

ARCHITECTURE FOR ENTERPRISE BUSINESS INTELLIGENCE

AN OVERVIEW OF THE MICROSTRATEGY
PLATFORM ARCHITECTURE FOR BIG DATA,
CLOUD BI, AND MOBILE APPLICATIONS



ARCHITECTURE FOR ENTERPRISE BUSINESS INTELLIGENCE

AN OVERVIEW OF THE MICROSTRATEGY
PLATFORM ARCHITECTURE FOR BIG DATA,
CLOUD BI, AND MOBILE APPLICATIONS



TABLE OF CONTENTS

INTRODUCTION

1	Foreword	3
2	Platform Design	7
2.1	MicroStrategy's Philosophy – Delivering an Architecture for the Long Term	7
2.2	Architectural Tenets for Enterprise Business Intelligence	8
2.3	Complete BI Functionality Delivered on an Organically-developed Architecture	10
2.4	Enterprise BI on an Integrated Backplane and a Unified Web Interface	11
2.5	Summary	14
3	Product Architecture	17
3.1	MicroStrategy has a Single Unified Architecture	17
3.2	MicroStrategy Intelligence Server	19
3.3	MicroStrategy Web	19
3.4	MicroStrategy Mobile	19
3.5	MicroStrategy Report Services	20
3.6	MicroStrategy OLAP Services	20
3.7	MicroStrategy Transaction Services	20
3.8	MicroStrategy Distribution Services	21
3.9	MicroStrategy Office	21
3.10	MicroStrategy Desktop	21
3.11	MicroStrategy Architect	22
3.12	MicroStrategy MultiSource Option	22
3.13	MicroStrategy Clustering Option	23
3.14	MicroStrategy Object Manager	23
3.15	MicroStrategy Integrity Manager	23
3.16	MicroStrategy Enterprise Manager	24
3.17	MicroStrategy Command Manager	24
3.18	MicroStrategy SDK	24
3.19	Summary	25

PLATFORM COMPONENTS

4	Metadata and the Logical Model	29
4.1	A Complete Metadata is Key for Efficient BI Processing	29
4.2	Configuration Objects	31
4.3	Data Abstraction Objects	32
4.4	Business Abstraction Objects	35
4.5	Report Components	37
4.6	Report and Document Definitions	38
4.7	Visual Analyses	41
4.8	Delivery Objects	41
4.9	Comprehensive Documentation of Metadata	43
4.10	Multi-Lingual BI Applications	44
4.11	Object Change Journaling	47
4.12	Benefits of the Metadata Design	48
4.13	Summary	52

5	MicroStrategy Intelligence Server Architecture	55
5.1	Powering the MicroStrategy Business Intelligence Platform	56
5.2	Intelligence Server Engines and Query Flow	57
5.3	Automatic Resource Allocation	59
5.4	64-bit Business Intelligence	60
5.5	Cluster-Capable Intelligence Server	60
5.6	Optimized for Heterogeneous Data Sources	61
5.7	Sophisticated Analysis	66
5.8	In-Memory Intelligent Cubes	71
5.9	Multi-Level Shared Report Caching	74
5.10	Scheduling Reports and Administrative Tasks	77
5.11	Exporting Data	78
5.12	Formatting Data	79
5.13	Simplified Central Administration	81
5.14	Summary	85
6	MicroStrategy Web Architecture	87
6.1	Core Design Principles	87
6.2	Optimize Performance with a Service-Oriented Architecture	87
6.3	Open, Layered Design Optimizes Performance, Integration, and Customization	89
6.4	MicroStrategy Web is built for Integration and Customization	90
6.5	A Single, Platform-Independent Code Base Provides Flexibility and Consistency	92
6.6	All Data in the BI Application is Secure	93
6.7	Employ Leading Edge Technologies to Maximize Ease-of-Use and Interactivity	95
6.8	Seamless Support for Enterprise Portals	97
6.9	Data Integrated with Geo-Spatial Information Systems	99
6.10	Summary	100
7	MicroStrategy Mobile Architecture	103
7.1	Core Design Principles	103
7.2	Deliver Mobile Apps in a Multi-Tiered Architecture	103
7.3	MicroStrategy Mobile App Architecture	105
7.4	MicroStrategy Mobile Server Architecture	107
7.5	All Data in the MOBILE BI APPLICATION is Secure	108
7.6	Leading Edge Technologies Maximize Ease-of-Use and Interactivity	109
7.7	Summary	110
8	Report Scheduling and Delivery	113
8.1	Designing for High-Volume Automated Report Delivery	113
8.2	Self-Service vs. Centrally-Managed Subscriptions	115
8.3	Exception Reporting	117
8.4	High Performance Report Deliveries	118
8.5	Summary	120

PLATFORM FUNCTIONALITY

9	Security	125
9.1	Securing BI Applications	125
9.2	User Authentication	126
9.3	User Authorization	128
9.4	Multi-Tier Web Architecture and Transmission Security	134
9.5	Mobile Security	137
9.6	Summary	141

10	Reliability and Fault Tolerance	143
10.1	Stateful and Stateless Architectures	143
10.2	Shared, unified metadata	145
10.3	Communication between Cluster Nodes	145
10.4	MicroStrategy Clustering Capabilities	146
10.5	Fault Tolerance with Built-in Failover	149
10.6	Adaptation to Changing Conditions	149
10.7	Summary	150
11	Scalability and Performance	153
11.1	High-Performance Initiative	153
11.2	Multi-Level Shared Caching	154
11.3	In-Memory Cubes	157
11.4	Query Optimization	159
11.5	Distributable Execution	167
11.6	64-bit Business Intelligence	169
11.7	Multi-Threaded Processing	170
11.8	Efficient Communications	172
11.9	Performance Tuning	174
11.10	Proven Scalability and Performance	174
11.11	Summary	175
12	Closed-Loop Business Intelligence	177
12.1	Closed-Loop Applications Reduce the Time to Act	177
12.2	Closed-Loop BI Using MicroStrategy Transaction Services	178
12.3	Enabling transactional functionality in MicroStrategy	180
12.4	Closed-Loop BI Using Dynamic Datamarts	183
12.5	Summary	185
13	Data Mining and Predictive Analysis	187
13.1	Integrating Data Mining Into Business Intelligence	188
13.2	Data Mining Services in MicroStrategy	190
13.3	Data Mining Algorithms	191
13.4	Benefits of Integrating Business Intelligence and Data Mining	198
13.5	Summary	199
14	Heterogeneous Data Access	201
14.1	Data is Stored in Many Different Places	201
14.2	Relational Data Sources	202
14.3	Multi-Dimensional Sources	205
14.4	Data Access to Other Data Sources	206
14.5	Joining Data from Heterogeneous Data Sources	208
14.6	Summary	211
15	Exporting Data and Reports	213
15.1	MicroStrategy Supports Flexible Exporting for All Uses	213
15.2	MicroStrategy Provides Efficient Exporting of Data	214
15.3	1.3 MicroStrategy Office Integrates With Excel, Word, and PowerPoint	217
15.4	Summary	220
<hr/>		
USER INTERACTIVITY		
16	End User Experience	225
16.1	Consistent Reports, Functionality, and Security through Many User Interfaces	225
16.2	The Business Intelligence Application Spectrum	230
16.3	MicroStrategy User Experience Design Tenets	231

16.4	Dashboards and Scorecards	234
16.5	Visual Insight Analysis	236
16.6	Transactions	237
16.7	Enterprise reporting	238
16.8	Ad-hoc Reporting and OLAP Analysis	239
16.9	Advanced and Predictive Analysis	245
16.10	Alerts and proactive notifications	246
16.11	User Functionality by User Role	247
16.12	Customizing MicroStrategy Functionality for the end user	250
16.13	Summary	250
17	Developer Experience	253
17.1	Multiple Interfaces for BI Development	253
17.2	MicroStrategy Architect	253
17.3	Map Database Structures to Qualitative and Quantitative Business Terms	257
17.4	MicroStrategy Desktop	260
17.5	Business Abstraction Objects	261
17.6	Report Objects	265
17.7	Report creation using the dynamic SQL engine	265
17.8	Creating Reports from Intelligent Cubes	267
17.9	Build and Execute Reports with the Dynamic MDX Engine	268
17.10	Operational Reports using the Freeform SQL engine	269
17.11	1.11 Transaction Reports	270
17.12	Building Graphs in MicroStrategy Desktop	270
17.13	Data Mining and Predictive Reports in MicroStrategy Desktop	271
17.14	Report Services Document Creation	272
17.15	Advanced Visualizations	274
17.16	Creating Dashboards and Reports in Web	276
17.17	Creating Pixel-Perfect Dashboards and Scorecards over the Web	281
17.18	Transaction Documents	283
17.19	Customization / SDK Developer	285
17.20	Summary	286
18	Administrator Experience	289
18.1	In-depth Administration across the Whole Platform	289
18.2	Real-Time Monitoring and Control	290
18.3	Error Logging and Diagnostic Files	292
18.4	MicroStrategy Enterprise Manager	293
18.5	Centralized Administration	294
18.6	Change Journaling	297
18.7	MicroStrategy Command Manager	300
18.8	MicroStrategy Object Manager	303
18.9	Integrity Manager	307
18.10	MicroStrategy Web Administration	310
18.11	MicroStrategy Mobile Administration	313
18.12	Health Center	315
18.13	System Management Software Integration	316
18.14	Summary	317

