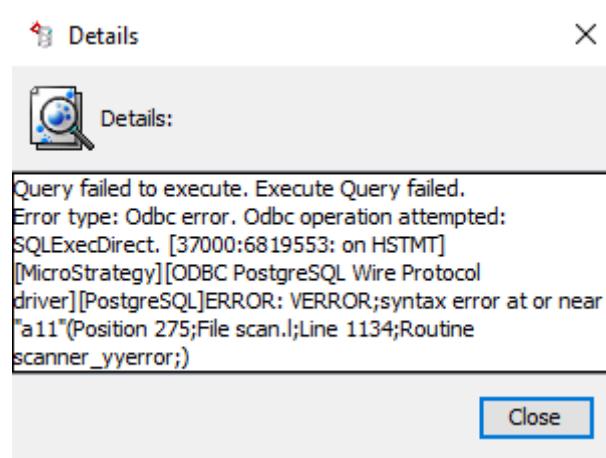
- 9 Copy the following SQL and paste it into the right pane of the DB Query tool:

```
select a11.CUSTOMER_ID CUSTOMER_ID,
       max(a12.CUST_LAST_NAME) CUST_LAST_NAME,
       max(a12.CUST_FIRST_NAME) CUST_FIRST_NAME,
       sum(a11.TOT_DOLLAR_SALES) Revenue
  from public.customer_sls AS a11
 join public.lu_customer AS a12
    on (a11.CUSTOMER_ID = a12.CUSTOMER_ID)
 group a11.CUSTOMER_ID
```

10 Execute the query. You will get the following error:



11 In the Messages pane, double-click the message to see the details:



What can you deduce from the message and how can you fix the issue?

12 The message indicates that there is a syntax error in the last line of your SQL.

To resolve it, type **by** after group, and re-execute the SQL. The error is resolved and the tool brings back results:

The screenshot shows the MicroStrategy DB Query Tool interface. The SQL pane contains the following query:

```
select a11.CUSTOMER_ID CUSTOMER_ID,
       max(a12.CUST_LAST_NAME) CUST_LAST_NAME,
       max(a12.CUST_FIRST_NAME) CUST_FIRST_NAME,
       sum(a11.TOT_DOLLAR_SALES) Revenue
  from public.customer_ids AS a11
  join public.lu_customer AS a12
    on (a11.CUSTOMER_ID = a12.CUSTOMER_ID)
 group by a11.CUSTOMER_ID
```

The Results pane displays the following data:

customer_id	cust_last_name	cust_first_name	revenue
6114	Sigman	Del	2107.5
4790	Harkema	Jodi	1315.70000...
273	Klein	Charles	4355.39999...
3936	Hayden	Sidney	4814.10000...
5761	Botelho	Gardner	3406.30000...
5697	Billingsley	Christie	4785
4321	Alphandary	Ofelia	1483.25

The Messages pane shows a successful execution message:

Description	SQL Executed	Returned Rows	Affected Rows	Execution Time	Session
Query executed correctly.	selecta11.CUSTOMER_ID CUSTOMER_ID...	10000	0	0:00	00:00

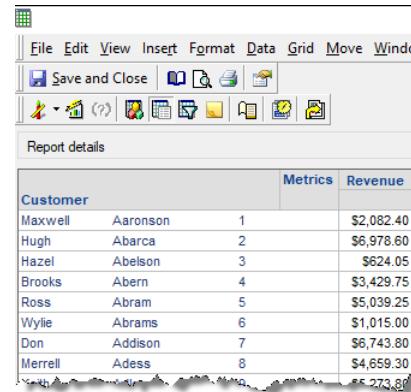
At the bottom, the status bar indicates "Task(s) has run successfully..." and "Execution Time: 00:00:01".

Re-run Freeform SQL report in Developer

13 Return to Developer and in the MicroStrategy Tutorial project, access the **PA_FreeForm** report in Design View.

14 On the **Data** menu, select **Freeform SQL Definition**. Then in the SQL pane, add **by** after **group** in the last line of your Freeform SQL.

15 Close the editor and save your Freeform SQL report. Then, save and run the grid report. The report now returns data without any error:



Customer	Metrics	Revenue	
Maxwell	Aaronson	1	\$2,082.40
Hugh	Abarca	2	\$6,978.60
Hazel	Abelson	3	\$624.05
Brooks	Abern	4	\$3,429.75
Ross	Abram	5	\$5,039.25
Wylie	Abrams	6	\$1,015.00
Don	Addison	7	\$6,743.80
Merrell	Adess	8	\$4,659.30
			\$5,273.80

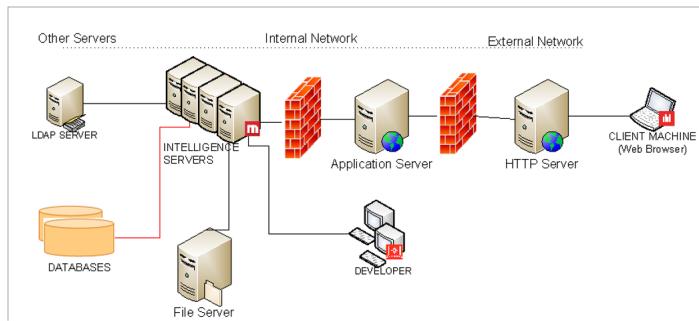
PLATFORM OPTIMIZATIONS

An Intelligent Enterprise drives adoption and the success of enterprise business intelligence. The Platform Administrator is responsible for driving initiatives that advance and enhance the analytics platform user experience to enable intelligence everywhere for the stakeholders.

Efficiently delivering data enables users to have ready access to the information they need to analyze. The better performance you can provide, the more beneficial your system is to users. It is important to configure your analytics platform in a manner that optimizes performance. When users do not have to query the data sources as often, when you can automate reports, or when you can make more processing power available to users to do their jobs, you achieve performance gains.

A four-tier MicroStrategy implementation consists of various hardware and software components, including the web and mobile servers, Intelligence Server,

ODBC drivers, networks, operating systems, database servers, client machines, and web browsers.



The performance of each of these components impacts the overall system performance in a four-tier environment. Consequently, when implementing strategies for improving overall system performance, the Platform Administrator should consider each component. In addition, the Platform Administrator should also coordinate with application designers and architects as their design of project, reports, documents, and other objects also impacts system performance.

In this chapter, we review:

- Optimizing environments: reducing computational distance
- Implementing for performance: product placement considerations
- Optimizing systems for performance: server lifecycle management
- Optimizing web experience: tuning MicroStrategy Web and application servers
- Tuning Intelligence Server for performance

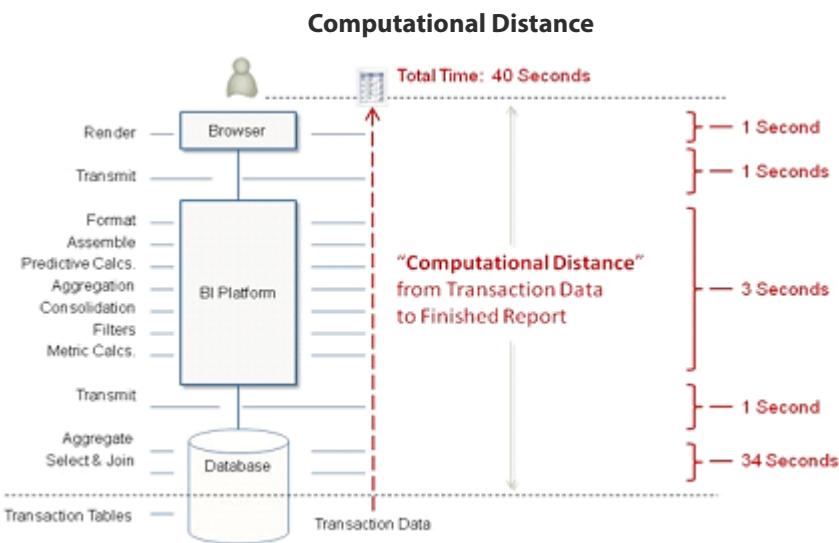
Optimizing environments: Reducing computational distance

Performance is a product of multiple factors which the Platform Administrator should review and optimize to increase user adoption. To achieve optimal enterprise application performance, pay attention to each workflow layer.

At its core, any business intelligence system consists of a series of processes and tools that take raw data at the transactional level in a database and use varying technologies to analyze and transform that data from its raw state up to the finished answer the user needs. At every step along the way some kind of work is

done, either on the database, on the network, on the BI platform, or on the client (such as browser or mobile application).

To optimize system performance, you need to reduce computational distance which refers to the length of time in terms of systems, transformations, and other processes that the data must undergo from its lowest level all the way to being rendered on a browser. The longer the computational distance is for a given report, the longer it will take to execute and render.



The preceding image shows a hypothetical example of a report that runs for 40 seconds before it displays to the end user in a four-tier web environment. Each processing step on that report such as aggregation, formatting, and rendering adds to the report's computational distance and thus to the report's overall execution time.

The Platform Administrator should establish performance optimization standards to reduce computational distance (using options such as caching, cubes, and scheduling) to provide the best-in-class analytics environments.

Implementing for performance: Product placement considerations

When implementing the MicroStrategy platform, in coordination with the System Administrator, the Platform Administrator needs to make important decisions regarding the placement of various components, such as the following:

- Whether to install the metadata and the data source on the same machine or separate machines

- Whether to install MicroStrategy Intelligence Server and MicroStrategy Web on the same machine or separate machines
- How many web servers you need for each Intelligence Server machine
- Where to physically locate various component machines from a network configuration perspective

Best Practice

There are several possible arrangements for the various products available in the MicroStrategy platform. What you decide to do depends largely on your particular environment and requirements. The following guidelines can help you make placement decisions:

- **Metadata and data source placement**—In a MicroStrategy analytics environment, the metadata is used not only for mapping the logical model to the physical warehouse schema, but also for storing all the application, configuration and schema objects that are created in your environment.

Similarly, a data warehouse and other large data sources (such as Hadoop) are used for storing your analytical data. As Intelligence Server requires access to both the metadata and data source for processing a large number of user requests, it is a good practice to install the metadata and the data source on separate machines in production environments. Such a configuration is beneficial from a network traffic perspective as metadata and data source requests are transmitted to different machines rather than to a single machine hosting both the databases.

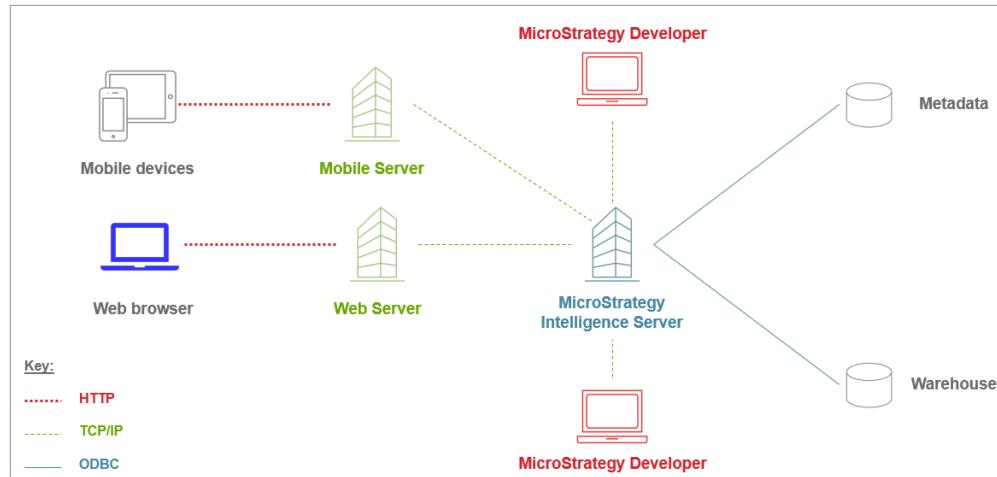
Although you may start out with a relatively small project size, as more users get added and more objects get created, the size of the metadata grows over time. Similarly, the size of your data warehouse and other data sources can grow as more and more data gets added. Therefore, the metadata should be installed on a dedicated machine in production environments. Similarly, the data warehouse or any other large data source should be installed on a separate machine. Use of dedicated machines for the metadata and the data sources provides greater flexibility in terms of metadata and data source scalability. It enables the data source to have more system resources reserved for itself to service resource-intensive requests.



In development and test environments, you may place the metadata and the data source on the same machine, if having them on separate machines is not feasible.

- **Intelligence Server and MicroStrategy Web placement**—As the Intelligence Server performs a majority of resource-intensive functionality, including tasks such as SQL generation, analytical processing, query

execution, and cache management, in many instances it should be installed on a dedicated machine in production environments.



- **Network configuration recommendations**—To minimize delays in data transmission between various components in your MicroStrategy analytics environment, it is a good practice to do the following :
 - Locate the MicroStrategy Web server physically close to the Intelligence Server machine
 - Locate the machine that has the metadata repository physically close to the Intelligence Server machine
 - Place all server components in the same network segment to minimize latency between the Intelligence Server, web server, mobile server, data warehouse server, and metadata server
 - Expand bandwidth and minimize latency between servers and clients to ensure that bottlenecks do not occur when data is requested over your network
 - Use HTTP compression between the web servers and web clients to achieve maximum throughput between these components
 - Configure a web proxy server to handle caching and reduce the load on the web server

Machines that have a great deal of traffic traveling between them experience better network performance as the physical distance between the machines is reduced. By locating the machines as close as possible to each other, you avoid reducing system performance that could result if there are a large number of network "hops" between the machines. The exception to this rule is for the disaster recovery servers. It is a good practice to have the disaster

recovery servers in separate datacenters, geographically located far away from the production servers.



The available network bandwidth is also impacted by the number of concurrent users and the user activity. For example, as the number of concurrent users and the size of reports exported increases, the amount of available network bandwidth decreases, thereby impacting the system performance. You can use various governing settings to manage network bandwidth usage associated with the number of concurrent users and user activity.

Optimizing systems for performance: Server lifecycle management

As enterprises become more dependent on analytics applications, it is imperative for the Platform Administrator to leverage consistent hardware lifecycle best practices for a coordinated implementation and continual upgrades.

As such, the Platform Administrator works with the System Administrator, architects, and other Intelligence Center architects to follow server lifecycle best practices, including the procurement, installation, maintenance, and upgrades of hardware, software, and network resources that a server needs to perform successfully.

System optimization: Server requirements

An Intelligent Enterprise continually optimizes hardware capacity to maximize and take advantage of the MicroStrategy platform. The following factors play an important role in Enterprise Architecture:

- Processor (CPU/core) type, speed and number
- Operating system type and version
- Service upgrades
- File system and disk space
- System memory - physical memory and page file (swap memory)
- Network bandwidth and speed

The Platform Administrator should work with the System Administrator to implement and continually upgrade enterprise hardware to make the analytics platform deployment a successful one and continue to engage users.

Server deployment and upgrades

An Intelligent Enterprise successfully coordinates their BI implementations, and the Intelligence Server, Web server, and Mobile Server are a key part of this effort. To ensure that the enterprise has a smooth server installation and subsequent version upgrades, you should work with the System Administrator and Intelligence Center architects to deploy and plan upgrades of these servers.

Installation and version upgrade

As the Platform Administrator, you will oversee the installation and configuration processes of the various components of the MicroStrategy analytics platform.

Best Practice

For upgrades, you need to be aware that if you created any of the following, they are not automatically transferred during the upgrade:

- Custom web plugins
- Mobile device configurations
- Any saved caches, cubes, or history list messages
- Images from the MicroStrategy Photo Uploader widget

If you created any of the preceding objects, you should always manually back up before the upgrade and then restore them after the upgrade.



For system requirements and specific instructions to install or upgrade MicroStrategy, see the *MicroStrategy Installation and Configuration Guide* and the *MicroStrategy Upgrade Guide*.

Reviewing upgrade components

The Platform Administrator is responsible for reviewing all components of the MicroStrategy analytics platform after an upgrade is complete. The Platform Administrator should:

- 1 Review and update the analytics platform connectivity, diagnostics, statistics, and security to facilitate the communication between various components of the MicroStrategy platform.
- 2 Test reports, documents, and dossiers using tools such as Integrity Manager to ensure they are returning the correct data.

- 3 Restore any web custom plugins, caches, cubes, history list messages, images, and mobile configurations from previous environment to the upgraded environment.
- 4 In every MicroStrategy version, user privileges may change because of new functionality. These changes can impact the privileges associated with the security roles, user groups, and users. As a result, you should test for all security-related settings after an upgrade.

Optimizing web performance: Tuning MicroStrategy Web and application servers

Web, mobile, and application servers are other influencers of performance in a four-tier environment. As the Platform Administrator, you want to tune these servers as part of your strategy to optimize overall system performance.

 This section focuses on the web and application server tuning. For MicroStrategy Mobile Server tuning, refer to the Mobile Architect course.

Use the guidelines below to decide what the Platform Administrator should consider for the web and application server tuning.

- **Increasing the Java heap size**—Typically, you should increase the Java heap size to avoid Out of Memory errors. Your ideal heap size settings will depend on a number of factors including system specifications, user concurrency, and load type. Typically, you should configure identical values for both the minimum and the maximum Java heap size.

 Based on your application server, you can use third-party tools (such as IBM Pattern Modeling and Analysis Tool (PMAT) for WebSphere) to determine ideal Java heap size. In addition, you can refer to your third-party application server documentation for information on how to determine a satisfactory Java heap size for your environment and the steps to change it.
- **Pre-compiling JSP files**—To avoid the time taken to load the web pages in the application server when you access it for the first time, it is a good practice to precompile the Java Server Pages (JSP) files either before or after deploying the application. If you configure the application server to load all the pages in the application, when you connect to MicroStrategy Web, the pages are already loaded and the performance is better.

 Refer to your third-party application server documentation for precompiling JSP files.

Exercise 8.1: Increase Java heap size

The goal of this exercise is to show you how to increase the Java heap size for Tomcat. For Tomcat, you will modify the minimum amount of Java heap size by defining the JAVA_OPTS parameter in the catalina.sh file. In your cloud environment, this file is located in the /opt/apache/tomcat/apache-tomcat-8.043/bin/directory. You will set the minimum amount of heap size to 6144 MB (same amount as the maximum amount of heap size) using Xms parameter. As you do not have the permissions, you can exit without saving the file.



The Tomcat version and location may be different at your site.

Access WinSCP

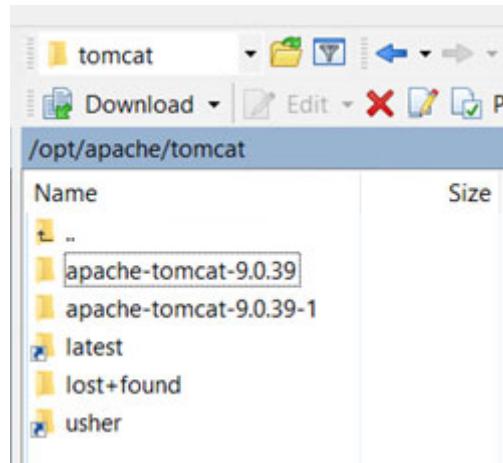
You will use WinSCP to locate the catalina.sh file. As you used WinSCP in an earlier exercise, you can just maximize and access the WinSCP window.

- 1 On your Windows desktop, access **WinSCP**.



You may need to log in again.

- 2 In the WinSCP window, in the drop down list on top of the right pane, from the root directory, browse to the **/opt/apache/tomcat/latest** directory.



- 3 Double-click **bin** to navigate to that subdirectory. Then, double-click the **setenv.sh** file.
- 4 Under the comments section, for modifying the value of the minimum amount of Java heap size, update the value of **Xms**:

JAVA_OPTS = "-Xms6144m"

A screenshot of the WinSCP terminal window. The title bar shows the path: /opt/apache/tomcat/apache-tomcat-9.0.12/bin/setenv.sh. The main area contains the contents of the setenv.sh file. The JAVA_OPTS line is highlighted with a red box: export JAVA_OPTS="-Xms3096m -Xmx3096m -XX:MaxPermSize=512m -verbose:gc +Prio". The bottom status bar shows: Line: 1/4, Column: 1, Character: 101 (0x65), Encoding: 1252 (ANSI).

As you do not have the permissions to save the file in your AWS environment, you can close the file without saving it.

Tuning Intelligence Server for performance

Another critical component that the Platform Administrator must consider for tuning is Intelligence Server. Use the guidelines below to consider some of the optimizations you can implement related to Intelligence Server:

- Enable caching
- Use Intelligent Cubes
- Use scheduling
- Configure History List
- Optimize database connections-related settings
- Use governing

Performance influencers: Caching

The biggest influencer for performance is caching — as the Platform Administrator, you want to implement and enforce a comprehensive caching strategy. Caching helps optimize performance by bypassing key bottlenecks on the server, database, and network side.

With caching, the data is retrieved from the memory of the Intelligence Server, then passed through the MicroStrategy Web or Mobile Server to the user browser or app, improving performance.

Planning a caching strategy

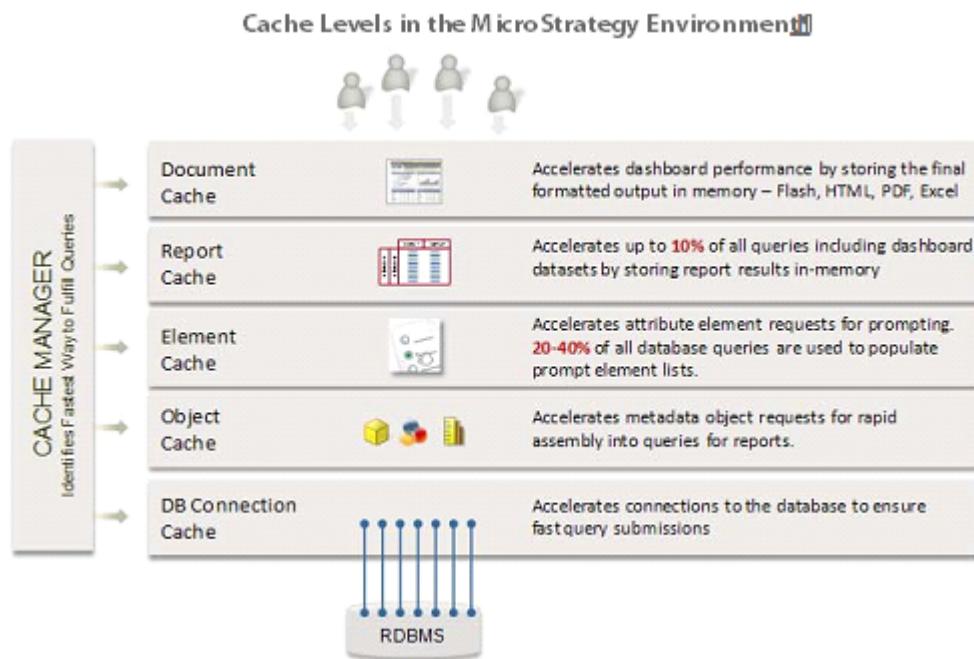
To enhance the user experience, the Platform Administrator should evaluate and determine the enterprise's caching strategy. Use the guidelines below to decide what methods are best for your environment.

Best Practice

The steps below outline what the Platform Administrator should consider and enact for an effective caching strategy, including cache best practices.

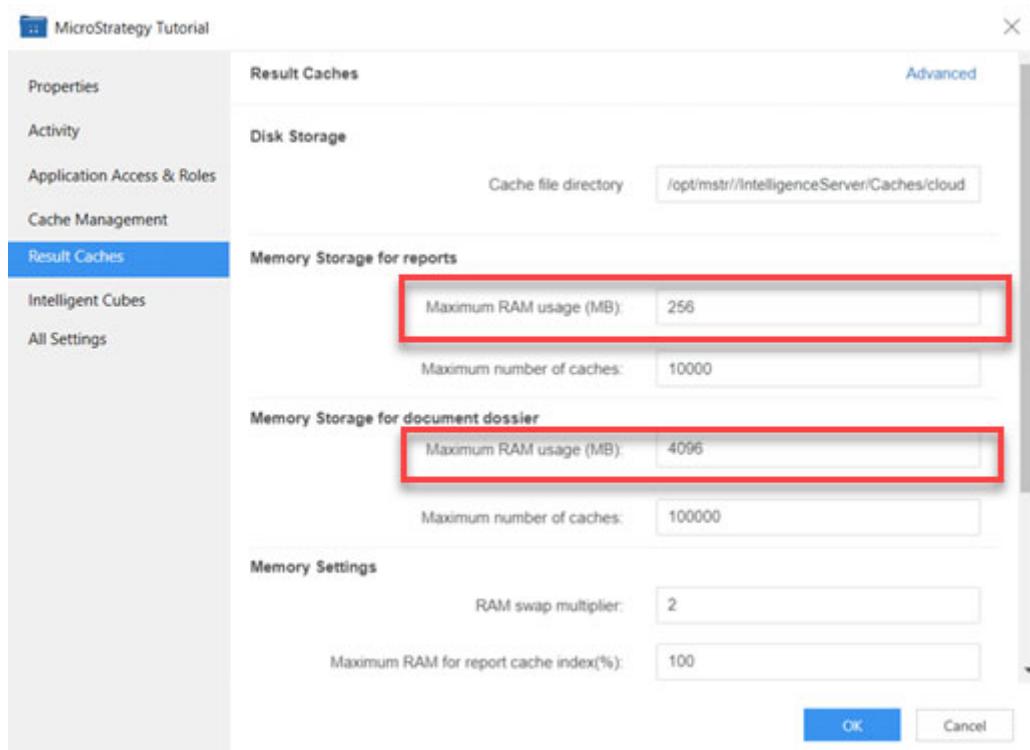
- 1 In coordination with the Intelligence Center architects, obtain a list of reports, documents, and dossiers that need to be cached.
- 2 To take full advantage of caching to improve performance, enable caching at different levels, from database connections to element caches to reports. Any

break in the workflow or availability of caches at any level can prohibit you from utilizing caches in general.

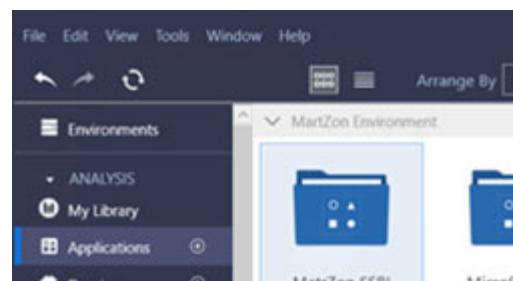


- 3 Ensure that enough memory is allocated for report and dossier caches. Insufficient memory reserved for caches causes reloading of cache files from

disk. By allocating sufficient memory, you can eliminate or minimize repeated loading of caches.

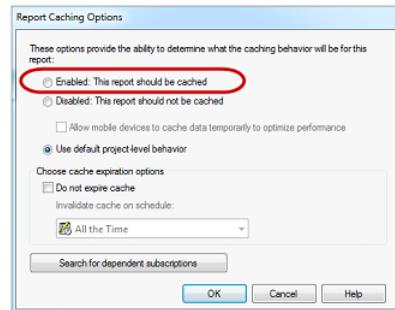


To access cache settings for applications using Workstation, navigate to **Applications**, right-click your application, then select **Properties**. In the application properties windows, select the **Result Caches** tab, then modify the value in the right pane and select **OK** to save your modification.

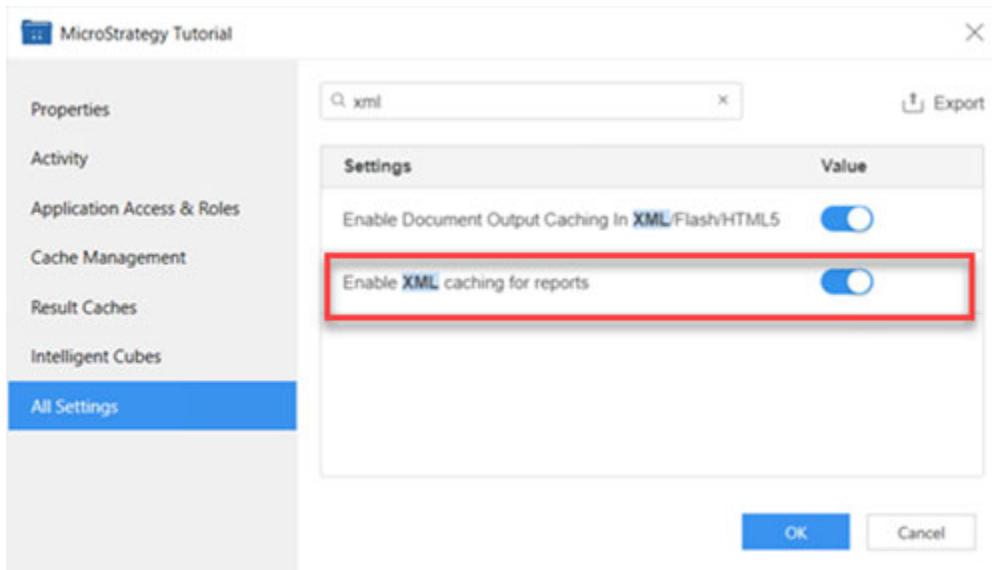


- 4 Caching allows for improved performance in response to report queries. However, creating a large amount of caches can compromise Intelligence Server resources. Document caches in particular tend to use a lot of memory space. As a result, avoid creating unnecessary caches that will get stored in memory without being hit by any execution.
 - Determine which reports, dossiers, and documents will be used frequently and enable caching for those objects only. Platform Analytics Cache

Telemetry Dossier provides information about how many jobs were executed, with numbers of hits for caches and Intelligent Cubes.

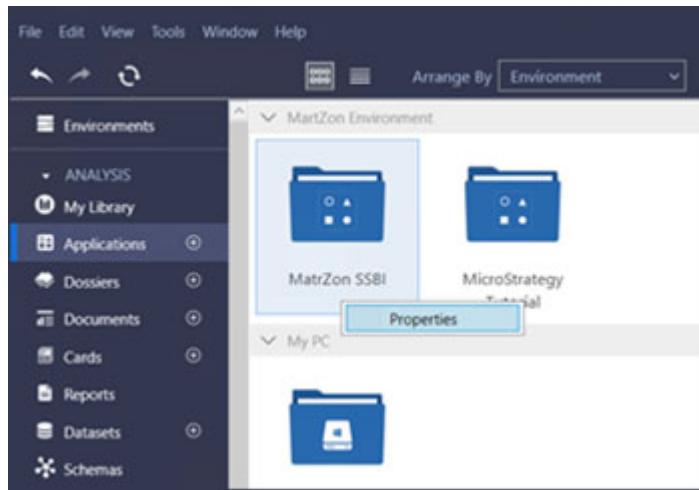


- Disable caching for highly prompted reports and documents, or prompts on attributes. Instead, for highly prompted datasets, require the use of Intelligent Cubes with prompted OLAP reports as document data sources, to increase data retrieval speed. Ensure attribute elements are cached, as this increases prompt execution speed.
- Enable XML caching for reports. XML caching stores the attributes to which all users in MicroStrategy Web can drill in the report's XML cache, resulting in faster response times.



To access XML caching for applications using Workstation, navigate to **Applications**, right-click your application, then select **Properties**. In the application properties windows, select the **All Settings** tab, type **XML** in the

search box, then modify the value in the right pane and select **OK** to save your modification.



- 5 Report caches are stored both in the Intelligence Server memory and on a drive. For performance reasons, the drive that holds the result caches should always have at least 10% of its capacity available.
- 6 If using MicroStrategy Mobile, device caches are a must for optimal performance. Device caches can render directly from device memory with no need for interaction over the network or server and hence are the most optimal method of running apps on mobile devices.
- 7 Use Subscription Caching when personalized caches are necessary due to the dependency of prompts within the supporting datasets. The caches can be delivered to the user's History List, mobile device, or update the existing caches.
- 8 Create cache refresh schedules in collaboration with the Intelligence Center architects who can provide information on the governed (certified) datasets or dossiers (and the underlying cubes, if any) that need to be refreshed using schedules. In addition, stagger and schedule the execution of jobs for creating/refreshing caches during off-peak hours so as to avoid impacting the performance of the production environment during regular work hours.
- 9 Enact cache maintenance policies to clean and replace older versions. Effective cache management can enhance performance by ensuring Intelligence Server resources are not overloaded by unnecessary caches.

There are many techniques for removing report caches. Based on your requirements it is important to choose the right cache maintenance strategy.

The ideal strategy may be one of the following, or a combination of different strategies:

- **Invalidating report caches**—Invalidation is a preventive measure that renders a cache unusable by nullifying it. It makes the cache ineligible in the matching process so it is not used to fulfill a report request.

You should invalidate a cache typically when a set event takes place such as when your data warehouse is refreshed.

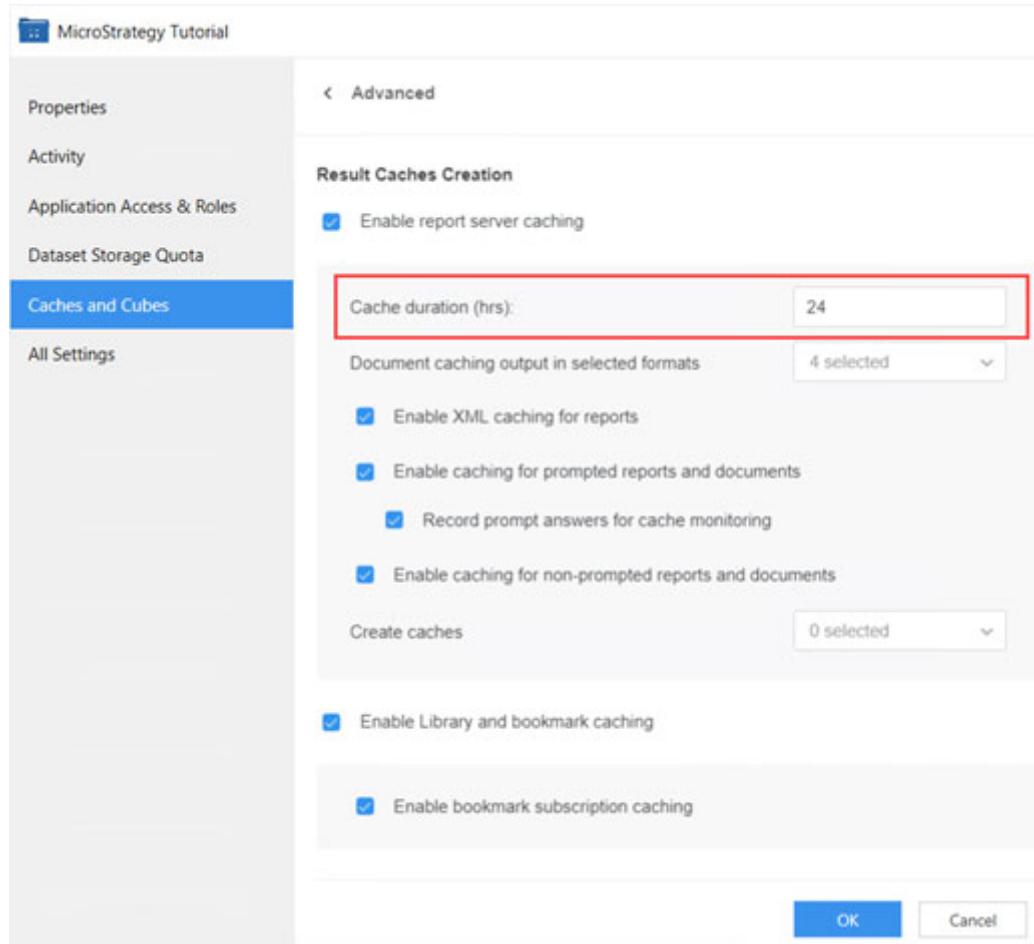
- **Expiring report caches**—Cache expiration is a process that renders a cache unusable by terminating its useful life.



Cache expiration occurs automatically as per the Cache duration (Hours) setting in the Project Configuration Editor.