IMPLEMENTING BI APPLICATIONS

19	Developing Business Intelligence Applications	321
19.1	Ad-Hoc Data Discovery	321
19.2	Application Project Lifecycle	323

19.3	Evolve from Departmental to Enterprise BI Applications	326
19.4	Estimating the Capacity Requirements of the BI Application	330
19.5	Summary	334
20	Portable Analytic Modules	337
20.1	Rapid Application Development Framework	337
20.2	Portable Analytic Applications	338
20.3	MicroStrategy BI Developer Kit Analytic Modules	339
20.4	Summary	347
21	Extensibility and the SDK	349
21.1	Access All BI Functionality in a Service-Oriented Architecture with Open APIs	349
21.2	MicroStrategy Web API	351
21.3	Task Framework	356
21.4	MicroStrategy Mobile API	357
21.5	MicroStrategy Visualization API	358
21.6	MicroStrategy Office API	360
21.7	MicroStrategy Intelligence Server API	361
21.8	Comprehensive Documentation That Describes the API in Detail	361
21.9	Portal Integration	363
21.10	Web Services Development	363
21.11	Sample Applications	365
21.12	Customization and Extension Examples	367
21.13	Summary	373

APPENDICES

Appendix A: Schema Support	377
Appendix B: Very Large Database (VLDB) Properties	381
Appendix C: Analytical Functions	389
Appendix D: Supported Graph Types	392
Appendix E: Performance Counters and Key Performance Indicators	394
Appendix F: Supported Data Sources	402
Appendix G: Operating Platforms	405
Appendix H: Security Privileges	410
Appendix I: API Classes, Methods, Properties, and Interfaces	412



INTRODUCTION

1

FOREWARD**Partnering With Our Customers to Achieve Enterprise Business Intelligence**

The MicroStrategy® architecture that you see today is the culmination of roughly 5,000 man-years of engineering effort and well over half a billion dollars in engineering investment. More significant is the fact that the MicroStrategy architecture embodies the cumulative engineering advances gained through over 20 years of experience in supporting the most demanding BI applications in the world.

Our customers drive our technology development. Ever since MicroStrategy's first customer implementation, we have been challenged to solve the most technically difficult problems in BI. After countless implementations for over thousands of customers, we fully understand the value of a close partnership of co-development and BI innovation that is continually fueled by our customers' uniquely aggressive and ever-expanding requirements. The effort we have invested in supporting our customers' initiatives and applications has propelled our software faster and farther than any competing BI technology on the market.

Very early we realized that our customers' BI requirements would continue to grow as their data volumes and user populations increased, and as their business requirements became broader and sophisticated. In fact, it became clear to us that our customers were driving the technology toward the goal of "Enterprise-wide Business Intelligence." We took this goal to heart when we designed our new product architecture – beginning with MicroStrategy 7™ in 2000, and evolving to MicroStrategy 8™ by 2005 and significantly expanding our platform capabilities with MicroStrategy 9™ in 2009.

Nearly every BI vendor now claims to have an "integrated BI architecture" suitable for Enterprise BI. What is often overlooked is the fact that MicroStrategy's end goal was not integration – it was the starting point. We view integration not as a feature but as a requirement of a scalable, maintainable, and high performance architecture. Any architecture with duplicative or redundant metadata, data structures, processes, queues, caches, and thread management facilities has inherent performance inefficiencies, is difficult to manage, and will be more costly to maintain. Given that performance and scalability have always been the dominant limiting factors to successfully deploy enterprise systems, an architecture designed from the start for Enterprise BI was needed – not one forcibly mashed together from a set of existing products and legacy code.

As such, we determined that we needed to discard every line of code in our previous architecture (version 6 and prior) and start over from scratch. While this business decision was considered risky at the time, the strategy was necessary if we wanted to meet the long-term needs of our customers. It allowed us to start by designing an architecture built specifically for performance and maintainability. Additionally, it allowed us to build new features as part of an integrated architecture, rather than worry about integrating legacy products after the fact. Interestingly, none of the competitive BI technologies followed our approach. Instead, they decided to make incremental gains towards integration, and chose architectures built primarily to achieve legacy integration rather than for performance and flexibility. In retrospect, our strategy has turned out to have extraordinarily beneficial consequences for both MicroStrategy and our customers.

We have continued to innovate and advance the BI platform to keep up with our customers' and the market's requirements. Full-featured business user and developer Web interfaces, dashboards, data

discovery, visualizations, in-memory cubes, mobile BI, Office integration, data input and write-back, and advanced analytics and data mining have been added. The result of this 20 year journey has resulted in a mature, Industrial-Strength Business Intelligence™ platform that is available as on-premises installations or on MicroStrategy's Cloud infrastructure.

Our architecture can be compared to an iceberg where 90% of the mass lies under the surface, hidden from casual view. Often, our customers do not begin to appreciate the full depth and thoughtfulness of our product's capabilities until they have used the technology for several years, and grown their BI implementation into a world-class application beyond their initial expectations and vision.

We offer this book as a means of formally presenting our architecture – coded by our engineers and co-developed by our customers for over 20 years. I hope the content will help provide a level of detail that is lacking in most BI literature today, and provide some clarity as to what you can achieve with the MicroStrategy Platform. We believe that we have a unique offering and are confident that as you dig into the detail, you will conclude the same.

A handwritten signature in black ink that reads "Jeff Bedell". The signature is fluid and cursive, with "Jeff" on the left and "Bedell" on the right, connected by a single stroke.

Jeff Bedell
Chief Technology Officer, MicroStrategy

2

PLATFORM DESIGN

Since 1989, MicroStrategy's engineers and designers have methodically tackled, and solved the hardest technical problems in business intelligence – delivering the greatest user interactivity, against the largest databases, with the best response times, containing the most sophisticated analytics, and accessed by the largest number of users. Today, MicroStrategy technology supports the most demanding BI applications across all industries.

2.1 MICROSTRATEGY'S PHILOSOPHY – DELIVERING AN ARCHITECTURE FOR THE LONG TERM

MicroStrategy was designed specifically to usher in a new era in business intelligence, with an architecture that is organically integrated and expressly designed for the Enterprise, Mobile, or Cloud. All components of MicroStrategy's architecture are the result of organic growth from a single architectural baseline. The integrated BI architectures from other vendors are not organic, but are the result of integrating proprietary BI technologies, some of which are more than 10 years old.

The MicroStrategy Business Intelligence Platform Enables Standardization of Enterprise BI Applications

Business Intelligence applications are an integral part of an organization's fabric providing critical information to various departments. As the number of departmental business intelligence applications, powered by different technologies, in an organization grow, the IT departments face a challenge to manage and maintain them. Ultimately, an organization will advance from having isolated BI applications towards BI standardization.

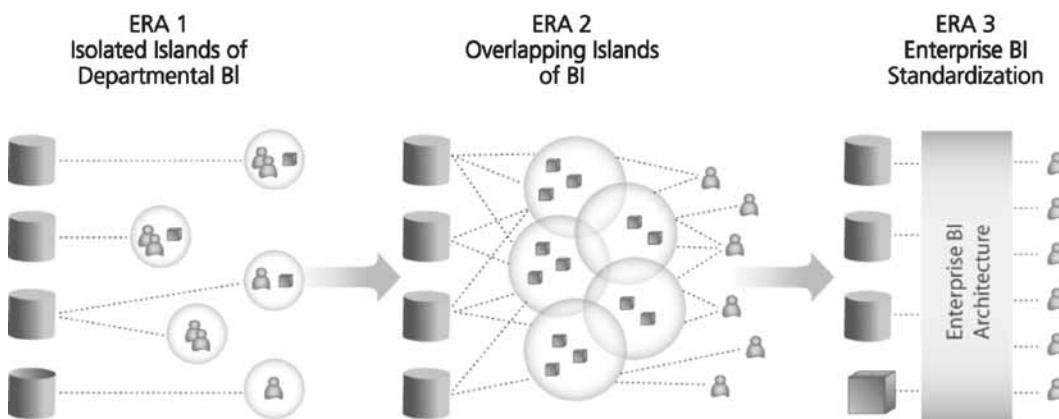


Figure 2-1 MicroStrategy's organically integrated architecture was specifically designed to usher in a new era of enterprise business intelligence, and help enterprises evolve from previous eras of islands of disparate BI technologies and inconsistent departmental BI applications.

- Era 1: Isolated Departmental Islands of BI Are an Initial Success

Many companies deploy BI applications as departmental solutions, and in the process, have accumulated a large collection of disparate BI technologies as a result. Each distinct technology supported a specific user population and database, within a well-defined “island of BI.” At first, these islands of BI satisfied the initial needs of the business, but early success in departmental deployment sowed the seeds for new problems as the applications grew.

- Era 2: Overlapping Disparate Islands of BI Have Become an Enterprise Liability

Successful applications always expand. The second era of BI is hallmark by BI applications that have expanded to the point where they are no longer isolated islands. Instead, they overlap in user populations, data access, and analytic coverage. As a result, organizations are now faced with an untenable situation. The enterprise is getting conflicting versions of the truth through the multiple disparate BI systems, and there is no way to harmonize them without an extraordinary ongoing manual effort of synchronization. Equally problematic is the fact that business users are forced to use many different BI tools depending on what data they want.

- Era 3: Enterprise BI Standardization Delivers a Single Version of the Truth with Lowest Total Cost of Ownership

The third era of business intelligence is one where a single BI architecture delivers one version of the truth through a single user interface to all people across the enterprise. It can access all of the data, administer all of the people, eliminate repetitive data access, reduce the administrative effort, and reduce the time to deploy new BI applications. This is not just a vision of Enterprise Business Intelligence. Many companies are successfully achieving this new era of business intelligence with MicroStrategy technology.