When a cache is updated, the current cache lifetime is used to determine the cache expiration date based on the last update time of the cache. This means that changing the Cache duration (Hours) setting does not affect the expiration date of the already existing caches. It only affects the new caches that are being or will be processed.

Expiration of cache is typically not recommended as cache removal should be based on changes to the data in the data warehouse rather than some arbitrary time duration.



- **Deleting report caches**—Cache deletion is a process that deletes the cache from memory as well as disk. Report caches are automatically deleted by Intelligence Server if cache invalidation and History Lists are performed and maintained properly.
- **Purging report caches**—Cache purging is a process whereby all report caches can be deleted in bulk, even the caches referenced by History List messages.



The difference between purging and deleting caches is that purging automatically eliminates all caches in a project, while deletion can be performed for individual report caches.

Regardless of the cache removal option used, you can save time by automating the deletion of report caches according to a specific schedule or set of guidelines.

Now that you've created your caching strategy per the guidelines, let's execute these methods to improve analytics platform performance.

Exercise 8.2: Configure subscription caching to History List

The Category Management Dashboard has a region prompt so that managers only view data for their region. Per your caching guidelines, when reports, dossiers, or documents use prompts or security filters, it is necessary to use subscription caching. This way, caches are delivered to the user History List.

In the exercise below, you will create a History List subscription in Developer. You will run the subscription immediately as well as whenever the On Database Load event is triggered. After creating the subscription, you will view it using Subscriptions Manager.

Finally, while logged into Developer as the mstr user, you will verify the creation of the message in the History folder.

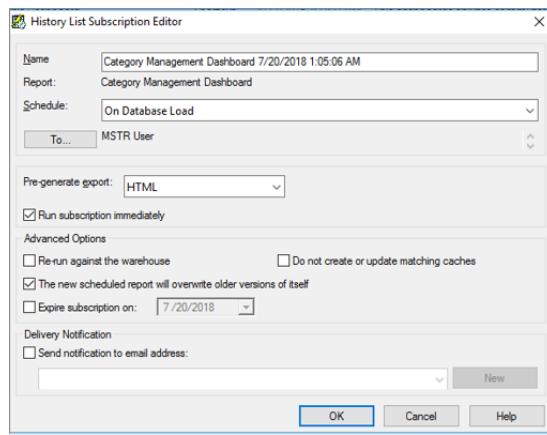
Access the History List Subscription Editor

- 1 In Developer, in the MicroStrategy Tutorial project, access the **Public Objects\Reports\Documents and Scorecards** folder.
- 2 In the Documents and Scorecards folder, right-click the **Category Management Dashboard**.

The Category Management Dashboard has a large dataset and takes time to load. To help with loading time, you want to set up History List caching.
- 3 Hover over **Schedule Delivery To**, then click **History List**. The History List Subscription Editor displays.
- 4 From the **Schedule** drop-down list, you can select a schedule to control how often the subscription occurs. You want the schedule delivery to History List upon database refresh. Click the **Schedule** drop-down, and select **On Database Load**.

You can create new schedules in the MicroStrategy Developer Schedule Manager. For steps, see Scheduling Jobs and Administrative Tasks in the System Administration Guide.

- 5 By default, you are the recipient for the subscription. To add additional recipients or to remove yourself from the subscription, click **To**. The Recipients Browser window opens. Select the users and groups that you want to receive the subscription and click **OK**. For this exercise, leave yourself (**mstr**) as the recipient.
- 6 To execute the report or document immediately when the subscription is saved, select **Run subscription immediately**.
- 7 To send a notification when the subscription executes and the cache is updated, you can select the **Send notification to email address** check box. For this exercise, leave the check box cleared.



- 8 Click **OK**. Your subscription should execute immediately.

View the subscription in Subscription Manager

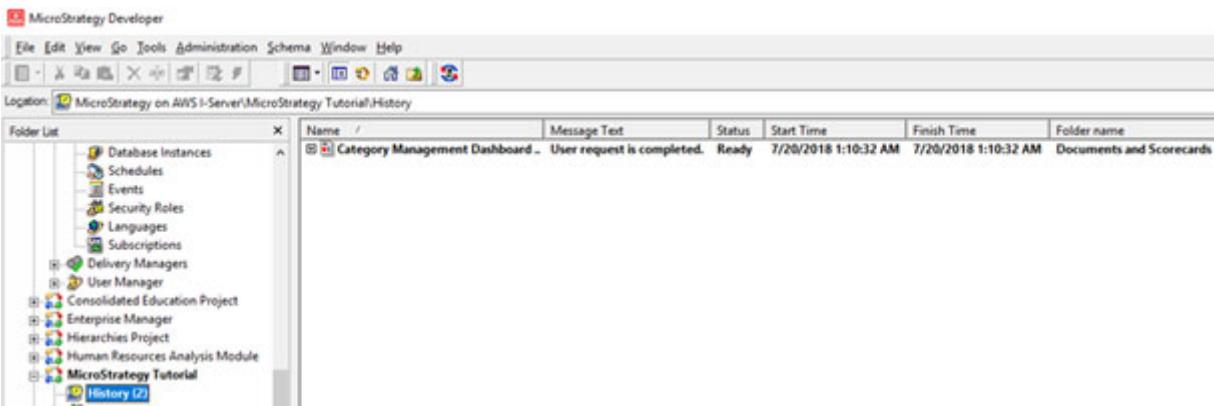
- 9 In Developer, expand **Administration** in the Folder List.
- 10 Expand **Configuration Managers**, and select **Subscriptions**.

You should be able to see your subscription in the pane on the right. If you need to make any changes to your subscription, you can double-click it and make the changes in the History List Subscription Editor.

Verify History List message in the History folder

- 11 In the MicroStrategy Tutorial project, select the **History** folder.

You should see the message. If any other users were also subscribed to this document, those users can also see the message in their respective History folder in Developer or under History List in MicroStrategy Web.

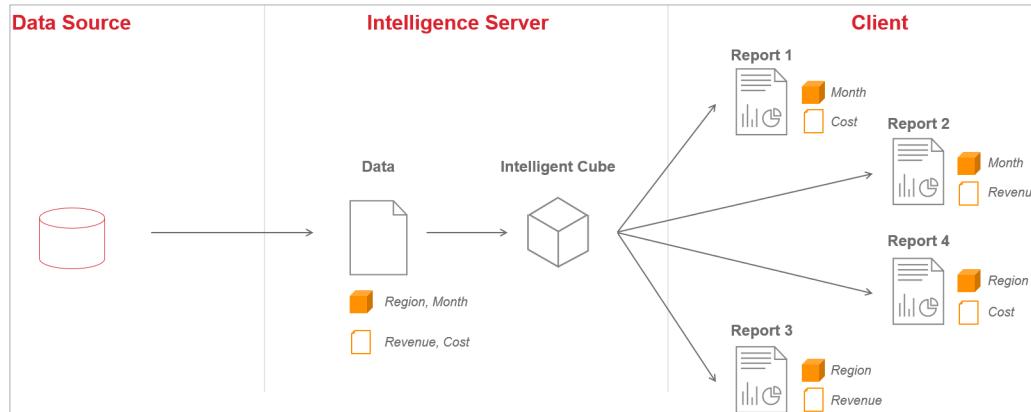


Performance influences: Intelligent Cubes

While caching has a big impact on performance is caching, it may not always be feasible to create caches for every single report. To create caches, each report has to first run against the data warehouse. In addition, many of the reports may return overlapping data, and therefore many caches may contain redundant data.

As the Platform Administrator, you need to implement a comprehensive, multi-pronged platform optimization strategies. One of the strategies to overcome the challenges of caching is to utilize Intelligent Cubes.

Intelligent Cubes allow multiple reports to retrieve data from a single shared in-memory set of data for faster response times. Intelligent Cubes act as a layer between your data source and MicroStrategy reports that analyze and display data, as shown below:



Intelligent Cubes are fully scalable, limiting excessive data consumption and redundant data by allowing you to build only the sets of data you require.

Define thresholds for memory consumption

Depending on the amount of data to be retrieved, the size of an Intelligent Cube can be very large. To strike a balance between user demands and available memory resources, it is important for the Platform Administrator to coordinate with Intelligence Center architects, developers, and System Administrator and establish memory thresholds for the cubes that are stored in memory.

As memory consumption increases, data analysts may experience a negative impact on performance. To alleviate memory consumption problems, establish guidelines to help developers and platform administrators define thresholds for memory consumption. For example, based on your environment and in coordination with the Intelligence Center developers, architects, and system administrators, the Platform Administrator should set guidelines for the following:

- The maximum cube size that can be created by individual users
- The maximum number of cubes that can be created in a project. By default, Intelligence Server is configured to store 1,000 cubes.

Best Practice



If an attempt to load an Intelligent Cube is made that would exceed the limit, an Intelligent Cube is removed from Intelligence Server memory before the new Intelligent Cube is loaded into memory.

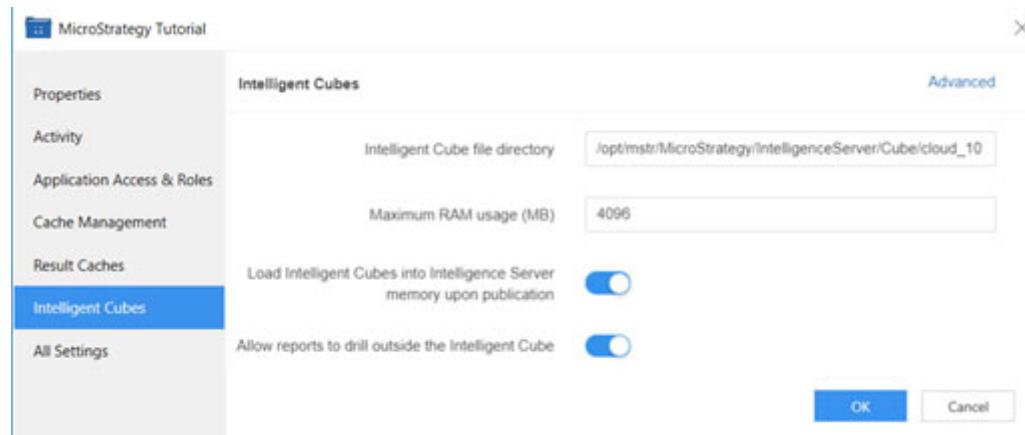
- The amount of Intelligence Server RAM that can be used for storing cubes
- The maximum cube size, in megabytes, that can be downloaded from Intelligent Server
- Whether to create Intelligent Cubes by database connection. If using connection mapping, you should define your Intelligent Cubes to use and support connection mapping to ensure users are able to access data they authorized to access.
- Whether to load cubes on Intelligence Server startup. If the cubes are configured to load on Intelligence Server startup, then the reports accessing Intelligent Cubes are executed faster as the cube for the report has already been loaded. However, the overhead experienced during Intelligence Server startup is increased because of the processing of loading cubes.



It is generally a good practice to disable loading of Intelligent Cubes on startup unless a project uses only a small number of cubes. Restarting the Intelligence Servers is not a very common operation in production environments and it is preferable to have the Intelligence Server start as fast as possible.

- Whether to allow reports to drill outside the Intelligent Cube: By enabling drilling outside an Intelligent Cube, reports that access the Intelligent Cube have ROLAP access to data in the warehouse through drilling. While this extends the analysis and data access capabilities of reports that access Intelligent Cubes, drilling outside an Intelligent Cube can put additional load on the Intelligence Server and data warehouse. This is because drilling outside an Intelligent Cube requires a new report to be executed against the data warehouse.
- Whether to load Intelligent Cubes into Intelligence Server memory upon publication. It is generally a good practice to load Intelligent Cubes into Intelligence Server memory upon publication as it results in faster performance. If you do not load Intelligent Cubes into Intelligence Server memory upon publication, the Intelligent Cubes will be stored only in secondary storage when published. The Intelligent Cube can then be loaded into Intelligence Server memory manually, using schedules, or whenever a report attempts to access the Intelligent Cube.

Best Practice



To access Intelligent Cube setting for applications using Workstation, navigate to **Applications**, right-click your application, then select **Properties**. In the application properties windows, select the **Intelligent Cube** tab, then modify the values in the right pane and select **OK** to save your modification.

To access Dynamic Sourcing settings, click the Advanced link on the top right.

Best Practice

Best practices for Intelligent Cubes management

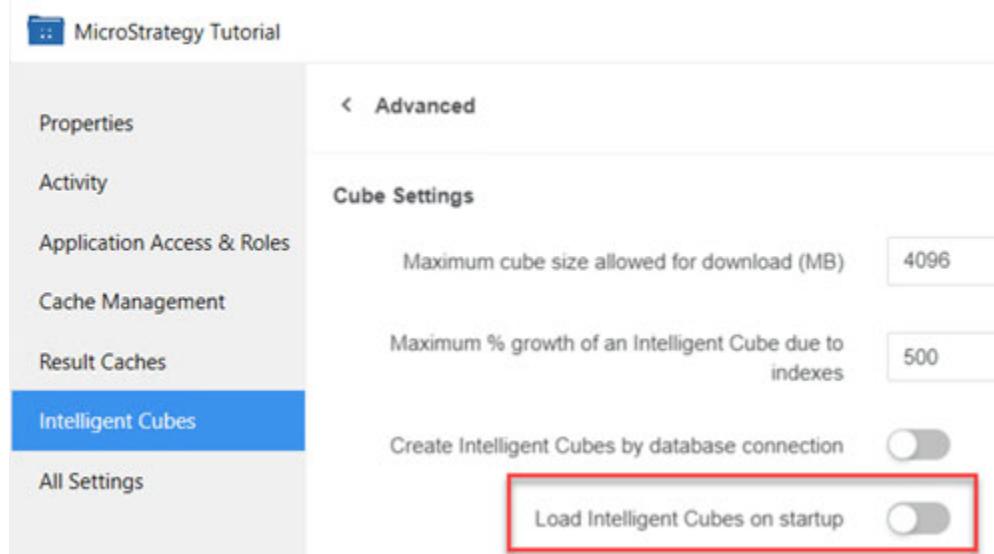
- 1 When possible, run reports, documents, and dossiers against an Intelligent Cube to increase data retrieval speed. If the cube is large, use prompted reports that access the cube to reduce the columns and rows being sent to the user.

- 2 Create cube refresh schedules in collaboration with the Intelligence Center architects who can provide information on the governed (certified) datasets or dossiers (and the underlying cubes, if any) that need to be refreshed using schedules. In addition, stagger and schedule the execution of jobs for creating/refreshing cubes during off-peak hours so as to avoid impacting the performance of the production environment.
- 3 During the cube publication process, the construction of a cube requires more memory than will ultimately be used by a fully published cube. The peak memory used by Intelligence Server is expected to be two to five times the cube size while publishing a cube. This happens because of the data processing that the Analytical Engine performs to create the necessary data structures to support the cube.

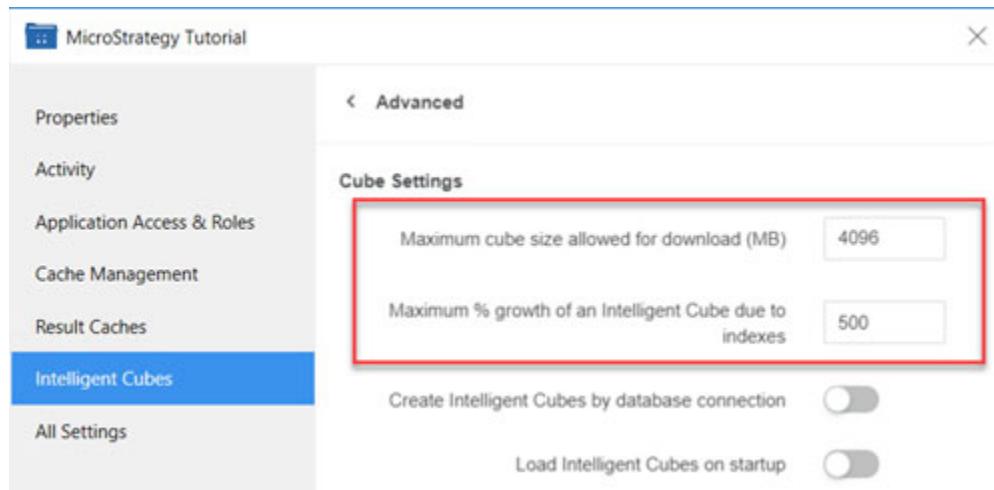
This is an important factor to consider when calculating how much server memory might be needed to create Intelligent Cubes to support a set of reports and documents. Exceeding the available memory in a server during cube publication does not cause the process to fail but it does slow the process down. The reason for this is that the operating system will use memory swapping to disk, which severely impacts cube publication times.

- 4 If you want to prevent reports that access cubes from loading them to memory, you can either deactivate or delete a cube. You should deactivate a cube if you want to disable it for a short time, and will later enable it again. However, delete the cube if it contains obsolete data or will not use it again.
- 5 By default, cubes load when Intelligence Server starts. Loading a cube can require memory resources that are more than the size of the cube, which can affect performance. Therefore, it is a good practice to disable the Load

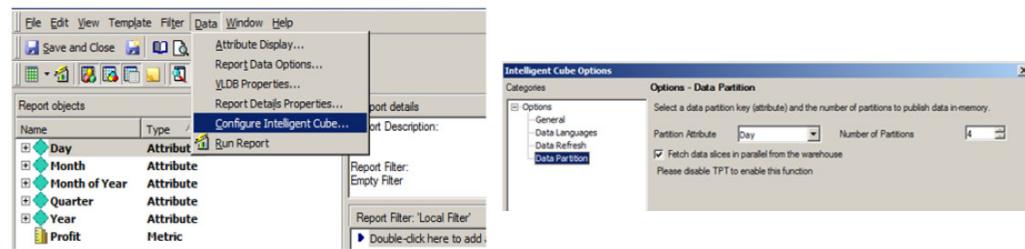
Intelligent Cubes on startup setting. When a report accessing an active but unloaded cube is executed, that cube is automatically loaded into memory.