The bedrock of MicroStrategy’s architecture philosophy is to deliver a living architecture – one that can continually evolve to greater functionality, greater scalability, with seamless integration, and ever greater economies of scale. Adhering to this basic philosophy often entails making difficult technical choices where the tradeoff is between expediency (building something quickly just to satisfy the immediate market requirement) vs. long-term flexibility (designing something with primary emphasis on extensibility and maintainability over time). This philosophy has ensured that we can easily extend or expand our platform, and adapt to any new trend that drives the business intelligence market, whether it is Cloud, Mobile, or Enterprise Business Intelligence.

Unlike software companies that are driven by the expediency of day-to-day market trends, MicroStrategy has never wavered from its commitment to building a long-term enterprise-caliber technology. In the end, organizations will make a choice of their preferred BI deployment path – Enterprise BI, Cloud, Mobile or any other – but they will need a BI platform that delivers high performance, flexibility and adaptability that will guarantee success in any path they choose. MicroStrategy firmly believes that our platform has passed the test of time, and that we are in a unique position to enable organizations to successfully deploy a high performance, well-rounded BI application either on premises or in the Cloud.

Throughout this book there are many examples of where MicroStrategy seemingly chose the most technically difficult route. The reason is that our designs invariably address far more rigorous requirements than are immediately apparent. In fact, every design decision we take must explicitly adhere to architectural tenets for long-term flexibility and Enterprise BI usage.

2.2 ARCHITECTURAL TENETS FOR ENTERPRISE BUSINESS INTELLIGENCE

1. Scalability and High Performance

All design decisions must ensure that MicroStrategy can deliver consistently high performance as the system scales upward, and must anticipate order-of-magnitude growth beyond today’s state-of-the-art standard (user scale, data scale, and application scale).

2. Economies of Scale

All design decisions must explicitly deliver greater economies of scale as a system grows – using techniques like in-memory data processing, caching, object reuse, automated administration, and collaborative analytics. We make sure that MicroStrategy implementations require an absolute minimum of IT personnel, require the fewest servers, and minimize the workload on expensive database resources.

3. Complete Functionality

All design decisions must ensure that the MicroStrategy architecture offers the full range of BI functionality on a single services-oriented architecture, so that customers can satisfy all of their BI requirements without the need for additional integration work.

4. Incremental Growth

All design decisions must ensure that customers can incrementally grow their BI infrastructures – from small to large, from departmental scope to enterprise scope, from isolated islands to consolidated applications, and from reporting to dashboards to OLAP to ad hoc analysis to alerting to Mobile apps. This allows our customers to initially buy just the functionality they need, and to incrementally grow their BI solution as their requirements naturally expand. We have designed our plug-n-play components so that each added component brings new functionality to all pre-existing components and builds on pre-existing content so that no rework is required.

5. Openness and Extensibility

All design decisions must ensure that MicroStrategy's vast functionality continues to be fully accessible through Web services APIs. The degree of commitment MicroStrategy has toward its APIs is best illustrated by the fact that MicroStrategy technology internally uses the same thousands of APIs that we expose externally in our SDK documentation.

6. Centralized Consistency with Distributed Governance and Self-service

All design decisions must support the goal of a consistent single version of the truth throughout the enterprise using a single shared metadata and pervasive security architecture. Yet, the architecture must also provide a high degree of autonomy to distributed development teams (managed by departments and divisions), and to individual users allowing them to create enterprise-consistent solutions at a local level.

7. Rapid Development and Deployment

All design decisions must promote rapid development and deployment of new reports and applications. MicroStrategy has invested significant engineering energy in creating a vast array of reusable metadata objects, by creating a security architecture that is applied automatically and pervasively with no effort on the part of the report designer, and through WYSIWYG design paradigms that allow novices and experts to play a role in accelerating the report design process.

8. Consistent Experience

MicroStrategy continuously works towards providing the same BI experience from any user interface – desktop, web browsers, or mobile devices. Any feature, however simple it may be, is added to the platform in a way that it can be easily available from any user interface. This philosophy enables business users to seamlessly change their interface to access critical business reports without losing any functionality.

These requirements do not come from MicroStrategy; they are requirements from our long-time customers who originally invested in MicroStrategy technology to support their high-scale BI applications. Today, the goal of these same customers is to host many diverse BI applications on a Cloud instance, or make these applications Mobile. The architectural tenets discussed earlier ensure their long-term success in this process.

2.3 COMPLETE BI FUNCTIONALITY DELIVERED ON AN ORGANICALLY-DEVELOPED ARCHITECTURE

The 1990s represented a formative period for business intelligence applications and products. During this period, organizations realized that they needed a range of BI capabilities to satisfy a diverse set of user needs. Ultimately, this range could be grouped into five distinct categories or “Styles of BI” applications. The 5 Styles of BI applications include:

1. Data Mining and Advanced Analysis
2. Visual and OLAP analysis
3. Enterprise Reporting
4. Dashboards and Scorecards
5. Mobile Apps and Alerts

In response to this fragmented market demand, most BI vendors developed different products, each designed to support one of the 5 Styles of BI.



Figure 2-2 The MicroStrategy architecture offers all 5 Styles of business intelligence with a single organically-developed architecture.

Companies adopted a variety of BI products, and built BI applications where each application was centered on a single style of BI. Today, companies have concluded that the boundaries between Styles of BI (e.g., the boundaries between reporting and scorecards and analysis) are not natural business boundaries, but are artifacts of BI product limitations. In the ideal world, all BI products would be able to seamlessly support any or all 5 Styles of Business Intelligence.

MicroStrategy Delivers All 5 Styles of Business Intelligence

Beginning in 1996, MicroStrategy embarked on a massive software development effort to re-engineer its highly scalable architecture so that it could support all 5 Styles of Business Intelligence. By 2000, MicroStrategy delivered MicroStrategy 7 which laid the foundation for this vision, and explicitly supported the two most difficult styles of BI: 1) Advanced Analysis, and 2) Alerting. By mid-2002, MicroStrategy added

In-Memory to the platform and further enhanced this in 2009. In 2003, MicroStrategy added Enterprise Reporting and Scorecards & Dashboards. Data Mining was added in 2005. MicroStrategy released its Mobile App for Apple iOS phones and tablets in 2010 and for Google Android devices in 2011.

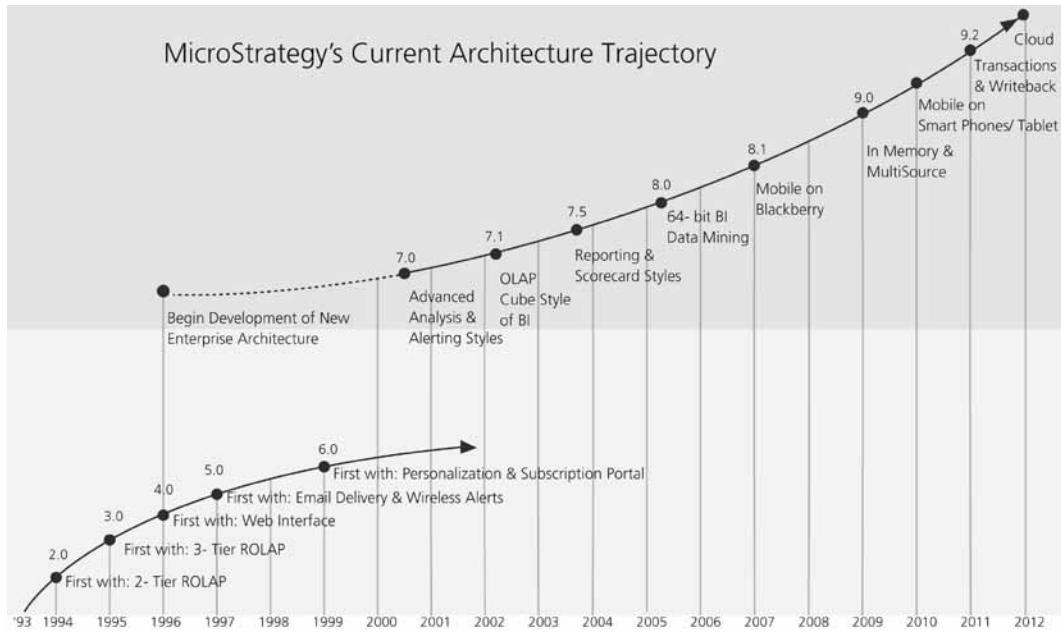


Figure 2-3 Despite a series of market innovations on MicroStrategy's original software architecture, MicroStrategy embarked on a total re-engineering of the software in 1996 to produce an architecture capable of supporting all 5 styles of business intelligence.

2.4 ENTERPRISE BI ON AN INTEGRATED BACKPLANE AND A UNIFIED WEB INTERFACE

The modern MicroStrategy architecture delivers all 5 Styles of business intelligence on a unified backplane of common services and through a single Web and Mobile architecture.

Service Oriented Integrated Backplane

The integrated backplane is the heart of the MicroStrategy architecture. It provides the common services of metadata, prompt generation, scheduling, in-memory data, shared caching, security, user management, query generation, query governing, and administration. More importantly, it is the core engine which supports each of the 5 Styles of BI as plug-n-play “service modules” that can be mixed and matched in any combination. These modules include MicroStrategy Report Services for pixel perfect reporting, Mobile apps, dashboards and scorecards, MicroStrategy OLAP Services for speed-of-thought slicing and dicing, MicroStrategy Distribution Services for alerting proactive notification functionality, and MicroStrategy Transaction Services for data input and write-back capabilities. MicroStrategy ROLAP and analytics are always an integral part of the core engine.