- 6 While cubes can improve performance of reports by storing data in the memory of Intelligence Server, loading too much data in memory may cause a negative impact on job execution. For this reason, it is important to manage and govern the amount of cube data to store in the memory of Intelligence Server.



- 7 For loading data into memory in parallel, configure the Intelligent Cube for parallel execution using the Intelligent Cube Editor.

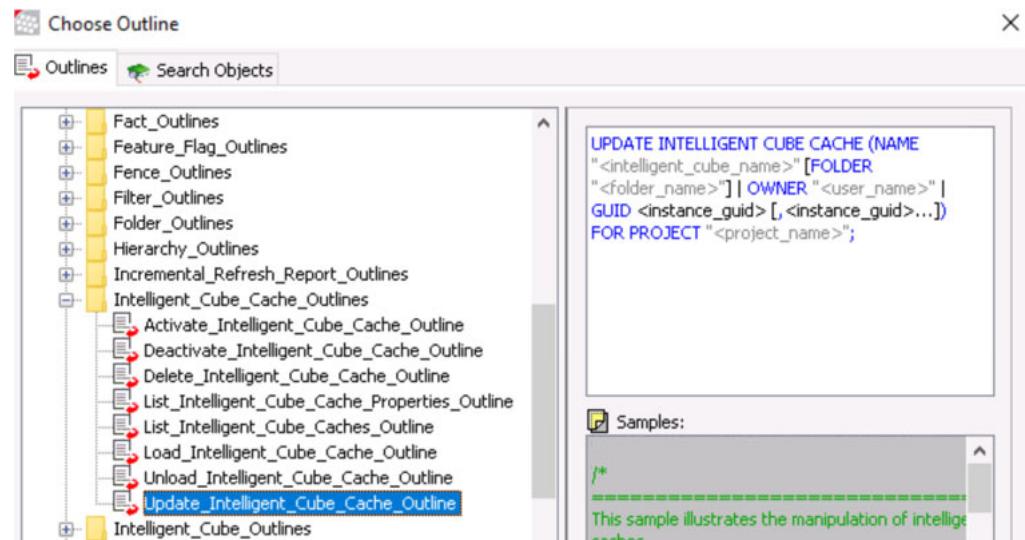


Exercise 8.3: Automate cube refresh using Command Manager

In this exercise, you will use Command Manager to refresh cache of Intelligent Cube - Geography located in the Public Objects\Reports\MicroStrategy Platform Capabilities\MicroStrategy OLAP Services\Intelligent Cubes and View Reports folder.

Access Command Manager

- 1 In Command Manager, on the toolbar, click **Insert Outline**.
- 2 In the Choose Outline window, expand the **Intelligent_Cube_Cache_Outlines** folder and select **Update_Intelligent_Cube_Cache_Outline**. Click **Insert**.



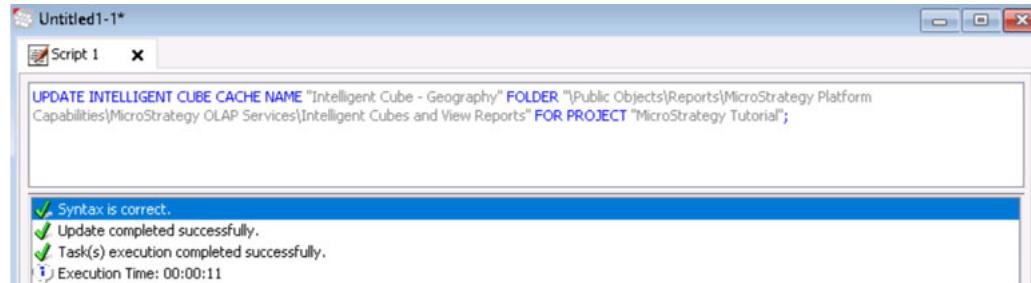
- 3 Customize the syntax to the following:

```
UPDATE INTELLIGENT CUBE CACHE NAME "Intelligent Cube - Geography"
FOLDER "\Public Objects\Reports\MicroStrategy Platform Capabilities\
MicroStrategy OLAP Services\Intelligent Cubes and View Reports" FOR
PROJECT "MicroStrategy Tutorial";
```

Check syntax and run the script

- 4 On the toolbar, click **Check Syntax**. If an error message displays, correct the syntax of the script and then re-check it.

- 5** Execute the script. A message displays on the Messages tab specifying that the various objects have been created.



- 6** Save the script to **C:\MSTR** and name it **CubeRefresh**.

- 7** Exit Command Manager.

Verify cube refresh using Caches Monitor

- 8** In Developer, access the Caches Monitor. Check the cube information and verify with timestamp that the cube was successfully refreshed.

The screenshot shows the MicroStrategy Developer interface with the title bar "MicroStrategy on AWS I-Server". On the left, a navigation tree under "Administration" shows "System Administration" expanded, with "Caches" selected. A red box highlights the "Intelligent Cubes" node. To the right is a table titled "Intelligent Cube Report Name /". The table lists various cubes with their project names, statuses, last update times, hit counts, and sizes. One entry for "Intelligent Cube - Geography" is highlighted with a red box and has a red arrow pointing from the "Last Update Time" column towards the timestamp "7/17/2018 8:00:10 PM". The bottom status bar shows "MicroStrategy on AWS I-Server Server Online" and the date "7/17/2018".

Project Name	Status	Last Update Time	Hit Count	Size (KB)
MicroStrategy Tutorial	A, L, F	10/26/2017 5:19:20 PM	0	76738
MicroStrategy Tutorial	A, L, F	10/26/2017 5:18:51 PM	0	590
MicroStrategy Tutorial	A, F	2/6/2017 5:32:50 PM	0	7304
MicroStrategy Tutorial	A, F	2/6/2017 5:29:27 PM	0	1029
MicroStrategy Tutorial	A, L, F	8/4/2016 6:34:27 PM	0	571
MicroStrategy Tutorial	A, F	2/6/2017 8:02:18 PM	0	1864
MicroStrategy Tutorial	A, L, F	8/3/2016 6:53:56 PM	0	2561
MicroStrategy Tutorial	A, F	8/4/2016 1:23:07 PM	0	970
MicroStrategy Tutorial	A, F	8/3/2016 6:55:05 PM	0	6398
MicroStrategy Tutorial	A, L, F	7/17/2018 8:00:10 PM	0	1586
MicroStrategy Tutorial	A, F	8/3/2016 6:53:42 PM	0	1421
MicroStrategy Tutorial	A, F	8/3/2016 6:51:13 PM	0	3245
MicroStrategy Tutorial	A, L, F	8/3/2016 6:48:54 PM	0	1104
MicroStrategy Tutorial	A, F	8/4/2016 1:54:57 PM	0	16900
MicroStrategy Tutorial	A, F	2/6/2017 5:29:27 PM	0	1353
MicroStrategy Tutorial	A, L, F	10/29/2017 6:00:36 PM	0	1388
MicroStrategy Tutorial	A, L, F	2/2/2018 4:33:13 PM	0	326
MicroStrategy Tutorial	A, L, F	2/2/2018 4:33:11 PM	0	326
MicroStrategy Tutorial	A, L, F	2/2/2018 9:16:59 PM	0	310
MicroStrategy Tutorial	A, L, F	2/2/2018 9:16:59 PM	0	721
MicroStrategy Tutorial	A, L, F	2/2/2018 9:17:00 PM	0	575
MicroStrategy Tutorial	A, F	2/6/2017 5:32:24 PM	0	351
MicroStrategy Tutorial	A, L, F	10/26/2017 5:18:51 PM	0	590

Performance influencers: Scheduling

Another optimization techniques that the Platform Administrator should consider is scheduling. Scheduling enables you to execute jobs, create caches, and update cubes during off-peak hours. As the caches have already been created, it results in faster response times when users subsequently execute the jobs.

As the Platform Administrator, you should also be aware that MicroStrategy Scheduler opens a different session for each user that has a scheduled request. Since all reports tied to a schedule are submitted at one time to a single Intelligence Server, MicroStrategy Scheduler has the potential to introduce a large

amount of stress onto the system, depending on the size and the analytic complexity of each report. Even in a clustered environment, when a schedule is initiated, the single, primary server node of the cluster runs all the reports that are tied to the time-based schedule, while the event-based schedules are executed on the node the event is triggered.

Best Practice

The Platform Administrator can use the following guidelines for implementing a memory-efficient scheduling strategy:

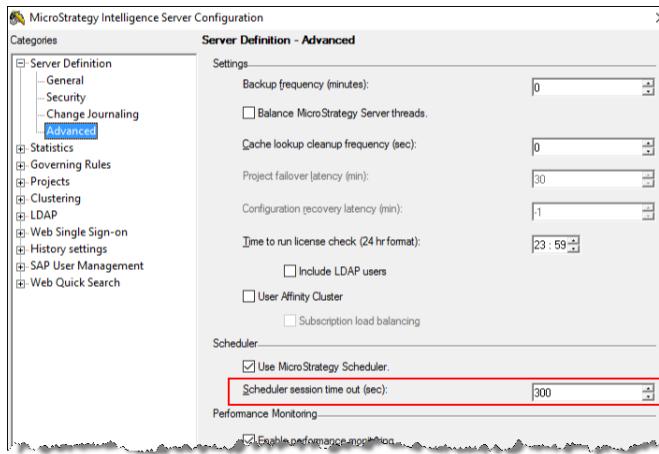
- When possible, schedule execution of reports, documents, and dossiers during off hours. In addition, these objects should be associated with different schedules, with an appropriate time interval between each schedule. By staggering scheduled jobs, you minimize the load on Intelligence Server at any given instant, while spreading the load over a longer duration.
- You can also use event based scheduling in a clustered environment. Instead of triggering the event on one node, you can associate reports with different event based schedules and trigger the events on different nodes. By triggering events on different nodes, you distribute the load among multiple Intelligence Server nodes.

This also ensures that other event based schedules still get executed even if a specific event does not get triggered due to the failure of a node on which it was supposed to trigger.

- You can also configure the Scheduler session time out (sec) setting. MicroStrategy Scheduler opens a new session for each user that has scheduled a request. Just like a MicroStrategy Developer user session, a MicroStrategy Scheduler open session consumes memory. The only way a session is closed is when it reaches the Scheduler session time out (sec) limit. Otherwise, a scheduled session could remain open for a very long duration (such as in case of a scheduled prompted report for which no prompt answer is provided).

It is important that the Scheduler session time out (sec) setting not be set to unlimited (in other words, -1 or 0) on a system that is using MicroStrategy

Scheduler. You should set it for a duration that is slightly higher than the duration it takes for the longest batch of scheduled reports to run.



Performance influencers: History List

To enhance the system performance, the Platform Administrator also need to develop guidelines for History List which is an in-memory message list that points to reports a user has executed or scheduled. Use of History List enables you to perform asynchronous report execution in real time or when scheduled. Users can then view the results by accessing the History List messages.

History List messages are loaded into the Intelligence Server memory when users log in and can consume a large amount of memory if users add too many reports or documents to their History List.

Best Practice

As the Platform Administrator, you can use the following guidelines for implementing an optimized History List strategy:

- **Use hybrid storage mode**—There are several ways that the History List repository can be configured to store data for the History List. It is a good practice to use a hybrid History List repository model because this approach preserves the scalability of the database-based History List, while maintaining the improved performance of the file-based History List.



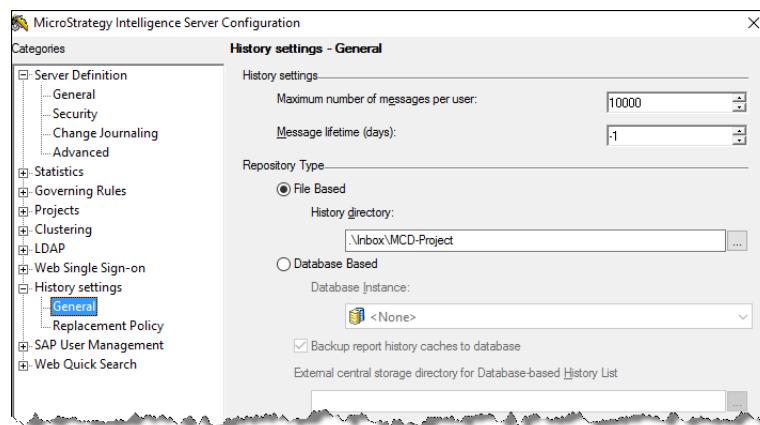
For additional details on hybrid History List, refer to the Knowledge Base article, KB44171.

- **Control History List size**—Control the size of the History List and thus control resource usage through the following settings:
 - **Maximum number of messages per user**—This setting, which can be configured both at the server and project levels, limits the number of

messages that users can have in their History List. You can set this value in the range of 10 to 30, depending on the number of projects, and increase it if necessary in your environment.

 In MicroStrategy Web the oldest message is removed when the maximum number of History List messages is reached.

- **Message lifetime (days)**—This setting controls the number of History List messages by automatically purging the History List on a periodic basis. The default value of -1 means no limit; messages stay in the system until the user deletes them. You should specify an appropriate value in this field to automatically clean up History List.



- **Do not add reports and documents automatically to History List**—As document sizes can be very large, do not add reports and documents automatically to History List in MicroStrategy Web. If needed, users can add a report or document manually to their History List.

PREFERENCES LEVEL <ul style="list-style-type: none"> • User Preferences • Project Defaults PREFERENCES <ul style="list-style-type: none"> • General • Color Palette • Folder browsing • Grid display • Graph display • History List • Export Reports • Print Reports (PDF) • Drill mode • Prompts • Report Services • Security • Project Display • Office 	History List <p>Add reports and documents to my History List:</p> <p><input type="radio"/> Automatically <input checked="" type="radio"/> Manually</p> <p>If manually, how many of the most recently run reports and documents do you want to keep available for manipulation?: <input type="text" value="10"/> Minimum Value: 3</p> <p>Note: These reports and documents will be available for manipulation even if they are not saved to the History List.</p> <p>The results of history list subscriptions get added to the History List.</p> <p><input checked="" type="checkbox"/> The new scheduled report or document will overwrite older versions of itself.</p> <p>This option is automatically enabled for users without access to the History List.</p> <p>Pre-generate export in subscriptions <input type="button" value="HTML"/></p> <p><input type="checkbox"/> Duplicate Message on Reprompt or Refresh</p>
---	---

Exercise 8.4: Automate deletion of History List messages

You should proactively delete old History List messages to reduce the load on system resources. In this exercise, you will schedule the automatic deletion of History List messages that have been read and are older than 50 days in the MicroStrategy Tutorial project.

Schedule the History List message deletion

- 1 In Developer, on the **Administration** menu, point to **Scheduling**, and select **Schedule Administration Tasks**.
- 2 In the Schedule Administration Tasks window, in the **Available projects** list, select the **MicroStrategy Tutorial** project.
- 3 In the **Choose one of the following actions** drop-down list, select **Delete History List messages**.
- 4 Select the **At Close of Business (Weekday)** schedule. Under Lifetime (days), type **50**. Next, in the Status drop-down list, select **Read**, and then click **OK**.

From now onwards, the specified History List messages will automatically be deleted on the specified scheduled.



You should schedule the deletion of History Lists when Intelligence Server is not busy, such as during off-hours. Otherwise, this maintenance task could potentially overload the server. Also, deleting History Lists for a large group places a heavy load on Intelligence Server. Therefore, instead of scheduling the deletion of History Lists for all users at one time, it is best to schedule the deletion for smaller groups staggered over a period of time, for example, each with a different schedule at 1 AM, 2 AM, and 3 AM, and so on.

Optimizing query performance: Database connections

Intelligence Server executes queries against the warehouse using database connections. If a database connection is busy executing other queries, the incoming query is placed in a queue until a database connection becomes available. As the Platform Administrator, you need to provide guidelines on the solutions your team can implement to reduce the job response time. Possible solutions include:

- **Database connection threads and processing units**—When you execute any job (such as a report or document) in the MicroStrategy environment, the Intelligence Server processes it as a collection of tasks. Each task is associated with a different activity such as prompt resolution, SQL generation, SQL execution, and so forth, and is handled by a processing unit (PU).

A PU is a container and manager of the tasks, threads, and queues.



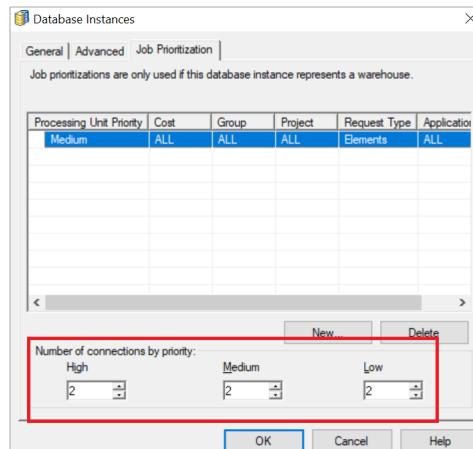
Every PU has one or more threads associated with it, with each thread executing a task that is being handled by the PU. For example, if a task requires document execution, the thread can run the document execution code. Similarly, if the task requires SQL generation, the thread processes that task. As each thread can handle only one task at a time, you can determine the number of threads as follows:

Number of threads in PU = Number of tasks that can execute concurrently in that PU

For example, when the Intelligence Server executes a query against the data warehouse, it uses database connection threads associated with the PU to execute the query. If there are not sufficient database connection threads, a job request is queued. As a result, you need to ensure that you have configured sufficient number of database connection threads to achieve the expected system response time.

Additionally, when configuring the number of database connection threads, you need to be aware that when Intelligence Server starts, it creates all database connection threads that are used for communicating with the warehouse. Each thread consumes around 1 MB of system memory whether or not it is connected to the database. While having a smaller number of threads decreases the amount of memory consumed by Intelligence Server at startup, you need to have sufficient number of threads to enable more user queries to be processed in less amount of time so as to provide good system performance. You can start with the default setting of six threads and then increase it, as necessary, based on your environment. In addition, it is good practice to use same database connections across projects as it results in more efficient Intelligence Server and database server governing. You can configure

the number of database connection threads using the Database Instances editor.



- **Jobs prioritization**—By default, jobs are usually executed as first-come, first-served. However, jobs prioritization enables you to process higher priority jobs to be executed against the warehouse, ahead of other concurrent jobs.

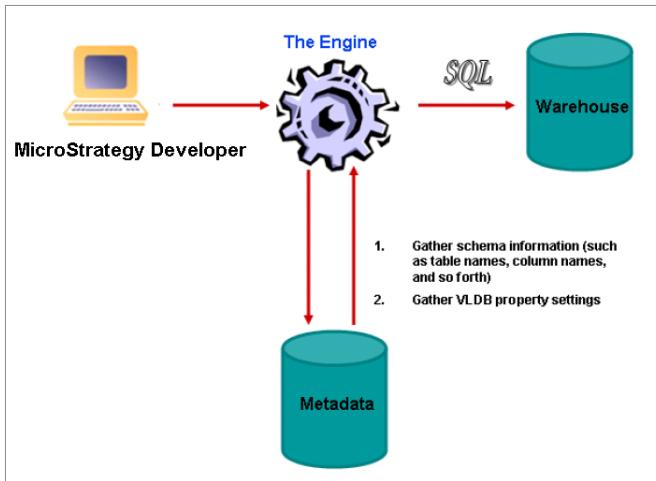
Intelligence Server processes a job on a database connection that corresponds to the job's priority. If no priority is specified for a job, Intelligence Server processes the job on a low-priority connection. For example, jobs with high priority are processed by high-priority connections, and jobs with low or no priority are processed by a low-priority connection.

Intelligence Server also engages in connection borrowing when processing jobs. Connection borrowing occurs when Intelligence Server executes a job on a lower priority connection because no connections that correspond to the job's priority are available at execution time. High-priority jobs can run on high-, medium-, and low-priority connections. Likewise, medium-priority jobs can run on medium- and low-priority connections.

When a job is submitted and no connections are available to process it, either with the same priority or with a lower priority, Intelligence Server places the job in queue and then processes it when a connection becomes available.

- **VLDB settings**—Each database has a unique range of performance optimizations that can be leveraged by MicroStrategy to improve query performance. These settings can be modified through MicroStrategy's Very Large Database (VLDB) settings. As shown in the following image, the MicroStrategy Engine reads schema information and VLDB properties from

the metadata before it generates the SQL required to interact with your data warehouse.



As the Platform Administrator, you should work with the developers to identify VLDB properties that should be modified. VLDB properties provide flexibility to customize SQL generation for a variety of situations. For example, VLDB settings can be modified to reduce the number of SQL passes for a given query, or they can modify VLDB settings in a given dataset to eliminate unnecessary joins.

Best Practice

Best practices for query optimizations

- 1 When a request for database access reaches the Intelligence Server, the server process will attempt to allocate a thread from the corresponding thread pool. If no threads of the appropriate priority are available, it will attempt to *borrow* a thread from any other thread pools that are of lower priority. If it is still unable to assign the request to an available thread, the request will be queued and put into *Waiting* status until a thread becomes available.

By default, all jobs are assigned a low priority. As a result, you must have at least one thread of low priority.

- 2 Enable database connection caching to reduce the overhead associated with Intelligence Server repeatedly connecting and disconnecting to the database. It is a good practice to efficiently use the database connection threads after they are established. You can adjust the following settings to adjust how long you want connections to be kept open (Cached or Busy):
 - **Connection lifetime (sec)**—When a connection thread is created, a timer starts counting and compares it to this limit. The entire duration of the connection cannot be longer than this limit. Depending on the status of the connection at the time the limit is reached, several things can happen:

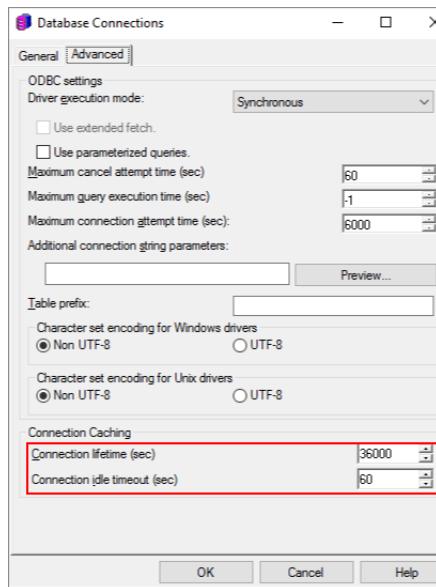
- If the Database Connection has a status of Cached (in other words, it is idle but available) when the limit is reached, the connection is deleted.
- If the Database Connection has a status of Busy (in other words, it is executing a job) when the limit is reached, the job completes, and then the connection is deleted and does not go into a Cached state.

The default value for this setting is 36000 seconds. A value of 0 implies no connection caching which means a connection is opened just to process a job and dropped as soon as the job has been processed. A value of -1 indicates no limit.

If this setting is too long, the data warehouse may have a limit that deletes the connection whether in the middle of the job or not. Ideally, you should set the Connection lifetime (sec) setting to a finite value which is the difference of the data warehouse timeout limit minus the time of the longest job.

- **Connection idle timeout (sec)**—This is the amount of time an inactive Database Connection thread remains cached in Intelligence Server until it is terminated. When a database connection finishes a job and there is no job waiting to use it, the database connection is cached, and a timer starts counting. If the time reaches the Connection idle timeout (sec) limit, the database connection thread is dropped. This is used to prevent idle connections from tying up the data warehouse and Intelligence Server resources if they are not needed. A value of 0 implies that the connection is never cached, while a value of -1 indicates no limit. If the connection

lifetime limit is smaller than than the idle timeout limit, the connection will get deleted once the connection lifetime limit is reached.



 For additional details on configuring these two settings, refer to Knowledge Base article KB5598.

- 3 Review jobs prioritization for each database instance. By default, if no prioritization is assigned, all jobs on the intelligence server run at a low priority.

As an initial starting point, it is recommended that prioritization categories be established according to the following guidelines:

Guidelines for Database Thread Prioritization

Recommended Database Thread Priority Configuration		
App Type	Request Type	Priority
All	Element	High
Web	Report	Medium
Developer	Report	Medium
Schedule	All	Low

- 4 Investigate the underlying database capacity for concurrent transactions. Increasing the number of available concurrent threads can produce enhanced reporting performance, therefore you can start by changing the default number of threads to a value between 5 and 10, and adjust them as follows:

- Using Enterprise Manager and database-level tools, monitor warehouse-level activity levels with adjustments/increases to the number of threads to determine at which point the best throughput is maintained.
- In coordination with the database administrator, perform a thorough series of tests against the data warehouse to determine the optimal number of concurrent database threads that the RDBMS can handle efficiently. These tests should make use of SQL queries generated from a variety of MicroStrategy reports as an input.

This result should then be used to configure the number of High, Medium, and Low priority threads available through the database connection.

The number of threads of each type should be allocated based on a subsequent analysis of typical activity and kept within the limits of the optimum number of overall database threads determined above. As Low is the default priority, always have at least one thread of Low priority for each active database instance.

Exercise 8.5: Configure DB connection timeouts

A warehouse database connection is initiated any time a user executes a report or browses elements that are not cached. As there is an overhead associated with establishing connectivity between the Intelligence Server and a data warehouse, you want to configure your environment so that when an ODBC connection is used for the initial query, it is cached for subsequent queries instead of being terminated immediately after a query has been executed against the data warehouse.

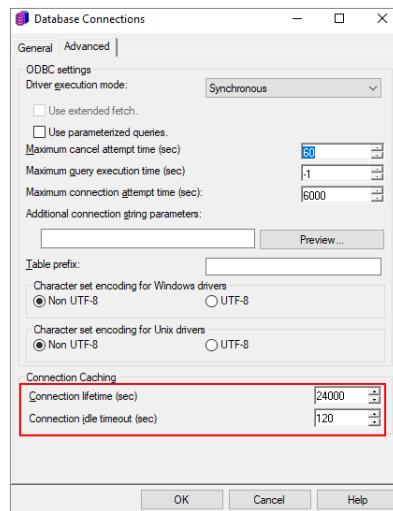
Based on the analysis of your environment, you have determined that MartZon should have a connection lifetime and connection idle timeout of 24,000 seconds and 120 seconds, respectively.

In this exercise, you will configure Connection lifetime (sec) and Connection idle timeout (sec) settings to 24,000 seconds and 120 seconds, respectively for the PostgresWarehouseConnection database connection in the PostgresWarehouseInstance database instance.

Access database instances editor

- 1 In Developer, log in to the **MicroStrategy on AWS I-Server** project source using the login credentials listed in the Welcome to MicroStrategy on AWS email.
- 2 Under **Administration**, expand **Configuration Managers**.
- 3 Select **Database Instances**. Then in the Object Browser window, double-click **PostgresWarehouseInstance**.
- 4 Under **Database connection (default)**, select **PostgresWarehouseConnection** and click **Modify**.
- 5 In the Database Connections window, select the **Advanced** tab.

- 6 Set the **Connection lifetime (sec)** and **Connection idle timeout (sec)** settings to **24000** seconds and **120** seconds, respectively.



- 7 Click **OK**. In the message window, click **OK**. In the Database Instances window, click **OK**.

After updating a database instance, you need to restart the Intelligence Server for the changes to take effect. As restarting the Intelligence Server can take a few minutes, for this exercise, you do not need to restart it.

Exercise 8.6: Configure connections and job priorities

If you have many concurrent jobs executing against the warehouse, you can improve the query response time of users in different groups or projects by using job prioritization. In this exercise, for the Tutorial Postgres database instance, you will:

- Set three connections of High and Medium priorities each
- Assign Medium priority to users in the MicroStrategy Web Professional group
- Assign High priority to requests coming from the MicroStrategy Tutorial project as well as to Element and MicroStrategy Developer requests
- Assign a cost in the range of 0 to 332 (which is by default designated as the Light cost range) to the Employee Headcount by Region report in the Human Resources Analysis folder so that it is executed ahead of other concurrent reports

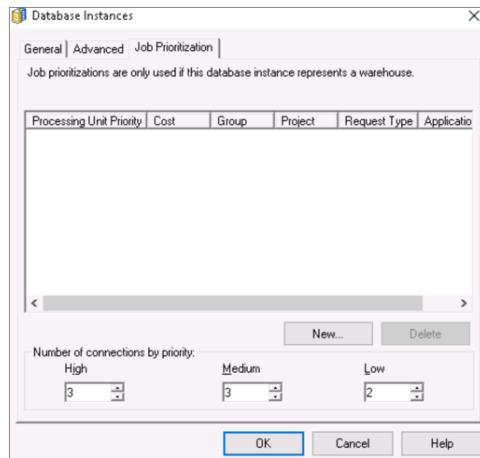
Access the Prioritization tab in Database Instances Editor

- 1 In Developer, log in to the **MicroStrategy on AWS I-Server** project source using the login credentials listed in the Welcome to MicroStrategy Cloud email.
- 2 Expand the **Administration** icon, followed by **Configuration Managers**. Then select **Database Instances**.
- 3 In the Object Viewer, right-click the **Tutorial Postgres** database instance and select **Prioritization**. The Database Instances Editor opens with the Job Prioritization tab selected.

Configure the number of High and Medium priority threads

You will now increase the number of connections of High and Medium priorities from two to three each.

- 4 Under Number of connections by priority, in the **High** and **Medium** boxes, enter **3** each as the number of connections of each priority.



Define job prioritization rules

Assign Medium priority to users in the MicroStrategy Web Professional group

You will now define two set of rules, one for the Medium priority and one for the High priority. You will first define the rule to assign Medium priority to users in the MicroStrategy Web Professional group.

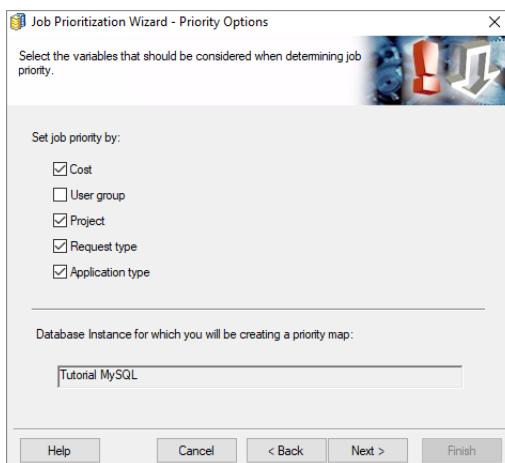
- 5 To define job prioritization rules, in the Database Instances editor, click **New**. In the Welcome window of the Job Prioritization Wizard, click **Next**.
- 6 In the Job Prioritization Wizard - Priority Options window, select the **User group** box and click **Next**.
- 7 In the Job Prioritization Wizard - Priority By User Group window, move the **MicroStrategy Web Professional** group from Available Groups to **Selected Groups** and click **Next**. Then in the Summary window, click **Finish**.

Define the rule to assign High priority

You will now define the second rule to assign High priority for requests coming from the MicroStrategy Tutorial project as well as to Element and MicroStrategy Developer requests.

- 8 In the Database Instances editor, click **New**. In the Welcome window of the Job Prioritization Wizard, click **Next**.

- 9** In the Job Prioritization Wizard - Priority By Job Cost window, select all options except **User group** and click **Next**.



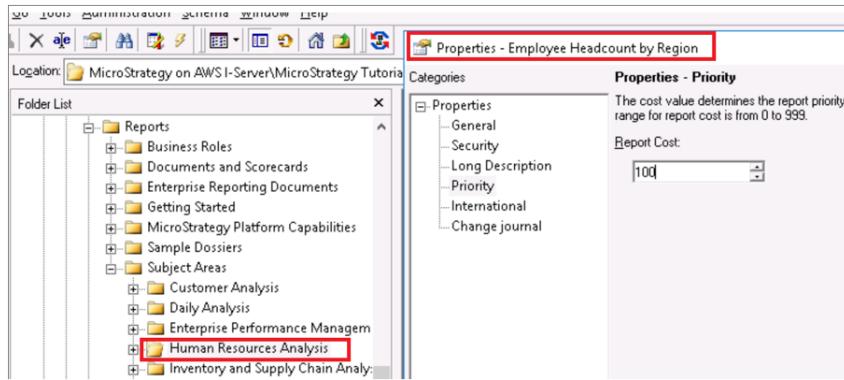
- 10** In the Job Prioritization Wizard - Priority By Job Cost window, accept default cost ranges and click **Next**.
- 11** In the Job Prioritization Wizard - Priority By Project, move **MicroStrategy Tutorial** to the **Selected Projects** pane and click **Next**.
- 12** In the Job Prioritization Wizard - Priority By Request Type window, select **Elements** and click **Next**.
- 13** In the Job Prioritization Wizard - Priority By Application Type window, select **Developer** and click **Next**. Then in the Summary window, click **Finish**.
- 14** In the Database Instances editor, click **OK**.

Assign report cost

You will now assign a cost in the range of 0 to 332 (which is by default designated as the Light cost range) to the Employee Profitability Analysis report in the Human Resources Analysis folder.

- 15** In the **Public Objects\Subject Areas\Human Resources Analysis** folder, right-click the **Employee Headcount by Region** report and select **Properties**.

16 In the Properties window, select **Priority** and in the Report Cost box, type **100** and click **OK**.



As by default, the Light cost range varies from 0 to 332, you can type any number in the 0 to 332 range; the MicroStrategy platform will consider each report that has an assigned cost from 0 to 332 as a Light report.

Placing limits on project use: Governors

Another optimization strategy that the Platform Administrator implement is to ensure that the analytics environments are governed. Intelligence Server provides governors that the platform administration team can use to place limits or controls on projects to keep overall system performance at acceptable levels.

For example, a governor may prevent a user from connecting to a project, keep a report from running, or limit the size of a report result set. There are governors to limit the number of interactive and scheduled jobs that can simultaneously run on the server and the number of users that can simultaneously connect to the server. You can also set idle timeouts. If a user is connected with no activity, the server automatically logs the user out after a certain period of time.

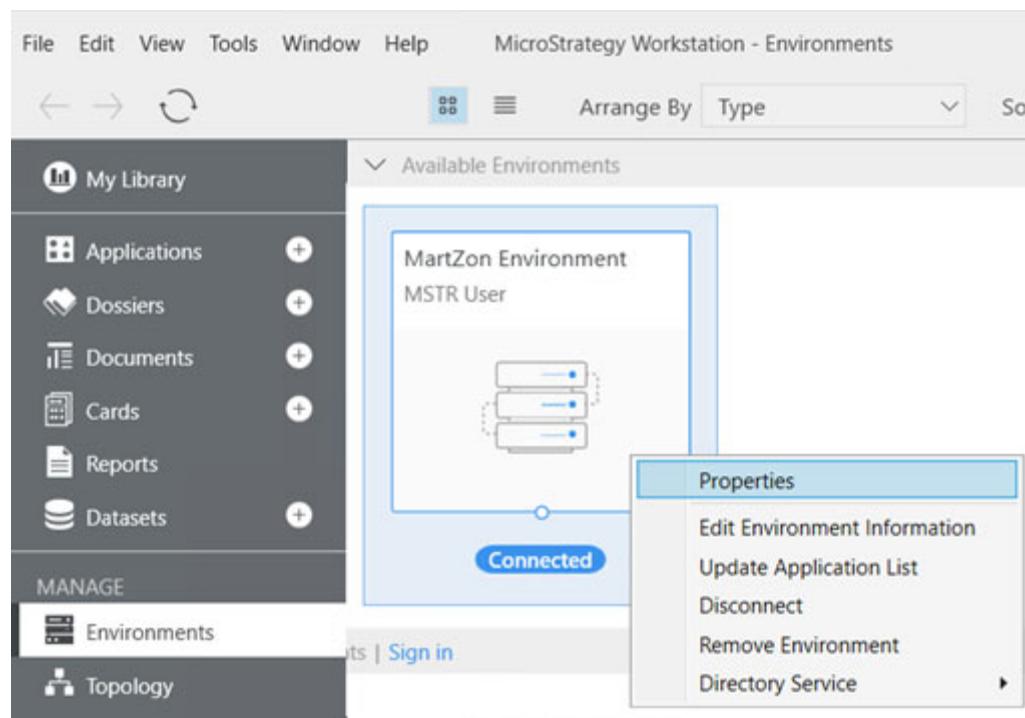
By specifying guidelines for the server-level and project-level governors, you can have the platform administration team set limits on a variety of that stress the system to help prevent the conditions that give rise to bottlenecks. When the limit for a governor is exceeded, the governor is invoked. Some of the important parameters for which you can consider providing guidelines for configuration include:

- Job-related governors
- User-related governors
- Memory-related governors

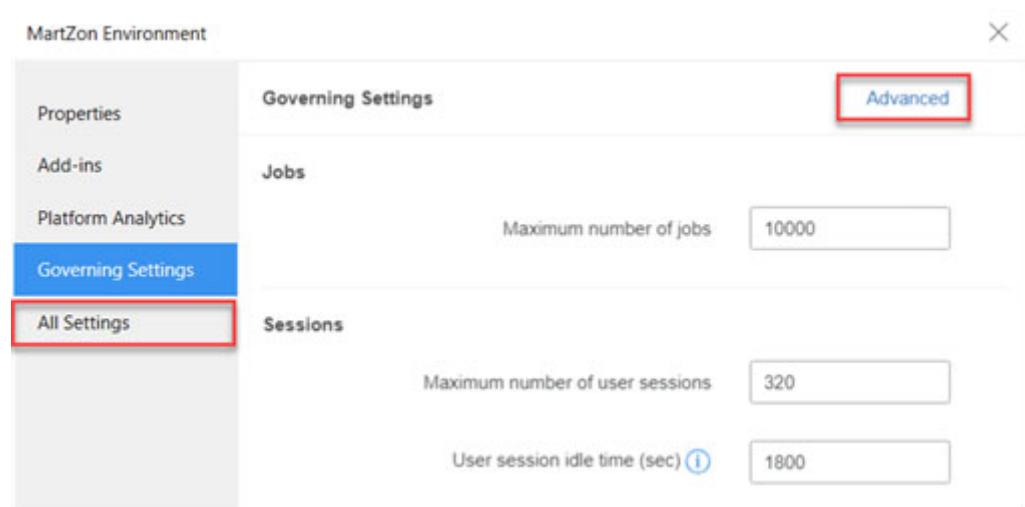
The following section each of these governors in more detail.