MicroStrategy customers can add various “service modules” to the core engine to incrementally extend its functionality. Each service module is designed to build on one another, adding new functionality to each other when used in combination. This means that a customer can add the OLAP Services module to the engine, and all previously-built grid reports automatically become Intelligent Cube™ reports, and inherit a wide range of new functionality. In addition, a customer can add the Report Services module, and be able to re-use all the previously-built grid and cube reports as datasets for the new pixel-perfect document. Furthermore, the Transaction Services module can be added to enable data input and write-back functionality on Report

Services documents and dashboards. And finally, it means that a customer can add the Distribution Services module, and immediately use any grid reports, intelligent cube reports, and document reports as the basis for e-mail alerts and proactive notifications.

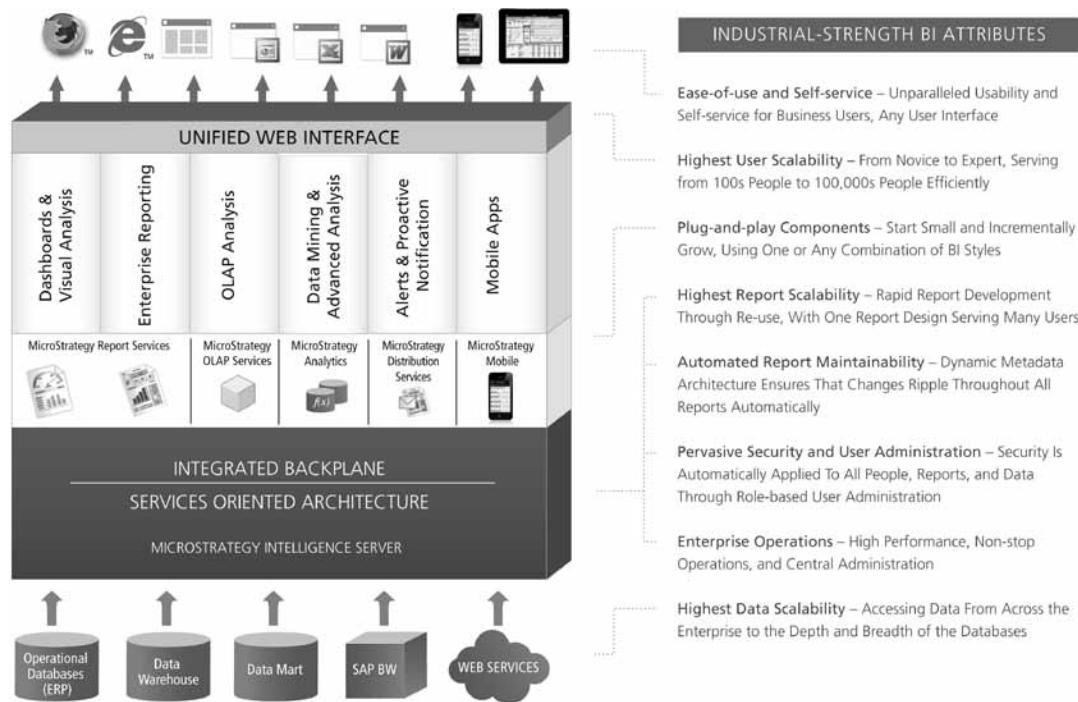


Figure 2-4 MicroStrategy is a uniquely unified architecture – delivering all styles of BI through a unified user interface and on a unified backplane of metadata and administration.

MicroStrategy's Unified Web Interface

MicroStrategy's unified Web interface is a single code base that provides BI functionality to all users, ranging from basic reporting functionality for novice users, to interactive analysis functionality for analysts, to full report design functionality for power users and IT professionals. Because of the way it is designed, the MicroStrategy Web interface can deliver BI functionality equally well to multiple different interfaces, including: different Web browsers, third-party enterprise portals, and even to the Microsoft® Office applications of Excel®, Word, and PowerPoint® using Web services.

From a user point of view, MicroStrategy's Web interface is a truly unified interface. It presents the same screen layout, the same drag-n-drop actions, the same menus, and the same toolbars regardless of user skill level and regardless of which style of BI a person is using. By contrast, competing “integrated Web interfaces” from other vendors are hardly unified at all. They present different screen layouts, different drag-n-drop actions, different reports from different metadata repositories, and different drop-down menus for each of their various “integrated interfaces” because many of those interfaces were acquired or were developed independently of one another over an extended period of time.

MicroStrategy Provides Organizations with Business Intelligence on Mobile Devices

The Mobile revolution has not only affected individuals but has also transformed how business is conducted. It is now universally accepted that business users will use mobile devices to conduct a large portion of their business tasks. Mobile devices provide businesses with the opportunity to take immediate decisions 24x7 and from any location. Mobile is going to expand the number of business intelligence users from 1,000s to 100s of thousands and will mandate that an organization possess a robust, scalable and high-performance Mobile Intelligence platform. MicroStrategy's more than 20 years of experience supporting demanding BI applications has enabled us to create a platform capable of supporting thousands of mobile users that will deliver critical business intelligence at any hour and location with highest performance.

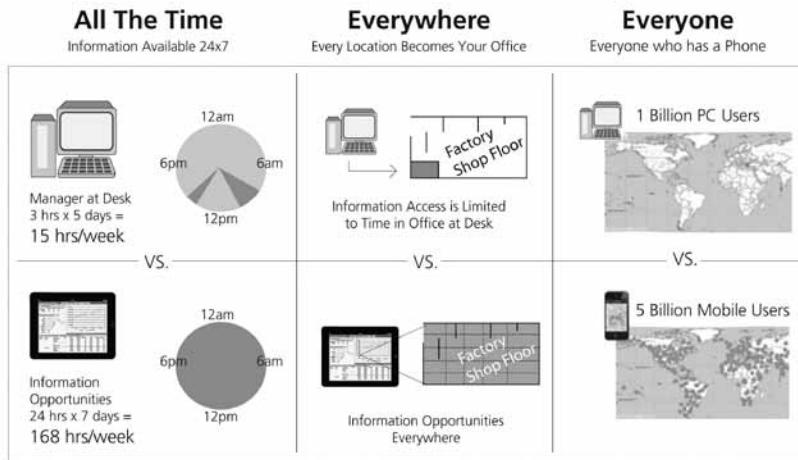


Figure 2-5 MicroStrategy's business intelligence platform enables organizations to take advantage of the enormous opportunities created by the Mobile revolution by making business information accessible all the time, everywhere and to anyone for immediate decision making.

Deploy BI Applications in the MicroStrategy Cloud

A critical trend that is clearly influencing business is the migration of traditional IT client-server solutions to the Cloud. As Cloud technology matures, it is inevitable that a large number of companies will offload some aspect of their BI applications from an on-premises solution to a public or private Cloud. MicroStrategy's BI platform is designed to be deployed seamlessly in the Cloud, enabling IT departments to create hosted BI applications without losing any BI functionality. A MicroStrategy based BI application, even when it's hosted on a cloud, will provide business users with all BI functionality through mobile devices or web browsers.

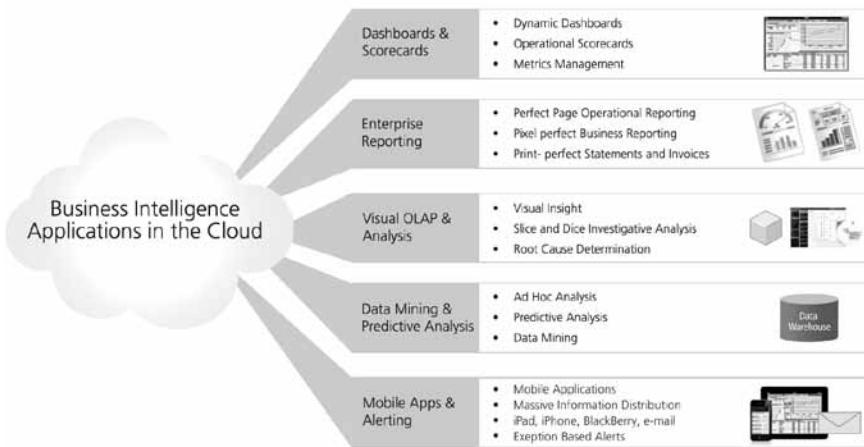


Figure 2-6 All features provided by the MicroStrategy Business Intelligence platform are available from a Cloud deployment to business users.

2.5 SUMMARY

MicroStrategy uniquely offers the only enterprise-class business intelligence technology suitable for the most demanding BI applications, suitable to deliver all 5 Styles of BI seamlessly on-premises or in the Cloud, delivered via mobile devices or web browsers, suitable to be the BI standard across the entire enterprise. The intent of this book is to justify this statement. To do so, this book will discuss in detail all aspects of MicroStrategy's elegant platform architecture. Organizationally, this book approaches the MicroStrategy platform as one might approach beginning development of a business intelligence application – starting at the infrastructural components, and working out to a finished application in production.