Accessing governing settings using Workstation

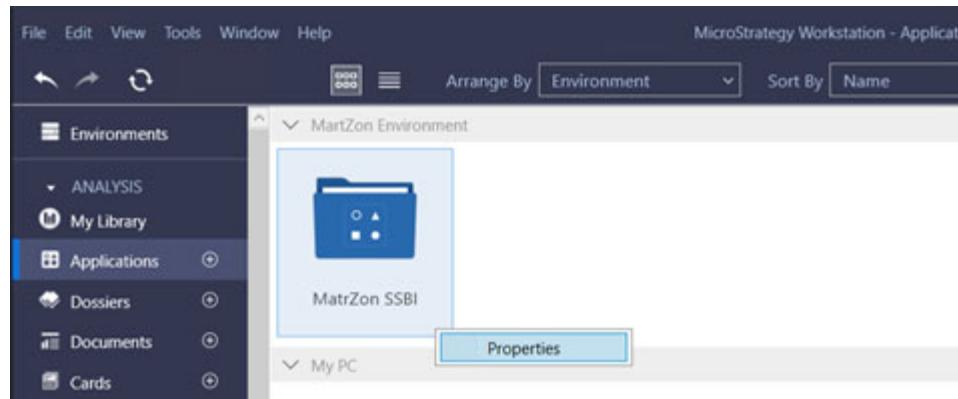
To access environment level governing setting from Workstation, connect to your environment, then right-click your environment connection and select **Properties**.



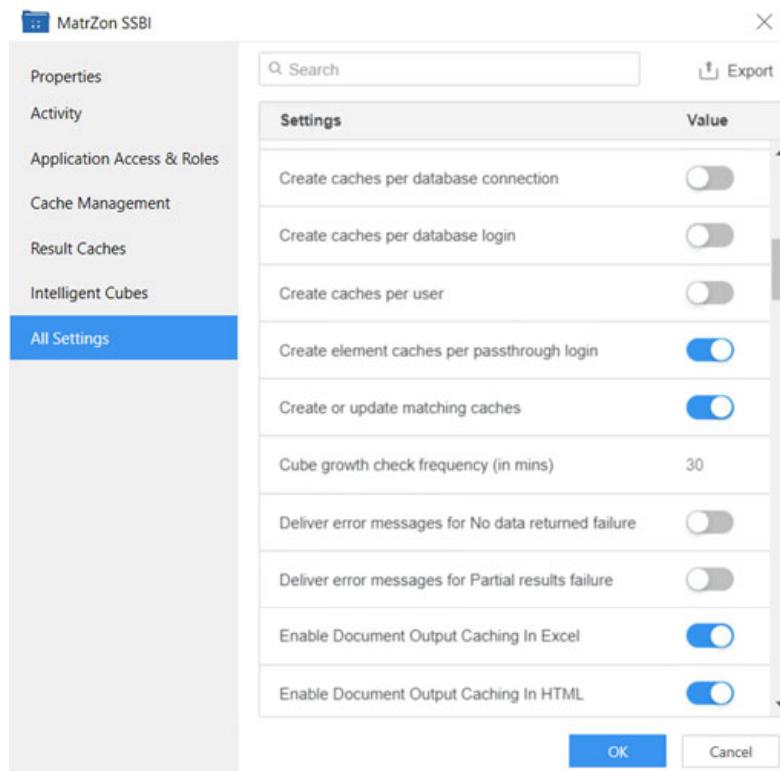
From the environment Window, select **Governing setting**. To access additional governing settings, select the **Advanced** link. You can access all environment settings by clicking on the **All Settings** tab.



To access application (project) level governing setting from Workstation, connect to your environment, navigate to the **Applications** tab, right-click your application and select **Properties**.



From the application properties window, select **All Settings**.



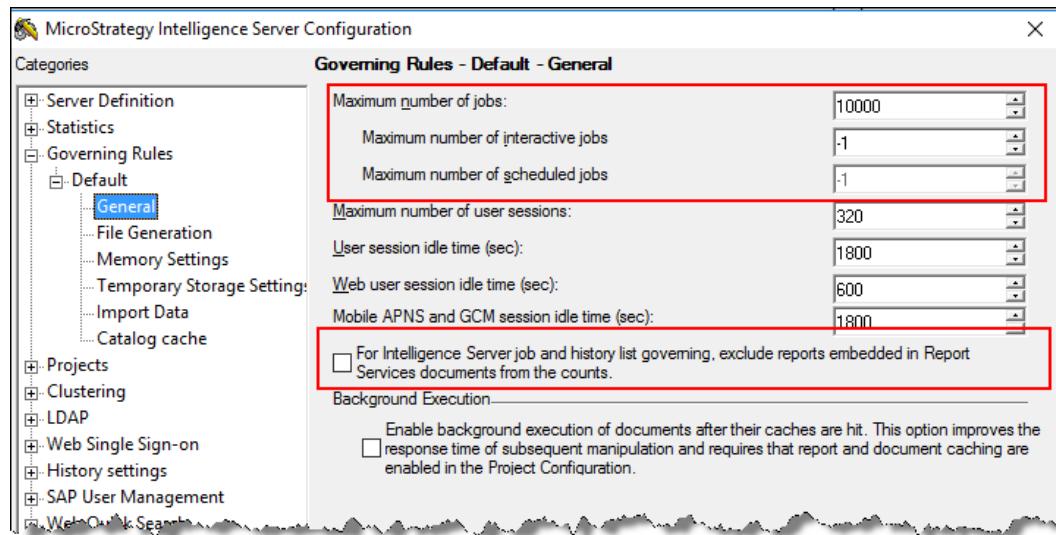
Job-related governors

Processing of the jobs uses up system resources (such as CPU time and server memory). The job-related governors limit the number of concurrent jobs executing at the server and project levels. Concurrent jobs include report, element, and auto prompt requests that are executing or waiting to execute.

Using the various server and project-level governors, you can reduce the potential for increased server memory consumption at runtime. However, setting these values too low can negatively limit runtime capacity and adversely impact user experience. Therefore, as the Platform Administrator, when setting values for these governors, take into consideration the maximum number of concurrent users; the desired system response time; the size of various reports, dossiers, and documents; and available system resources.

- **Server-level settings:**

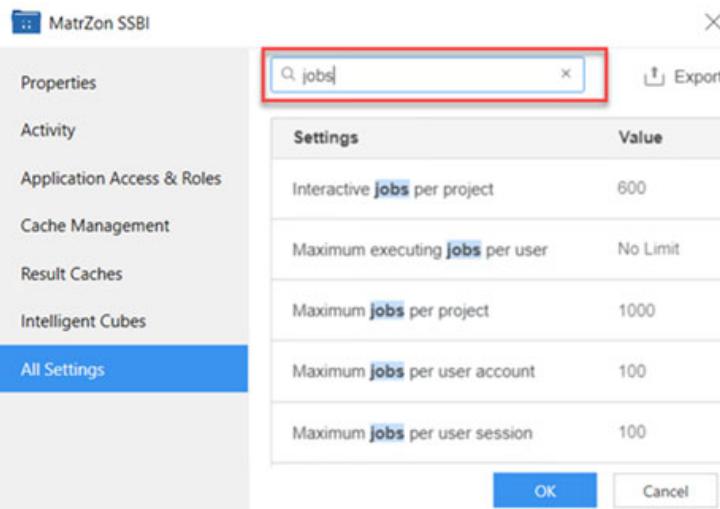
- **Maximum number of jobs**—Limits the number of concurrent jobs executing on Intelligence Server across all projects. Concurrent jobs include report, element, and auto prompt requests that are executing or waiting to execute. Using the *Maximum number of interactive jobs* and *Maximum number of scheduled jobs* settings, you can specify the maximum number of interactive jobs and the maximum number of scheduled jobs, respectively, that the Intelligence Server project processes at a time. Jobs are rejected if the limit is exceeded. A value of -1 indicates no limit.
- **For Intelligence Server job and history list governing, exclude reports embedded in Report Services Documents from the counts**—If this option is selected, reports executed as part of a Report Services Document are not counted against the server-level job limits above (and the project-level job limits set in the Project Configuration Editor).



- **Project-level settings:**

- **Jobs per user account**—Limits the maximum number of concurrent jobs for a given user account and a given project.

- **Jobs per user**—Limits the number of concurrent jobs that a user can run during a given session.
 Jobs are rejected if the preceding limits are exceeded.
- **Executing jobs per user**—Limits the memory consumption at runtime by limiting the number of concurrent jobs a single user account may have executing in the given project. If this condition is met, additional jobs are placed in the waiting queue until executing jobs finish. All requests are processed in the order in which they are received.
- **Jobs per project**—Limits the number of concurrent jobs that the project can process at any given time for all users. Using the *Interactive jobs per project* and *Scheduled jobs per project* settings, you can specify the maximum number of interactive jobs and the maximum number of scheduled jobs, respectively, that the selected project processes at a time. Jobs are rejected if the limit is exceeded.



User-related governors

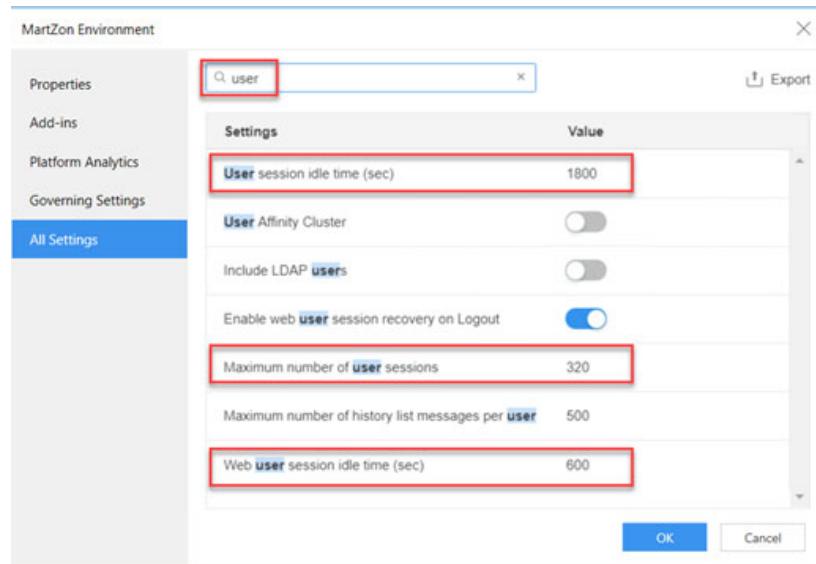
These governors control the number of user sessions. Reducing the number of user sessions, including active and idle sessions, reduces the runtime memory consumption by lowering both the memory used for connections and by diminishing the number of report requests present.

Using the various server and project-level governors, the platform administration team can reduce the potential for increased server memory consumption at runtime. However, setting these values too low can adversely impact the runtime system capacity. Therefore, as the Platform Administrator, when setting this value, take into consideration the maximum number of concurrent user sessions you

need to have, the expected system response time, and available system resources. Specify limits based on your peak usage conditions.

- **Server-level governors**

- **Maximum number of user sessions**—Limits the number of user sessions for Intelligence Server. Each session connects once to Intelligence Server and once to every project accessed by the user. If this limit is exceeded, new users cannot log in (except for an administrator).
- **User session idle time (sec)**—Represents the amount of time that a Developer user can remain idle before his session is ended. By effectively managing this setting, idle user's sessions are ended, thereby freeing up memory. The default value for this setting is 1800 seconds but you may initially set it to 600 seconds (10 minutes) and adjust it to a value suitable for your environment.
- **Web user session idle time (sec)**—Represents the amount of time that a MicroStrategy Web user can remain idle before his session is ended. By effectively managing this setting, idle user sessions are closed, thereby freeing up memory.



- **Project-level settings:**

- **User sessions per project**—Limits the number of user sessions (connections) that are allowed in the project. When the limit is reached, new users cannot log in, except for the administrator, who may wish to disconnect current users or increase the governing setting. A value of -1 indicates no limit.

- **Concurrent interactive project sessions per user**—Limits the number of concurrent interactive project sessions for a given user account. When the limit is reached, users cannot access new project sessions.

Settings	Value
Concurrent interactive project sessions per user	20

Memory-related governors

Intelligence Server components consume memory during startup and runtime. The memory-related governors minimize the memory consumption so as to delay or prevent the failure of Intelligence Server due to memory depletion.

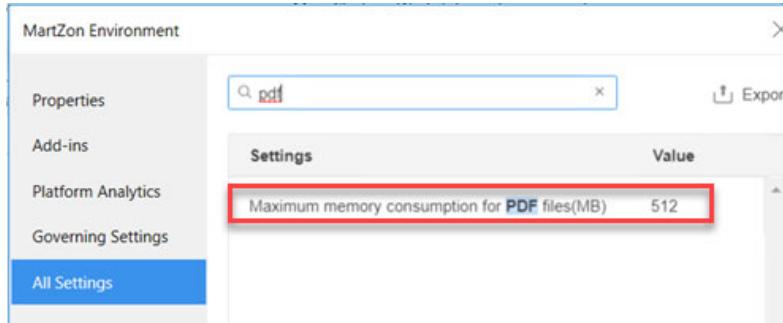
- **Server-level settings:**

- **Maximum number of XML cells**—Limits the maximum number of XML cells that the Intelligence Server can process at one time. This limit comes into play when the export operation requires Intelligence Server to return XML to the web server (for example, export to HTML, CSV, plain text, and Excel with plain text if exporting Portion displayed only).

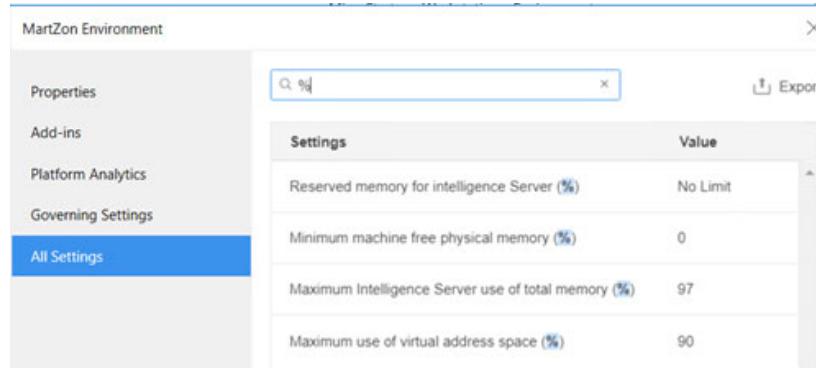
Decreasing the limit enables Intelligence Server to process less data at a time by using incremental fetch, which lowers memory consumption. When an export request exceeds the limit, the result set is automatically “chunked” in the background, but the entire result set still gets exported.

- **Maximum number of XML drill paths**—Limits the number of attributes to which users can drill in MicroStrategy Web. Attributes are displayed under the hierarchy to which they belong, and hierarchies are displayed in alphabetical order by the name of the hierarchy. If this setting is set to a low number, the available drill attributes may not all be displayed to the user. However, if it is set too high, performance may be affected because reports will consume more memory.
- **Maximum memory consumption for XML (MB)**—Limits the memory consumption for the XML files generated when MicroStrategy Web users submit requests (such as a report or document execution request) to Intelligence Server. As the Platform Administrator, set this limit based on the maximum expected size of the XML generated.
- **Maximum memory consumption for PDF/Excel/HTML files**—These three settings limit the memory consumption for PDF, Excel, and HTML

files. As the Platform Administrator, you should set the limits for these settings according to the expected size of the PDF/Excel/HTML files to be generated to avoid memory-related errors.



- **Enable Web request throttling**—Throttling occurs when Intelligence Server memory usage exceeds the governing limit specified for the Maximum Intelligence Server use of total memory (%) setting in the MicroStrategy Intelligence Server Configuration Editor. When Intelligence Server reaches throttling state, it denies all requests from a MicroStrategy Web client (or a client built with the MicroStrategy Web API) until the memory usage drops below the limit.