3

PRODUCT ARCHITECTURE

The MicroStrategy business intelligence platform is the culmination of over 20 years of BI experience. MicroStrategy offers the only enterprise-class BI technology suitable for the most demanding BI applications, suitable to deliver all 5 Styles of BI seamlessly, and suitable to be the BI standard across the entire enterprise. This enterprise-class technology ensures:

- Complete range of business user functionality — organically integrated scorecards, dashboards, reporting, analysis, alerts, and write-back actions.
- Depth of analytical power — full investigative analysis of the entire data warehouse, not just subsets of data, including data mining, predictive, statistical, financial, and mathematical analysis.
- Comprehensive administrative control — consistent with mission-critical system operation.
- Limitless application extensibility — open APIs exposed through industry standard Web services, Java™, XML/XSL, xHTML, AJAX, CSS, Flash, ActionScript, Objective-C, and C++ to customize applications and integrate with other systems.

3.1 MICROSTRATEGY HAS A SINGLE UNIFIED ARCHITECTURE

The MicroStrategy platform is built on one, unified, organically-developed architecture. To achieve the eight design tenets for Enterprise BI, the underlying architectural design establishes:

1. A single, unified object model to define and construct objects that represent any business.
2. An organic, platform of 17 distinct products that dynamically assemble, and re-assemble these objects in response to any business question or user action.

All platform products use object-oriented programming techniques of inheritance and encapsulation ensuring scalability and high performance.

The MicroStrategy object model is the genius of the MicroStrategy platform. The MicroStrategy metadata is the manifestation of the object model. The metadata contains the building blocks or objects necessary to represent an enterprise's business. The metadata stores these objects in a database for efficient re-use, manageability, and performance. The objects themselves are used to create new objects such as a report definition. Hence, if an object changes, every other object dependent on it also changes. This ensures consistency across business definitions, and actually minimizes the number of objects created, stored, and managed. What truly makes the MicroStrategy metadata unique is that it only stores these objects; it does not store a finished report complete with a static SQL statement. Because of this, the MicroStrategy metadata gives the MicroStrategy platform unparalleled analytical, functional, and administrative flexibility.

The MicroStrategy BI platform dynamically assembles the metadata objects to create reports, scorecards, dashboards, analyses, and alerts. The platform provides all 5 Styles of BI uniformly through a wide variety of user interfaces, including mobile phones, tablets, Web browsers, Enterprise Portals, Microsoft Office, E-mail, Microsoft Windows® workstations, network printers, and file servers. The MicroStrategy platform comprises of tightly integrated products that deliver unmatched functionality, power, control, and extensibility on enterprise-class technology.

Architectural Functionality	MicroStrategy Products
Mobile Applications with Transaction Write-back	Mobile, Intelligence Server, Report Services, Transaction Services
Reports, Scorecards, Dashboards, OLAP, Advanced, and Ad-Hoc Analysis, and Alerts and Proactive Notification	Intelligence Server, Web, Distribution Services, Report Services, OLAP Services
Business User Interactivity	Web, Office, Report Services, OLAP Services, Intelligence Server, Desktop
Batch Reporting	Distribution Services, Intelligence Server, Report Services, OLAP Services
Application Development (Metadata Object creation)	Architect, Desktop, Web, Intelligence Server
Administration	Command Manager, Enterprise Manager, Integrity Manager, Object Manager, Desktop, Intelligence Server
Integration With Other Applications	SDK, Office, Transaction Services

Figure 3-1 The MicroStrategy core products associated with the architectural functionality they provide. Additional products may be required depending on the precise functionality needed.

Every MicroStrategy BI application requires at a minimum:

1. Development products to build business abstraction objects in the metadata: MicroStrategy Architect and Desktop.
2. Core server to manage users, security, database communications, report formatting, and distribution: MicroStrategy Intelligence Server.
3. Interface products to run reports and manipulate the results for investigative analysis: MicroStrategy Mobile, Web, Office, or Distribution Services

Depending on the application demands, any number of products can be used together to scale to large business user populations. For example, MicroStrategy Distribution Services can distribute reports directly to business users via E-mail who simply need dashboards or static reports and not the full interactivity of MicroStrategy Mobile or Web. MicroStrategy Office is particularly popular with finance departments who like to do all their work in Microsoft Excel.

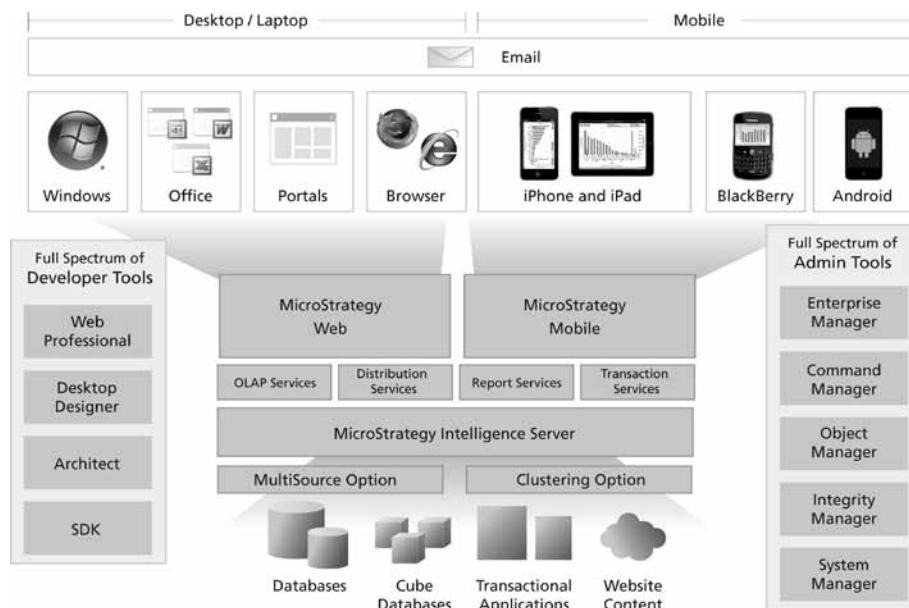


Figure 3-2 The MicroStrategy product architecture provides the full range of functionality, power, control, and extensibility on the industry's only enterprise-class BI technology. All business user interactivity is uniformly expressed through a Mobile devices, tablets, Web browsers, enterprise portals, Microsoft Office, Windows workstations, E-mail, network printers, and file servers.

Certainly mid to large BI environments would benefit from MicroStrategy's administration products – Enterprise Manager, Command Manager, Integrity Manager, and Object Manager – to centrally monitor, and manage the entire business intelligence infrastructure. If the BI applications require sophisticated closed-loop, actionable BI reports, MicroStrategy Transaction Services lets business users initiate transactions with backend systems and write back to databases and web services. The products that comprise the MicroStrategy platform are briefly described next.

3.2 MICROSTRATEGY INTELLIGENCE SERVER

MicroStrategy Intelligence Server™ is the architectural foundation of the MicroStrategy platform. As the central contact point to the metadata, Intelligence Server dynamically assembles the metadata objects to create optimized, multi-pass SQL queries for every major relational database, HiveQL queries for Hadoop distributions, and MDX queries for multidimensional data sources. Intelligence Server retrieves the data, performs any additional analytical calculations not available in the databases, formats the report, and delivers the reports to business users via MicroStrategy Mobile, Web, Office, Desktop, or Distribution Services.

Intelligence Server is a highly scalable, parallel-processing, self-tuning analytic server. Intelligence Server manages high performance interactions accessing terabytes of data by tens of thousands of users using in-memory Intelligent Cubes, caching, load balancing, resource prioritization, and connection pooling. It accesses and joins data from multiple data sources, such as data warehouses, operational databases, multidimensional (cube) databases, and even Web services and flat files. Intelligence Server also manages users, system security, data security, and user functionality access. A clustering option is available with Intelligence Server that increases scalability, and provides fault tolerance with automatic failover.

3.3 MICROSTRATEGY WEB

MicroStrategy Web™ provides a powerful and user-friendly environment for interactive analysis through any Web browser. Suitable for all levels of user sophistication from beginner to advanced, MicroStrategy Web contains report and dashboard viewing, formatting, exporting, pivoting, sorting, drilling, and ad hoc querying to WYSIWYG design and creation. Using advanced Web technology including xHTML, CSS, AJAX, Flash, and JavaScript, MicroStrategy Web delivers a highly interactive user experience spanning the full range of MicroStrategy functionality.

MicroStrategy Web accomplishes all of its functionality through a cookie-less, zero-footprint Web client without using ActiveX® or Java Applets resident in or downloaded to the Web browser. Thus, MicroStrategy Web works seamlessly through all standard Web firewalls making deployment straightforward and minimizing administration overhead.

MicroStrategy Web provides the Web interface using Active Server Pages (ASP) running Microsoft Internet Information Service (IIS) and using Java Server Pages (JSP) on either 32-bit or 64-bit processor computers running any supported application servers on any operating system.

3.4 MICROSTRATEGY MOBILE

MicroStrategy Mobile™ puts business reports, KPIs, documents, and dashboards in the hands of your employees and customers. MicroStrategy Mobile lets decision makers run an organization wherever they are and view business data on the iPad, iPhone, and Android smartphones and tablets in multi-touch tables, graphs, charts, visualizations, maps, documents, and dynamic dashboards. It securely connects to a MicroStrategy Mobile Server which manages the transmission of interactive reports, documents, and dashboards to the Mobile client applications.

MicroStrategy's platform for mobile apps enables organizations to build a wide variety of essential mobile apps:

- Business intelligence – view corporate data through interactive, visual dashboards
- Transactions – enact data-driven decisions from BI data, or input information for surveys and operational systems
- Multimedia content – distribute presentations, brochures, and videos to employees, customers, and partners

3.5 MICROSTRATEGY REPORT SERVICES

MicroStrategy Report Services™ is a plug-n-play extension to Intelligence Server providing pixel-perfect, print-perfect, and page-perfect app, dashboard, visual analysis, and report design and distribution through MicroStrategy Mobile, Web, Office, Desktop, and Distribution Services. Report Services delivers the most exacting report formatting for any of the following:

- Mobile Applications – highly interactive apps that deliver business intelligence, transactions, and multimedia content to the mobile workforce
- Dashboards and Scorecards – highly visual, interactive, pixel-perfect displays that provide “at-a-glance” view of the enterprise using gauges, dials, KPIs, and visualizations
- Visual Insight – visual exploration of data with a large library of interchangeable visualizations and speed-of-thought filtering to help you spot outliers and anomalies in your data quickly
- Enterprise Reports – classic production reporting requiring print-perfect layout with data organized and aggregated into hierarchies or bands of increasing finer detail
- Invoices and Statements – page-perfect layouts designed for billing applications and statutory reporting
- Business Reports – any report format, usually combining graphs, detail data, and often explanatory text, used to describe business performance

Report Services reports, dashboards, scorecards, analyses, invoices, statements, and business reports have fully interactive analytical, WYSIWYG design, and transactional capabilities over a zero-footprint Web browser and on Mobile devices.

3.6 MICROSTRATEGY OLAP SERVICES

MicroStrategy OLAP Services™ is a plug-n-play extension to Intelligence Server that adds in-memory data functionality to the standard ROLAP functionality of the MicroStrategy BI platform. OLAP Services creates and manages Intelligent Cubes, a multi-dimensional cache structure that speeds up access to the data your users use most. OLAP Services exposes the report objects (metrics and attributes) in the Intelligent Cube so that analysts can manipulate report objects, create derived metrics, group rows, and modify filter criteria.

One frequent user action is to drill down into the data. With OLAP Services, the Intelligent Cubes can contain attributes commonly requested via drilling. This greatly enhances the report performance because OLAP Services can return results without querying the data warehouse. OLAP Services allows seamless drill through to the data warehouse to investigate the full breadth and depth of the corporate data assets. Ad-hoc reports and drilling dynamically use Intelligent Cubes wherever possible, maneuvering between in-memory cubes and the data warehouse to retrieve the information requested.

3.7 MICROSTRATEGY TRANSACTION SERVICES

MicroStrategy Transactions Services™ is a plug-in extension to Intelligence Server that provides write-back capabilities from Report Services documents. The predominant use is in Mobile BI Apps, and it also works

from Web-based DHTML documents. It writes-back to ERP and other operational systems via Web services using XQuery; and to databases via freeform SQL.

MicroStrategy Transaction Services provides users the flexibility of designing their own freeform Transaction reports and hand-picking the data to perform each transaction. MicroStrategy Transaction Services allows users to perform any of the following three actions: update data, insert new information, or delete existing records.

3.8 MICROSTRATEGY DISTRIBUTION SERVICES

MicroStrategy Distribution Services™ is a plug-and-play add-on to Intelligence Server that proactively distributes personalized reports and alerts. In contrast to MicroStrategy Web's interactive BI environment, Distribution Services delivers interactive Flash dashboards and static reports via E-mail, folders, or printers. Often, the largest number of users simply needs to receive this type of corporate content. Those reports, however, must be sent automatically based on one or more of the following:

- A time schedule such as every Monday morning at 8 A.M.
- An event such as a completed database load
- A trigger such as business metric tracking outside an acceptable range

Serving these large user communities, Distribution Services can deliver many thousands of messages per hour. Because it is critical these users receive this information without having to request it, Distribution Services automatically sends those reports to a variety of devices, including E-mail, file servers, and networked printers. Business users can also subscribe to receive Distribution Services' proactive notification and alerts from the MicroStrategy Web interface.

The proactive notifications and alerts Distribution Services sends include the precise report formatting for the desired output device. The report can also include multiple attachments including:

- Interactive dashboards in Flash format
- Enterprise documents or scorecards in PDF format
- MicroStrategy reports in Microsoft Excel workbooks
- Large datasets in CSV format
- Zip files containing any of the above

3.9 MICROSTRATEGY OFFICE

MicroStrategy Office™ delivers MicroStrategy reporting and analysis to Microsoft Excel, PowerPoint, and Word using Web services. This is ideal for business users who want to use Microsoft Office for their BI interface. Excel, PowerPoint, and Word become live repositories of MicroStrategy scorecards, reports, and charts, all of which are continually linked back to Intelligence Server ensuring 100% data consistency across the enterprise.

Additionally, MicroStrategy Office offers outstanding offline analysis capabilities. Microsoft Office is the most natural place for business users to format, sort, and analyze data disconnected from a network. Once back online, MicroStrategy Office refreshes the files with the latest data with just one mouse click while preserving the analytical and formatting work.

3.10 MICROSTRATEGY DESKTOP

MicroStrategy Desktop™ is an advanced development and analysis interface for creating BI applications intended primarily for analysts, power users, and application developers. Desktop builds the metadata

objects used for designing reports, scorecards, and dashboards. It employs editors and wizards to speed-up application development without the need for coding. Together with MicroStrategy Architect, they are the key development tools required for create BI applications.

Desktop is an equally powerful BI interface for the most advanced analysts who aggressively investigate the data to uncover valuable insight. Desktop provides a full range of analytical functionality for reporting, data mining and predictive analysis, statistical analysis, financial analysis, mathematical analysis, set analysis, and time series analysis. Desktop offers the rich Windows interface and power of a thick-client, workstation environment.

3.11 MICROSTRATEGY ARCHITECT

MicroStrategy Architect™ is a rapid development tool that creates the metadata objects that map the physical structure of a database to a logical, object-oriented model of the business. Architect employs a graphical interface and editors to link the enterprise's business model to the physical database tables and columns.

The abstraction of the physical database into a logical business model makes subsequent report design using Desktop extremely fast and intuitive. The logical business model abstraction also provides a layer of isolation between the physical database and the reporting applications allowing each to change independently of the other as they evolve over time. Moreover, the object-oriented nature of the metadata repository allows all changes in the logical model to propagate immediately and transparently to all dependent objects.

The abstraction of physical models to logical models allows MicroStrategy-based analytical applications to be ported across different data warehouse platforms. This has profound benefits to analytic application developers who want their applications to be easily connected to any existing data warehouse.

MicroStrategy Architect also includes five starter BI applications or analytic modules designed to dramatically speed up the development and deployment of customized BI applications. The modules are a result of MicroStrategy's 20+ years of BI experience, and represent horizontal applications common to most every industry:

- Financial Reporting Analysis
- Human Resources Analysis
- Sales Force Analysis
- Customer Analysis
- Sales and Distribution Analysis

Each module includes a data model, an extensive library of foundation metrics, attributes, and hierarchies, and many business reports. Each module can quickly be linked to existing data warehouses to rapidly produce a working BI application. The application analytics can be easily adapted and extended to the unique needs of each enterprise. This application portability is unprecedented in the BI industry, and is only available for applications built on the MicroStrategy platform.

3.12 MICROSTRATEGY MULTISOURCE OPTION

MicroStrategy MultiSource Option™ is a plug-and-play add-on to Intelligence Server that allows users to seamlessly report, analyze, and monitor corporate data across multiple data sources through a single multi-dimensional view of the business. It extends the unparalleled scalability, analytical depth, and query performance of MicroStrategy's award-winning ROLAP technology to heterogeneous data source environments. Users and report developers can look at all relevant information as if they were looking at a single database.

MicroStrategy MultiSource Option employs a Multi-source Relational (ROLAP) architecture that pushes

calculations and all data joins down to the database utilizing the power of the database management system. This push-down design eliminates the need for vast hardware resources on the BI middle tier servers. Business users can seamlessly report, analyze, and monitor data across multiple sources. Companies can get BI applications up and running in almost no time with minimum data engineering.

3.13 MICROSTRATEGY CLUSTERING OPTION

MicroStrategy Clustering Option™ is a plug-and-play add-on to Intelligence Server that allows a group of Intelligence Servers running on separate machines, called nodes, to work together as a single logical system. This provides high availability and high performance for heavily-loaded, mission-critical business intelligence systems. Different BI applications can be loaded onto different Intelligence Server nodes enabling more control of the available hardware resources.

The Intelligence Server nodes communicate with each other in a peer-to-peer relationship so that each node maintains a backup of the other nodes, and information about the jobs, report caches and cubes created and maintained by the other nodes. If any node fails, the other active nodes can take over the work of the failed node.

The workload from MicroStrategy Web, Mobile, Office, and Distribution Services is distributed across the Intelligence Server nodes in a cluster. New user sessions are allocated to the node with the lightest workload based on the current session load on each node. Balancing factors can be applied to different nodes depending on the hardware resources of each node.

3.14 MICROSTRATEGY OBJECT MANAGER

MicroStrategy Object Manager™ is a change management tool that manages the application development lifecycle by assessing the impact of changes to the application, and migrating these changes across development, testing, and production environments. In short, MicroStrategy Object Manager makes change management accurate, fast and efficient. MicroStrategy Object Manager includes the following components:

- Graphical Interface – enables administrators to interactively and visually move objects between applications using a drag-and-drop or copy/paste from the source application into the destination application.
- Update Packages – extract changes into a standalone file (.mmp extension) outside of the MicroStrategy metadata. This file contains all the information about the objects allowing migration without a live connection to the source application.
- Project Merge – merges two separate BI applications by copying all objects from the source. Project Merge is useful when synchronizing or consolidating two applications.