When you enable Web request throttling, you can configure the Maximum Intelligence Server use of total memory (%) setting which specifies the maximum amount of total system memory (physical (RAM) and swap (disk page file)) that can be used by Intelligence Server compared to the total amount of memory on the machine. This setting helps in preventing the system from servicing a MicroStrategy Web request if memory is depleted. The default value for this setting is 97%.

You can also configure the Minimum machine free physical memory (%) setting which specifies the minimum amount of available RAM below which any additional MicroStrategy Web request would be denied until more RAM is freed up. This value is based on the percentage of the total amount of physical memory (RAM) on the machine. This setting is useful if

you want to increase the chances that MicroStrategy Web requests are serviced using RAM and not the disk page file, which is slower. If the condition is met, Intelligence Server denies all additional requests from a MicroStrategy Web client until the condition is no longer met. The default value for this setting is 0.

- **Enable memory contract management**—Memory Contract Manager is an Intelligence Server component that grants or denies memory requests from a set of tasks based on the amount of available memory. In production environments, Memory Contract Manager should be enabled to avoid Intelligence Server memory depletions.
- **Max RAM for Working Set cache (MB)**—The Working Set is a collection of messages in memory that reference report results. When a MicroStrategy Web user retrieves a message from his inbox or runs a report, the results from the report are added to the Working Set for that user's connection and stored in the memory on the Intelligence Server machine. Manipulations can then be performed on the data set without having to run SQL against the database.

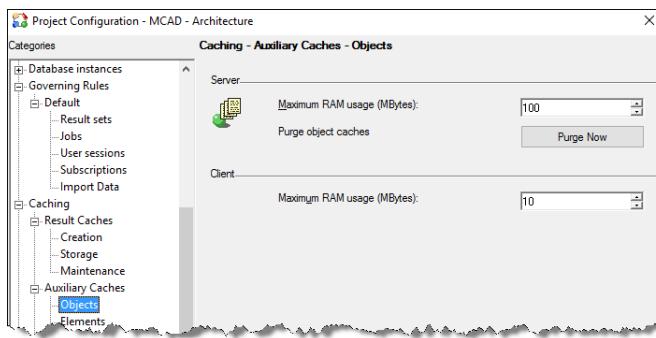
The Working Set space can be consumed quickly if users do a lot of manipulations. This is because the manipulated version of a report is added to the Working Set after any manipulation takes place.

In production environments, you should use the Maximum RAM for Working Set cache (MB) setting to specify the size of the pool of memory (in megabytes) allocated for creating and storing reports in the Working Set. This is also the size of the largest working set that can be created. The default value for this setting is 200 MB but you should initially increase it to 25% of available RAM for Intelligence Server, and adjust it based on your environment.

- **Project-level governors**

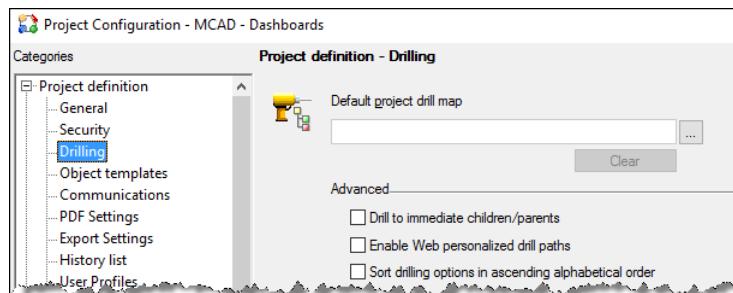
- **Objects - Server: Maximum RAM usage (MBytes)**—The object cache represents caches for schema objects (such as attributes) and application objects (such as reports) used by Intelligence Server to speed the retrieval of objects from the metadata. In production environments, set the maximum RAM usage (MB) for object caching on the Intelligence Server

machine initially in the 2-4 GB range, and adjust it based on your environment.



- **Elements - Server: Maximum RAM usage (MBytes)**—The element cache is stored in memory or in a cache file located on the Intelligence Server. In production environments, if using element prompts, set the maximum RAM usage (MB) for element caching on the Intelligence Server machine initially to at least 1 GB, and adjust it based on your environment.
 - ❖ Object and element caches exist both on the server and the client (Developer) machines.
- **Web personalized drill paths**—Web users have the option to see only personalized drill paths rather than all drill paths. Personalized drill paths are based on each object's access control list (ACL). If you set up ACLs, all drill paths are still displayed in MicroStrategy Web until you enable Web personalized drill paths.

In production environments, you should not enable this setting as it turns off XML caching which in turn adversely impacts MicroStrategy Web performance.



- ❖ As Enable personalized drill paths is a project-level setting, you need to manage it for each project separately.

Exercise 8.7: Set usage limits using governors

You can use server- and project-level governors to set usage limits on a variety of parameters. In this exercise, you will change two governors and then test the effects of your changes.

You will first modify a project-level governor for the Human Resources Analysis Module project. You will limit the final result rows for all other reports to 75.

Next, you will change a server-level governor to limit the maximum number of user sessions to 1.

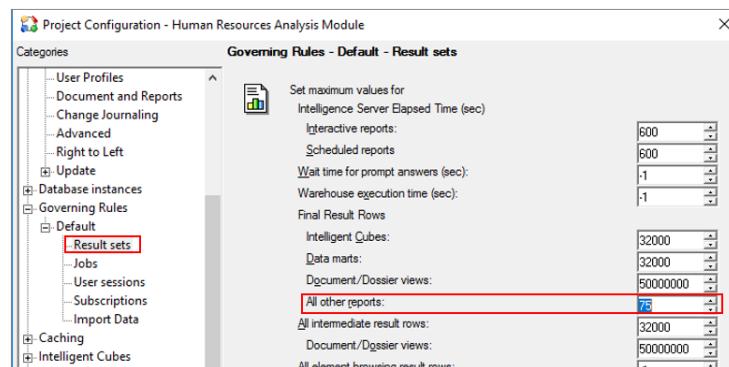
Finally, you will test the new governors by running the Time Off Overview by Department report located in the Human Resources Analysis Module\Public Objects\Reports\Benefit Analysis folder. You will receive an error.

You will then try to create a second user session by logging in to the Human Resources Analysis Module project in MicroStrategy Web as demo user with no password. You will again receive an error.

Govern your environment

Modify a project-level governor

- 1 In Developer, while logged in as the **mstr** user, right-click the **Human Resources Analysis Module** project and select **Project Configuration**.
- 2 In the Project Configuration Editor, expand **Governing Rules**.
- 3 In the Governing Rules - Default - Result sets pane, under Final Result Rows, in the **All other reports** box, type **75**. Click **OK**.



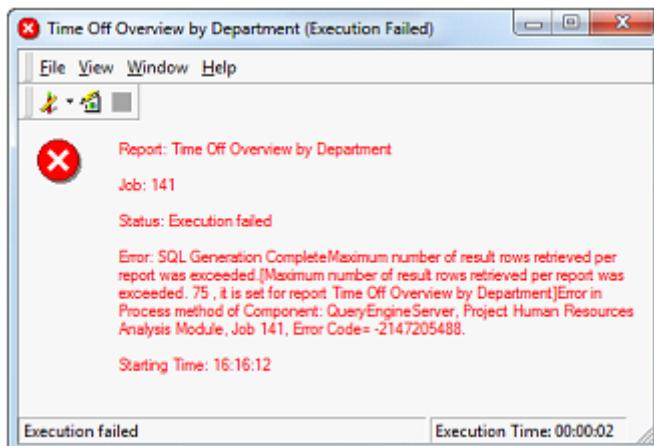
Modify a server-level governor

- 4 In the Folder List, right-click the **MicroStrategy on AWS I-Server** project source and select **Configure MicroStrategy Intelligence Server**.
- 5 In the Intelligence Server Configuration Editor, expand **Governing Rules**.
- 6 In the Governing Rules - Default - General pane, in the **Maximum number of user sessions** box, type **1**. Click **OK**.

Test the project-level governor

- 7 While still logged in Developer as the **mstr** user, expand the **Human Resources Analysis Module** project.
- 8 Navigate to the **Public Objects\Reports\Benefit Analysis** folder. Run the **Time Off Overview by Department** report.

The following message displays:



This message displays because the report returns 88 rows of data, which is larger than the 75 rows limit enforced by the governor.

- 9 Close the report. Stay logged in Developer as the **mstr** user.

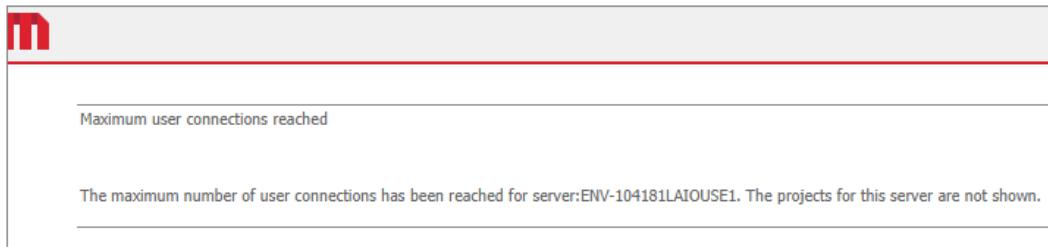
Test if you can create a new user session

You had configured the server-level governor limit for the maximum number of user sessions to 1. While you are still logged in Developer as the **mstr** user, you will

now try to create a second user session by logging in to the Human Resources Analysis Module project in MicroStrategy Web as demo user with no password.

- 1 In MicroStrategy Web, log in to the **Human Resources Analysis Module** project as the **ffdemo** user with a blank (no) password.

You should see a message regarding Maximum user connection reached because of the limit specified for the maximum number of user sessions governor.



Reset your project-level and server-level governor parameters

You will now reset the governor parameters back to the default values to enable you to work on other exercises in this course.

- 2 In Developer, reset:

- The final result rows for all other reports project-level governor for the Human Resources Analysis Module project to **32000**.
- The maximum number of user sessions server-level governor to **320**.

Exercise 8.8: Troubleshoot a governing issue

You have noticed that many Report Services documents scheduled using Distribution Services are failing with the following error in the DSSErrors.log:

MsiUserSession::AddUserJob: maximum jobs per user connection on project XXX exceeded

Upon further analysis, you noticed that the failing documents have multiple datasets. What can your platform administration do to resolve this issue?

Exercise 8.8 Solutions: Troubleshoot a governing issue

You have noticed that many Report Services documents scheduled using Distribution Services are failing with the following error in the DSSErrors.log:

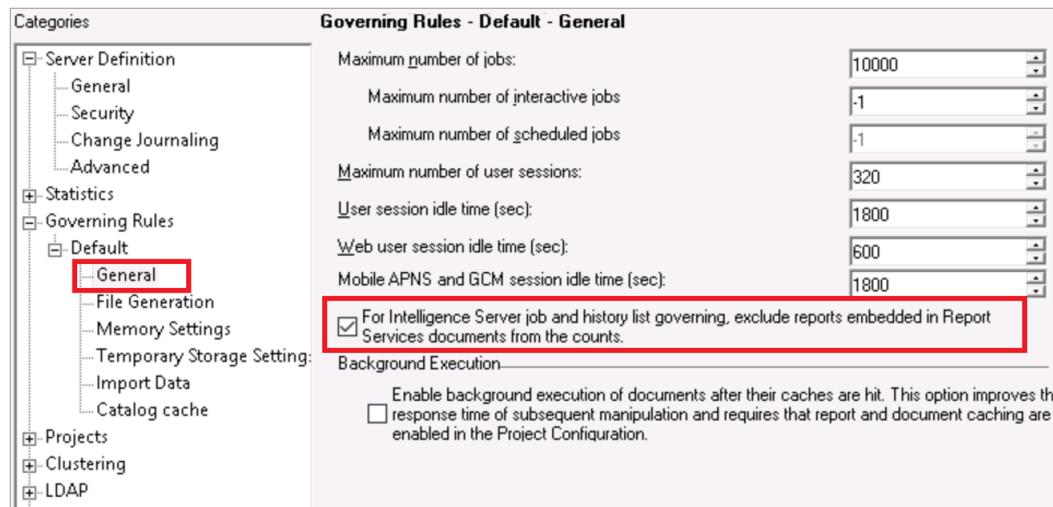
MsiUserSession::AddUserJob: maximum jobs per user connection on project XXX exceeded

Upon further analysis, you noticed that the failing documents have multiple datasets. What can your platform administration do to resolve this issue?

Resolving governing issue

As the failing documents have multiple datasets, one possible cause for the issue is that each of the reports being used as a dataset within the document is being counted against your governing setting limits.

To resolve the issue, in Developer, in the MicroStrategy Intelligence Server Configuration Editor, for the General option under Governing Rules, select **For Intelligence Server job and history list governing, exclude reports embedded in Report Services documents from the counts.**



Enabling the preceding option will allow reports which are embedded in a Report Services document to not count against Intelligence Server or Project governing settings, and should resolve the issue.

A

APPENDIX: PLATFORM ADMINISTRATOR CHECKLIST

Platform Administrator description



Responsible for the Enterprise Intelligence Platform Environments, including Project Configuration, Users Provisioning, Analytics Security, Distribution Services, Cube Configuration, Cache Management, Project Objects Migration, and Platform Environment Monitoring Tools.

The Enterprise Intelligence Platform Environment hosts the data, application, security, and related services necessary to deploy analytics and mobility applications throughout the Enterprise.

Check list overview

Assess

- 1** Number of Total Users
- 2** Number of Active Users
- 3** Number of Total Projects
- 4** Number of Active projects
- 5** Number of Datasets
- 6** Number of Active Datasets
- 7** Number of Total Environments
- 8** Number of Active Environments
- 9** Number of Total Objects
- 10** Number of Active Objects
- 11** Number of Subscriptions Successful
- 12** Number of Failed Subscriptions

Plan

- 1** Platform Environments Architecture
- 2** Platform Services Architecture
- 3** Platform Data Architecture
- 4** Platform Project Architecture
- 5** Platform Security Architecture
- 6** Platform Configuration Protocols
- 7** Analytics Security Architecture

Create

- 1** Users and User Groups
- 2** Configure Directory / Identity Integrations
- 3** Platform Environments
- 4** Distribution Services Subscriptions
- 5** Cache and Cube Refresh Schedules
- 6** Data Connectors within the Platform Environments

Publish

- 1** Platform Environments Synchronization
- 2** Platform Analytics
- 3** Configuration Documentation
- 4** Updated Upgrade Procedures
- 5** Operational Procedures

Operate

- 1** Monitor Platform Report
- 2** Support Intelligence Center Architects
- 3** Handle Platform Environment Cases
- 4** Troubleshoot Platform Issues
- 5** Coordinate with Intelligence Center
- 6** Provision User Access
- 7** Upgrade environments

Optimize

- 1** Platform Project Performance
- 2** Platform Environment Performance
- 3** Platform Services Performance
- 4** Enterprise Data Set Performance
- 5** Enterprise Applications Performance
- 6** Enterprise Mobile Applications Performance

Assets and tooling

Assess

- MicroStrategy Developer is an Administration and Architect tool
- MicroStrategy Command Manager is a command line tool
- Enterprise Manager/ Platform Analytics are used for MicroStrategy Project Statistics Collection and Reporting
- Collaborative platform (SharePoint) is used to securely share documentation

Plan

- MicroStrategy Installation Wizard is a step-by-step program to guide the software installation
- Installation response files are configuration files containing instructions for silent installation
- MicroStrategy Developer is an Administration and Architect tool
- MicroStrategy Command Manager is a command line tool
- MicroStrategy License Manager is a tool to manage MicroStrategy licenses
- Identity management tools are used to manage users, computers and other devices on a network

- Web/Mobile Application Server interface or command line (IIS, Apache) are interfaces to configure the Web/Mobile servers
- MicroStrategy Web Administrator page is used to configure MicroStrategy administrative settings
- Enterprise Manager/ Platform Analytics are used for MicroStrategy Project Statistics Collection and Reporting

Create

- MicroStrategy Developer used to build objects used in Mobile
MicroStrategy Developer is an Administration and Architect tool
- MicroStrategy Command Manager is a command line tool
- OS file system is used to backup MicroStrategy configuration files, or web/
Mobile configuration files

Publish

- Text editing tool (MS Word, Adobe) is used to edit documentation for several purposes
- Collaborative platform (SharePoint) is used to securely share documentation
- Enterprise Manager/ Platform Analytics are used for MicroStrategy Project Statistics Collection and Reporting
- Integrity Manager tool is used to automatically compare objects in MicroStrategy Projects, and determine how specific changes in a Project, such as the regular maintenance changes to metadata objects or hardware and software upgrades, affect the reports and documents in that Project
- Change Request Software is used to ensure the safety and reliability of objects migrations by providing current, approved, and released documentation for specific changes
- Version Control Software is used to record Project objects changes over time so that specific versions can be retrieved later

Operate

- MicroStrategy Developer is an Administration and Architect tool
- MicroStrategy Command Manager is a command line tool

- MicroStrategy Diagnostics Configuration Tool is used to collect Diagnostics and Performance Log Files
- MicroStrategy Server Manager is used to start, stop, or re-start the MicroStrategy processes
- MicroStrategy Installation Wizard is a step-by-step program to guide the software installation
- Installation response files are configuration files containing instructions for silent installation
- Integrity Manager tool is used to automatically compare objects in MicroStrategy Projects, and determine how specific changes in a Project, such as the regular maintenance changes to metadata objects or hardware and software upgrades, affect the reports and documents in that Project
- Collaborative platform (SharePoint) is used to securely share documentation
- Alert and Monitoring Tools are used to notify users in case certain thresholds are exceeded, or to report the Platform KPIs
- 3rd party status reporting tools are used to create and distribute Platform Environments status reports in case there is no MicroStrategy tool available for this purpose
- Case Management System is used to open and track Technical Support issues

Optimize

- MicroStrategy Developer is an Administration and Architect tool
- MicroStrategy Command Manager is a command line tool
- Collaborative platform (SharePoint) is used to securely share documentation
- Enterprise Manager/Platform Analytics are used for MicroStrategy Project Statistics Collection and Reporting
- MicroStrategy Web Administrator page is used to configure MicroStrategy administrative settings