3.15 MICROSTRATEGY INTEGRITY MANAGER

MicroStrategy Integrity Manager™ automates the detection of inconsistencies and errors so that business users can rely 100% on the accuracy of their information. It compares each dashboard or report, comparing its data, SQL, graph, Excel, PDF output and execution times. This saves the time needed to manually generate and compare report output. This also eliminates potential human errors when manually comparing hundreds of thousands of cells of data, or hundreds of lines of detailed SQL, or pages of pixel-perfect formatted enterprise documents, between two reports, documents, or dashboards.

Integrity Manager can detect the impact of any change to the BI ecosystem. The results of the reports can be stored as historical snapshots and used as a baseline for subsequent report comparisons. Integrity Manager is designed to handle thousands of reports thus adding value exponentially. Data and report integrity testing that previously took weeks if not months of IT effort can now be accomplished overnight.

3.16 MICROSTRATEGY ENTERPRISE MANAGER

MicroStrategy Enterprise Manager™ enables analysis of resource utilization, project performance, user statistics, and trends to facilitate performance tuning and resource planning of a MicroStrategy business intelligence implementation. Enterprise Manager is a packaged data warehouse containing historical analysis of MicroStrategy operating statistics, a data-load process that transforms the operating statistics to the data warehouse, and a MicroStrategy project built on top of the data warehouse.

With Enterprise Manager, administrators are more informed and can easily maintain application speed by eliminating unused reports, reduce downtime by identifying capacity or performance issues before they occur, and analyze historic usage patterns to plan for future application growth. Enterprise Manager enables report developers to better understand user adoption trends, identify patterns, and review project success. Project managers are equipped to plan for resources, analyze trends in project growth, and quantify and improve project ROI. Areas of analysis are grouped into four primary reporting areas:

- User Analysis
- Project Analysis
- Operational Analysis
- Performance Analysis

3.17 MICROSTRATEGY COMMAND MANAGER

MicroStrategy Command Manager™ enables script-based administration and maintenance of objects, security, and system configuration for large user communities using textual commands. These commands can be compiled into script files and run from a graphical interface, through a command line editor, or directly from the command line. The script files can be scheduled through operating system utilities or integrated into 3rd-party Systems Management Software, automating repetitive maintenance tasks.

MicroStrategy Command Manager is designed to minimize operational costs by automating BI administration. Scripted, tested, and automated task execution minimizes administrative effort. Tasks commonly used together can be combined via Java programming to create procedures. Procedures save time by automating processes that require multiple complicated steps for completion.

3.18 MICROSTRATEGY SDK

The MicroStrategy SDK™ is a comprehensive development environment primarily used for integrating MicroStrategy functionality into other existing systems, especially enterprise portals, and customizing and extending the functionality of the MicroStrategy Web. The MicroStrategy SDK includes:

- Documentation of the API and platform functionality
- Development kits for portal, external security, and Web services integration
- Utilities and sample code that show how the APIs can be used

The SDK is a key architectural component used to fully exploit the power of the MicroStrategy platform through its documentation of the platform's open APIs. The APIs expose the complete set of functionality contained in the platform. The platform includes the following groups of API's:

- MicroStrategy Web API
- MicroStrategy Mobile API

- Visualization API
- Portal integration API
- Web services API
- MicroStrategy Office API

3.19 SUMMARY

The MicroStrategy platform architecture offers the only true, unified, organically-developed enterprise class business intelligence technology required for today and tomorrow's most demanding BI applications. Based on one, object-orientated metadata, the MicroStrategy platform consists of 17 distinct products designed to deliver the complete range of BI functionality to business users, the depth of analytical power to analysts and developers, comprehensive administration to IT administrators, and limitless extensibility to IT developers and integrators. Only the MicroStrategy platform architecture is worthy of the moniker 'Enterprise-wide BI Standard' now and well into the future.



PLATFORM COMPONENTS

4

METADATA AND THE LOGICAL MODEL

Metadata is information that describes another set of data. In a BI platform, metadata links information in a data store to business entities and rules that define a BI application. At the lowest level, metadata translates qualitative and quantitative information from a data source into meaningful business terms. It unlocks valuable information stored within data sources by serving as an intuitive map to guide business users in building reports and conducting analyses. Today's BI platforms need centralized, reusable, and dynamic metadata to support enterprise-wide BI standardization. MicroStrategy provides the only metadata in the industry that is ready for BI standardization by incorporating the following architectural design tenets:

- Abstraction in a single, centralized location.
MicroStrategy's metadata is stored in a central repository, where it stores mappings of business concepts to underlying data structures as well as layouts and report configurations. As a result, all users, regardless of the interface through which they access the application, receive a single version of the truth.
- Fully reusable well-defined metadata objects.
MicroStrategy's metadata repository is comprised of multiple layers of well-defined objects, with each layer serving as the foundation upon which to build the next. This approach results in the broadest range of reusable objects, increasing development speed and minimizing maintenance costs.
- Easily portable from one BI application to another.
MicroStrategy metadata objects can be easily ported from one BI application to an unrelated application enabling sharing common business metadata definitions across the organization. So, when a new BI application is created, a BI application developer need not re-create the common metadata objects but just import the standardized definitions. This increases BI application development efficiency and eliminates duplication of effort.
- Dynamic, object-oriented metadata objects.
Every MicroStrategy metadata object contains "pointers" to other metadata objects. Whenever an object is used, it is dynamically assembled from the latest versions of all the other objects contained in the object. Changes to one object automatically cascade to related objects.
- Storage in a standard relational database.
In contrast to other vendors' proprietary metadata storage formats, MicroStrategy's metadata is stored in a relational database. This approach increases the scalability and manageability of BI applications.

4.1 A COMPLETE METADATA IS KEY FOR EFFICIENT BI PROCESSING

The MicroStrategy BI architecture incorporates seven primary layers of metadata objects. Each layer represents a progressively higher level of abstraction from the physical data structures:

1. Configuration Objects

Users, user groups, schedules, projects, and data source connections make up foundation of BI applications.

2. Data Abstraction Objects

Logical tables, attributes, and facts translate physical data structures to the logical business model.

3. Business Abstraction Objects

Metrics, hierarchies, custom groups, and transformations combine data abstraction objects into business definitions and organizational structures.

4. Report Components

Templates, filters, autostyles, and prompts are high level abstraction objects, and are part of the basic building blocks of a report definition.

5. Reports and Documents

Report definitions are highly formatted enterprise reports that include cross-tabular grids and graphs as well as scorecards, dashboards, visual analyses and mobile apps. These reports track key performance indicators and business processes.

6. Visual Analyses

A Visual Analysis object is a display of raw data as a visualization. With a Visual Analysis object, business users can quickly spot outliers, identify patterns and discover data visually eliminating the need to sift through large columns and rows of data for insights.

7. Delivery Objects

Delivery objects define personalized information services that distribute reports to E-mail recipients, and to printers and folders.

The abstractions defined by these metadata object layers enable quick and easy report creation, and distribution without the need for any programming. Objects are simply assembled directly from the metadata and resolved when the report is run. Since all MicroStrategy objects access the same metadata repository, any modifications to the metadata are automatically propagated to all objects and all reports.

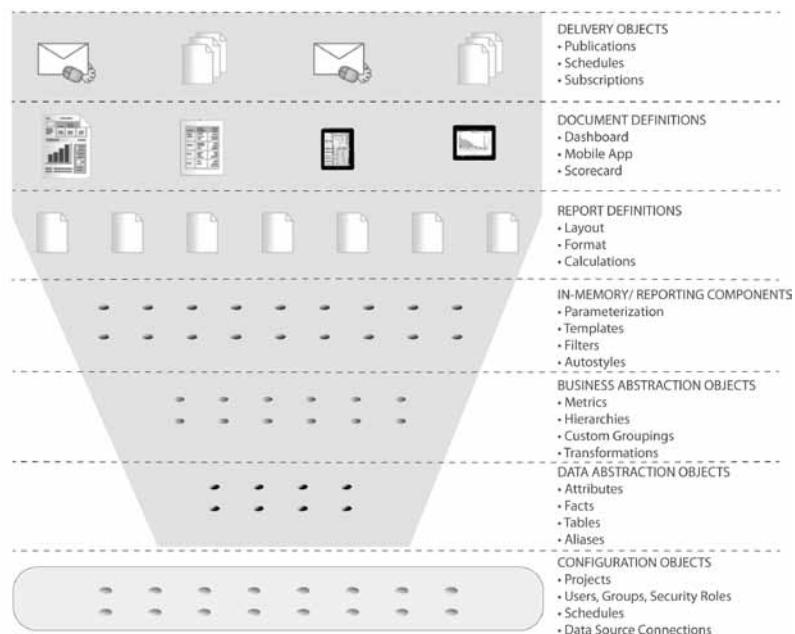


Figure 4-1 MicroStrategy's metadata speeds BI application development, and eases maintenance burdens with its object-oriented, reusable, and fully dynamic metadata.

4.2 CONFIGURATION OBJECTS

Configuration objects serve as the framework for building BI applications, and are shared across projects. They define the scope of BI applications built on the MicroStrategy BI platform by encapsulating the accessible data sources, and maintaining users and their groups. Business definitions relevant to the entire organization, such as Security Roles and Schedules, are persisted at this level.