Detailed check list

Assess

Environments

Key Performance Indicators:

- Number of Total Environments
- Number of Active Environments

Troubleshooting

- If the number of Active Users is significantly less than the previous day's value, If the number of Active Environments is less than the Total Environments, identify the root cause and decommission the obsolete Environments

Users and Projects

Key Performance Indicators:

- Number of Total Users
- Number of Active Users
- Number of Total Projects
- Number of Active Projects

Troubleshooting

- If the number of Active Users is less than the Total Users, identify the root cause and disable inactive users
- If the number of Active Projects is less than Total Projects, identify the root cause and work with the Analytics Architect and Application Architect to plan the deletion of the inactive Projects or merge similar Projects to reduce the metadata size

Project Objects

Key Performance Indicators:

- Number of Datasets
- Number of Active Datasets
- Number of Total Objects (schema and application)
- Number of Active Objects (schema and application)

Troubleshooting

- If the number of Objects or Datasets is less than the Total Objects or Datasets, identify the root cause and work with the Analytics Architect to plan a clean-up strategy

Subscriptions

Key Performance Indicators:

- Number of Subscriptions Successful
- Number of Failed Subscriptions

Troubleshooting

- Identify the root cause of subscriptions failures, contact the owners, and either remove the failed subscriptions, or remediate the failure reason
- Inventory the Successful Subscriptions and check if those are still current and used by the recipients. Plan a clean-up strategy for unused subscriptions

Plan

Platform Environments Architecture

- Install the MicroStrategy software on all Enterprise Intelligence Platform Environments (Intelligence Server, Web, Mobile, and Collaboration Servers, client products and tools) using MicroStrategy Installation Wizard or response files to ensure the Platform Environments are up and running

- Configure the Intelligence Server (server definition, governing settings, clustering, history list) using MicroStrategy Developer to ensure the Intelligence Server is setup and ready to integrate with the rest of the Enterprise Intelligence Platform Environments
- Configure the Web and Mobile components (application server settings, plug-ins) Web/Mobile/Collaboration Application Server interface/command line and the MicroStrategy configuration files to ensure the Web/Mobile/Collaboration Servers integrate properly with the Intelligence Server
- Apply MicroStrategy licensing and activation using MicroStrategy License Manager to check that all Server installations are active, and MicroStrategy software installation follows the license contract
- Platform Services Architecture
- Configure Distribution Services (devices, transmitters) using MicroStrategy Developer or MicroStrategy Command Manager to ensure a reliable and efficient distribution system
- Configure database connection threads using MicroStrategy Developer to balance user requests jobs response time and the overall system load
- Configure jobs prioritization using MicroStrategy Developer to determine which jobs are submitted to the data warehouse before other jobs in the queue
- Configure events and schedules using MicroStrategy Developer to control the Platform Services execution

Platform Data Architecture

- Configure database instances for the Intelligent Platform Environments data sources using MicroStrategy Developer to ensure proper connectivity with the Enterprise Intelligence Platform Environment data
- Enable internationalization (Data, Metadata, Web, and Mobile Servers) using MicroStrategy Developer to ensure that Intelligent Platform Environments are available to a multilingual audience

Platform Project Architecture

- Configure Projects using MicroStrategy Developer or MicroStrategy Command Manager to ensure that Platform Projects are governed
- Define the Cube configuration and refresh strategy using MicroStrategy Developer for easy data maintenance

- Define the Cache management strategy using MicroStrategy Developer to ensure fast data access

Platform Security Architecture

- Configure connectivity with third-party Directory / Identity providers using MicroStrategy Developer and Identity Management tools to ensure that the Intelligent Platform Environments access is seamless and secure
- Apply Platform Environments security communication settings (ports, certificates) using MicroStrategy Developer and the Web/Mobile/ Collaboration Application Server interface/command line to ensure the Enterprise Intelligence Platform Environments infrastructure security
- Define users' authentication modes using MicroStrategy Developer and MicroStrategy Web Administrator page to allow users to securely access the Enterprise Intelligence Platform Environments

Platform Configuration Protocols

- Configure Enterprise Manager / Platform Analytics using Enterprise Manager tool / Platform Analytics interface to report statistical data on Intelligent Platform Environments usage

Analytics Security Architecture

- In collaboration with the Intelligence Center Architects, define the MicroStrategy security configuration (security roles, permissions, privileges, security filters) using MicroStrategy Developer to safeguard the Intelligent Enterprise Platform

Create

Users and User Groups

- Create Users and User Groups using MicroStrategy Developer or MicroStrategy Command Manager
- Configure and assign Users authentication mode using MicroStrategy Developer or MicroStrategy Command Manager to ensure that users can securely access the Enterprise Intelligence Platform Environments
- Import Users and User Groups from 3rd party Directory / Identity providers using MicroStrategy Developer or MicroStrategy Command Manager to

synchronize Users and User Groups with 3rd party Directory providers and streamline User maintenance

- Assign Users to User groups and apply security roles using MicroStrategy Developer or MicroStrategy Command Manager to simplify and streamline User maintenance
- Assign permissions and privileges to User Groups using MicroStrategy Developer or MicroStrategy Command Manager
- Create security filters and assign them to User groups using MicroStrategy Developer or MicroStrategy Command Manager to ensure data security

Configure Directory / Identity Integrations

- Work with the System Administrator to obtain the required parameters of the Directory or Identity Integrators to manage the Platform Environments identity and access

Platform Environments

- Create scripts for recurring Platform administration tasks using MicroStrategy Command Manager to increase productivity and eliminate manual errors
- Work with the System Administrator to ensure that the Platform Environments configuration backup is scheduled

Distribution Services Subscriptions

- Inventory the Distribution Services Subscriptions in each Intelligent Platform Environment using MicroStrategy Developer or MicroStrategy Command Manager to have an overview of the existing number of subscriptions
- Create Subscriptions deletion scripts to remove unused subscriptions using MicroStrategy Command Manager for a clean environment

Cache and Cube Refresh Schedules

- Enable Element, Object, Report, Dossier, and Document Cache using MicroStrategy Developer or MicroStrategy Command Manager for faster data access
- Create Cache and History List management schedules and/or scripts (delete, update, invalidate, expire) using MicroStrategy Developer or MicroStrategy Command Manager for easy data maintenance

- Create Cube management schedules and/or scripts (publish, refresh, update) using MicroStrategy Developer or MicroStrategy Command Manager for easy data maintenance

Data Connectors within the Platform Environments

- Configure database instances, database connections, and database logins for Intelligent Platform Environments data sources using MicroStrategy Developer or MicroStrategy Command Manager to provide data access for Users and Platform Services
- Configure custom connectors using MicroStrategy Developer or MicroStrategy Command Manager to provide data access for Users and Platform Services

Publish

Platform Environments Synchronization

- Communicate and follow the Platform Environments change management process using a Collaborating Platform and a Change Request Software to ensure a proper Platform Environments governance
- Review the Change Request
- Communicate with the Change Request reviewer and approver to clarify the object migration scope
- Identify objects dependencies and conflict resolutions
- Execute object migrations between Platform Environments for object promotion using MicroStrategy Object Manager to make sure the latest updates are published to the users
- Ensure that the object migration is performed from the DEV Environment to UAT and then to Departmental / Enterprise Environments
- Validate migration using Integrity Manager
- Coordinate with the Intelligence Center Architects to ensure that unit, regression, functional, and performance testing is executed for the migrated objects and their dependencies
- Store object migration packages in a Version Control Software

Platform Analytics

- Publish Platform Analytics reports for an overview of the Enterprise Intelligence Platform Environments using Platform Analytics dossiers
- Publish Enterprise Manager reports for an overview of the Enterprise Intelligence Platform Environments using Enterprise Manager dossiers

Configuration Documentation

- Document how Platform Environments are configured using a Text Editing tool to ensure a proper Platform Environments governance
- Document how the Platform Environments configurations and their relationships have changed over time using a Text Editing tool and a Version Control Software to ensure a proper Platform Environments governance
- Maintain and publish the Platform Environments Configuration Documentation using a Collaborating Platform to ensure a proper Platform Environments governance
- Work with the Intelligence Center Architects to get the data dictionary and documentation for the reports and dossiers using a Collaborating Platform to understand how the Platform Environments are used

Updated Upgrade Procedures

- Work with the Intelligence Director to determine the Platform Environments upgrade schedule using a Collaborating Platform to ensure a proper Platform Environments governance
- Document the updated Upgrade Procedures for new product features, or changes to exiting features using a Text Editing tool to ensure that the Upgrade Procedures are current
- Maintain the Upgrade Procedures document control using a Collaborating Platform and a Version Control Software to ensure a proper Platform Environments governance

Operational Procedures

- Publish the Platform Environments administrative and maintenance scripts using a Collaborating Platform to ensure a proper Platform Environments governance

- Maintain the Operational Procedures document control using a Collaborating Platform and a Version Control Software to ensure a proper Platform Environments governance
- Notify end users for pre-planned or ad-hoc maintenance windows using a Collaborating Platform to ensure a proper Platform Environments governance

Operate

Monitor Platform Report

- Check the Intelligent Platform Servers status (Intelligence Server, Web, Mobile, Collaboration) and use Alerting and Monitoring Tools to ensure that Platform Environments are up and running
- Monitor User Connections, Database Connections, and Job statuses using MicroStrategy Developer to check long running session or job bottlenecks
- Monitor Cache and Cube utilization, and purge them using MicroStrategy Developer to ensure that the Intelligence Server memory is not depleted
- Audit the Intelligent Platform configuration and track configuration changes over time using MicroStrategy Developer or MicroStrategy Command Manager to establish a baseline for future comparisons
- Automate and distribute periodic reports containing the Intelligence Platform Environments health status using MicroStrategy Developer, MicroStrategy Command Manager, a Collaborating Platform, and 3rd party tools to ensure Platform Environments status visibility

Support Intelligence Center Architects

- Coordinate with the Intelligence Center Architects using a Collaborating Platform to convey immediate issues or areas for improvement
- Communicate and follow the Platform Environments governance process using a Collaborating Platform to ensure that business objectives are met

Handle Platform Environment Cases

- For critical issues such as Intelligence Server abnormal shutdown (crash), or Intelligence Server unresponsiveness (hang), do the following:
 - Work with System Administrator to bring the Intelligence Server up and running as soon as possible

- Log a case with MSTR tech support for root cause analysis using a Case Management System
- For non-critical issues, do the following:
 - Log a case with MSTR tech support for root cause analysis using a Case Management System
 - Follow up on the case until a resolution is reached and communicate the resolution to the case owner

Troubleshoot Platform Issues

- Capture, investigate, and backup the Intelligent Platform Servers logs using MicroStrategy Diagnostics Configuration Tool and the OS file system to ensure an accurate and timely issue resolution
- Start, stop, restart the Intelligent Platform Servers after configuration updates using MicroStrategy Service Manager to ensure that changes have been applied
- Collaborate with the System Administrator and Intelligence Center Architects to resolve the issues reported for the Platform Environments to ensure an accurate and timely issue resolution

Coordinate with Intelligence Center

- Attend daily Intelligence center/scrum meeting
- Provide update on the status of the Enterprise Intelligence Platform Environments

Provision User Access

- Grant Users Access to Projects and apply Project Objects security using MicroStrategy Developer or MicroStrategy Command Manager to safeguard the Enterprise Intelligence Platform Environments
- Collaborate with the System Administrator and Intelligence Center Architects to ensure that the security compliance requirements are met (HIPAA, SOX, GDPR)

Upgrade environments

- Upgrade the Platform Environments using MicroStrategy Installation Wizard or response files to take advantage of the latest product features

- Collaborate with the Intelligence Center Architects on the upgrade testing using Integrity Manager, MicroStrategy Developer or MicroStrategy Command Manager to ensure the upgrade is successful

Optimize

Platform Project Performance

- Review and adjust Project governor and VLDB settings using MicroStrategy Developer or MicroStrategy Command Manager to ensure optimal performance
- Review Cache and Cube settings and ensure enough memory is allocated using MicroStrategy Developer to balance disk swapping vs. Cache and Cube access performance

Platform Environment Performance

- Ensure that enough memory is allocated for the Working set manipulation using MicroStrategy Developer or MicroStrategy Web to improve Web performance
- Tune the Web server load balance factor using Web Server Administrator page to balance the Intelligence Servers load in a clustered environment

Platform Services Performance

- Review the number of users and jobs executed on the system using Platform Analytics or Enterprise Manager to assess whether the load on the system has remained constant or has changed significantly
- Automate Cache and Cube maintenance scripts using MicroStrategy Developer or MicroStrategy Command Manager to eliminate unnecessary objects and jobs

Enterprise Data Set Performance

- Adjust Database Instance connection and VLDB settings using MicroStrategy Developer to ensure optimal query execution time
- Tune Database connection using MicroStrategy Developer threads to improve query response time while not overloading the system

Enterprise Applications Performance

- Work with the Analytics Architect at the following tuning tasks:
- Monitor Cube execution time using MicroStrategy Developer for performance tuning
- Tune the Web Session Idle time and the User connection timeout using MicroStrategy Developer to balance the Intelligence Server load overhead incurred by having too many sessions opened

Enterprise Mobile Applications Performance

- Work with the Mobile Architect at the following tuning tasks:
- Execute update Mobile Cache subscriptions to enable Pre-Caching using MicroStrategy Developer or MicroStrategy Command Manager to improve application response times and make the application available for Offline use
- Tune the Mobile Push notifications using MicroStrategy Developer to improve mobile user experience

Definitions

Term	Definition
Analytics Security Architecture	Framework, tools, and processes to ensure the security of application objects and data is protected from unauthorized viewing, tampering, or modification
Cache	Information storage for faster access and reusability
Cache Refresh Schedules	Delete, invalidate, update, and expire caches in an automated manner
Configuration Documentation	Documents that capture all Intelligent Platform configuration information
Cube Refresh Schedules	Publish, refresh, and update cubes in an automated manner
Data Connectors	Drivers that facilitate communication to database systems, data lakes, web services, and file storage locations that store data for end user analysis

Term	Definition
Distribution Services Subscriptions	Enrollment in a delivery service, including recipient users, delivery method, owner, and frequency
Operational Procedures	Step-by-step instructions for daily maintenance of the environment
Platform Analytics	Discovery and interpretation of patterns in data to analyze health and performance
Platform Configuration Protocols	Settings that define the rules and conventions for communication between components within the Platform
Platform Data Architecture	Models, policies, rules, or standards that govern which data is collected, how it is stored, arranged, integrated, and used
Platform Environment Architecture	Hosts the data, application, security, and related services necessary to deploy analytics and mobility applications throughout the Enterprise
Platform Environment Cases	Issues reported by users centered around the platform
Platform Environment Performance	Responsiveness of the Environment within set benchmarks and guidelines
Platform Environment Synchronization	Migrate objects, configurations, settings, and patches across environments
Platform Lifecycle Management	Process of maintaining the life of the Platform; including product installation, configuration, and upgrade
Platform Project Architecture	Environment in which all related reporting and analysis is performed
Platform Project Performance	Responsiveness of the Project within set benchmarks and guidelines
Platform Report	Listing of Platform health and performance KPIs
Platform Security Architecture	Framework, tools, and processes to ensure the security of an entire Platform and its data is protected from unauthorized viewing, tampering, or modification
Platform Service Architecture	Jobs executed within the Platform, including interactive and scheduled jobs

Term	Definition
Platform Services Performance	Responsiveness of the Platform Service Architecture within set benchmarks and guidelines
Platform Upgrade Procedures	Rules and guidelines detailing processes to upgrade the environment
User	Person that interacts with the Enterprise Intelligence Platform to obtain a functional benefit
User Access	Permissions and privileges that define the user's ability to perform certain actions
User Group	Grouping of users who share one or more characteristics or purpose
User Provisioning	Creation and maintenance of user accounts

Copyright Information

All Contents Copyright © 2021 MicroStrategy Incorporated. All Rights Reserved.

Trademark Information

The following are either trademarks or registered trademarks of MicroStrategy Incorporated or its affiliates in the United States and certain other countries:

Dossier, Enterprise Semantic Graph, Expert.Now, HyperIntelligence, HyperMobile, HyperScreen, HyperVision, HyperVoice, HyperWeb, Information Like Water, Intelligent Enterprise, MicroStrategy, MicroStrategy 2019, MicroStrategy 2020, MicroStrategy 2021, MicroStrategy Analyst Pass, MicroStrategy Architect, MicroStrategy Architect Pass, MicroStrategy Badge, MicroStrategy Cloud, MicroStrategy Cloud Intelligence, MicroStrategy Command Manager, MicroStrategy Communicator, MicroStrategy Consulting, MicroStrategy Desktop, MicroStrategy Developer, MicroStrategy Distribution Services, MicroStrategy Education, MicroStrategy Embedded Intelligence, MicroStrategy Enterprise Manager, MicroStrategy Federated Analytics, MicroStrategy Geospatial Services, MicroStrategy Identity, MicroStrategy Identity Manager, MicroStrategy Identity Server, MicroStrategy Integrity Manager, MicroStrategy Intelligence Server, MicroStrategy Library, MicroStrategy Mobile, MicroStrategy Narrowcast Server, MicroStrategy Object Manager, MicroStrategy Office, MicroStrategy OLAP Services, MicroStrategy Parallel Relational In-Memory Engine (MicroStrategy PRIME), MicroStrategy R Integration, MicroStrategy Report Services, MicroStrategy SDK, MicroStrategy System Manager, MicroStrategy Transaction Services, MicroStrategy Usher, MicroStrategy Web, MicroStrategy Workstation, MicroStrategy World, Usher, and Zero-Click Intelligence.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Specifications subject to change without notice. MicroStrategy is not responsible for errors or omissions. MicroStrategy makes no warranties or commitments concerning the availability of future products or versions that may be planned or under development.

The Course and the Software are copyrighted and all rights are reserved by MicroStrategy. MicroStrategy reserves the right to make periodic modifications to the Course or the Software without obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of the Course or Software without prior written consent of an authorized representative of MicroStrategy are prohibited.