Building BI Applications Using Projects

A project is a group of related business definitions serving a common purpose within an organization. Together these business definitions serve as the building blocks for a BI application. By grouping them under a project, properties can be defined on top of all these related business definitions for administration and control. Each project can have its independent:

- User to Security Filters mapping
- User to User Authorization mapping
- Combination of Data Sources
- Usage Governors

Creating Users and Their Groupings

Each user within an organization can access a different set of BI applications. By placing user objects at the Configuration Object level, they are shared across projects. Within the MicroStrategy BI platform, users are grouped into different user groups for efficient user management. Each user group can contain multiple users, and each user can belong to multiple groups. User objects contain the following user information:

- Login
- Full Name
- Description
- Password
- Password Properties
 - User cannot change password
 - User must change password at first login
 - Expiration Date
 - Number of failed attempts before lock out
 - Password complexity
- Project Access
 - Security Role used
 - Privileges Granted
- Security Filter per Data Source
- User Groups membership
- Authentication Parameters for NTLM, Database, and LDAP Authentication

Representing Data Sources and Their Access Pathways

While each BI application within the MicroStrategy BI platform can access multiple data sources, information on their characteristics and connectivity is stored once and reused. Database objects encapsulate details on the nature of the relational, cube, or other data source along with the data source-specific MicroStrategy

optimizations. Database Connection objects contain details on how connectivity to various data sources is achieved and maintained while Database Logins hold information on how the Intelligence Server should authenticate to these data sources. Together these objects define the scope of data accessible by all business definitions within a MicroStrategy metadata repository.

Within each project, connection mappings allow different users to access each data source using different Database Login objects. When users share the same Database Connection and Database Login objects, the same connection to the data source is cached and reused for these users' queries.

Security Roles and Schedules

Each user within an organization has a different role, and should thus have visibility to different aspects of its operations. For example, a Human Resources manager has access to all employee hiring records but may be restricted from projected revenue data. Mirroring business operations in an organization, this same HR manager can have different privileges within the BI application for Human Resources and the BI application for Sales Forecasting. For ease of maintenance, these privileges are bundled under Security Role objects within the MicroStrategy BI platform.

Time and event-based schedules allow reports executions, deliveries, and administrative tasks to occur at any time of the day or week. Actions across different projects can share schedules for simultaneous processing.

4.3 DATA ABSTRACTION OBJECTS

Data abstraction objects map the physical data model of the data source into a logical business model using business terms and rules. Data abstraction objects serve as the building blocks for business abstraction objects and, ultimately, for all reporting, analysis, and monitoring.

Physical and Logical Models of Relational Data Warehouses

The MicroStrategy architecture supports a wide range of physical data warehouse schemas, including star and snowflake schemas. These warehouse schemas can contain sparse aggregate tables, partition tables, and split fact tables. MicroStrategy also provides direct access to operational data stored in third-normal form. When working with a traditional data warehouse, MicroStrategy Architect retrieves table, view, column, and data-type information directly from the database system catalogs. The logical business model, including the relationships between the business entities, is created from this information. The physical data structure and logical business model are presented in two graphical views within MicroStrategy Architect:

1. The physical table structure displays the data warehouse tables, columns, and data types.
2. The logical table structure displays the table and column information in the form of attributes and facts. The relationships between the attributes are displayed in a multi-dimensional view.

The screenshot shows two side-by-side windows for the 'Warehouse Table: ORDER_FACT'.

Logical View:

Name	Type
Employee	Attribute
Day	Attribute
Days to Ship	Attribute
Customer	Attribute
Order	Attribute
Payment Method	Attribute
Ship Date	Attribute
Shipper	Attribute
Phone Usage	Attribute
Freight	Fact
Revenue	Fact
Units Sold	Fact
Order-level Units Sold	Fact
Profit	Fact
Cost	Fact
Gross Revenue	Fact
Customer Id	Fact

Physical View:

Name	Data type
ORDER_ID	Integer (4)
EMP_ID	Integer (2)
ORDER_DATE	TimeStamp (0)
ORDER_AMT	Double
FREIGHT	Double
SHIP_DATE	TimeStamp (0)
QTY SOLD	Double
RUSH_ORDER	NVarChar (50)
ORDER_COST	Double
CUSTOMER_ID	Integer (4)
PVMT_TYPE	Integer (2)
SHIPPER_ID	Integer (2)
GROSS_DOLLAR_SALES	Double

Checkboxes at the bottom left: 'The key specified is the true key for the warehouse table' (checked), 'Logical size: 63', and 'Preserve this logical size when updating Schema information (Lock Logical Table Size)'.

Figure 4-2 The metadata stores a physical and a logical view of a database table.

Logical tables can also be defined from hand-coded freeform SQL statements that combine data from different physical tables and views in the data warehouse. This approach provides many of the benefits of database views while working exclusively within MicroStrategy's object-oriented metadata.

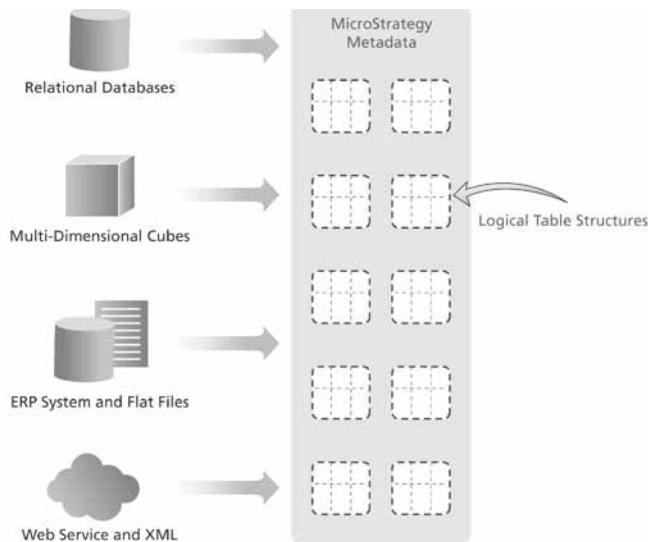


Figure 4-3 Information from all data sources is uniformly mapped in the MicroStrategy metadata.

Facts are the Basic Building Blocks of Business Metrics

A fact is a table column, or an expression combining multiple columns, that contains quantitative raw or aggregated values such as amounts, counts, quantities, forecasts, and other measurable business calculations. A fact stored as a data abstraction object in the MicroStrategy metadata is represented as:

- A single column, e.g. DOLLAR_SALES
- A single column with an aggregation function, e.g. SUM (DOLLAR_SALES)
- Several columns combined using functions and expressions, e.g. SUM(PROFIT / DOLLAR_SALES)

Attributes Give Context to Facts

Attributes are defined by a column or columns in one or more tables, and provide meaningful qualitative context to facts. Attributes are commonly displayed on reports. They also serve to set the aggregation level for calculations in a report, and for filtering the data that is retrieved from the data warehouse. Examples of attributes are regions, stores, customers, and months.

Attributes can have many descriptors or forms as they are known in MicroStrategy. An attribute form is one of several columns that are different representations of the same attribute. For example, the attribute Product might have the following forms — an ID, short name, long description, SKU, and image.

Attribute elements are the actual data values of an attribute stored in the data source. They are retrieved and cached by MicroStrategy when running reports and analyses. For example, New York, Paris, and Tokyo are elements of a City attribute.

Abstractions for Relational, Multi-Dimensional, and Operational Data

For other data sources, MicroStrategy's BI architecture translates the physical data structure – whether it is a database table, a multi-dimensional cube, a collection of ERP tables joined together, or an Excel spreadsheet – into a logical table structure. These logical table structures form the basis for defining attributes, facts, and higher level metadata objects.

MicroStrategy accesses cube-based data sources, and automatically converts the multi-dimensional models into logical MicroStrategy metadata objects. For example, SAP® InfoCube, QueryCube, and ODS objects can be used as MicroStrategy metadata objects. SAP levels, attributes, and structures are imported as MicroStrategy attributes, and key figures are imported as metrics. MicroStrategy also joins data across SAP BW InfoCubes, QueryCubes, and other MicroStrategy data sources, and can access multiple instances of SAP BW at once.

Attributes and metrics from all data sources can be mapped to each other. As a result, data from all these sources can be joined, and transparently presented in a single document to business users. Calculations involving metrics from different sources are also possible.

Freeform queries in the form of SQL or XQuery map operational data to MicroStrategy metadata objects. For any data source, custom queries can be provided to extract information, and mapped to existing or new attributes within the metadata. Any data source accessible via ODBC, and SOAP and RESTful Web services can be integrated into the MicroStrategy metadata.

Flexible Schema Combinations Make Complex Business Modeling Simple¹

Complex business models that are not stored as standard data structures are simple to define in MicroStrategy. In addition to basic column mapping, as described above, facts, attributes, and attribute forms can use any function supported by the database platform to combine and extend data in the data warehouse. This capability eliminates the need to change the physical data structure in order to support an organization's reporting and analysis requirements.

For example, only the customer's date of birth is stored in the database. There is no need to create a new column in the data warehouse for customer age, and periodically update this column. Instead, the attribute definition for the customer age is an expression that dynamically calculates the difference between the system date and the customer's date of birth.

¹See Appendix A for more information on schema support

4.4 BUSINESS ABSTRACTION OBJECTS

Business abstraction objects define the business terms and rules. They serve as the building blocks for other business abstraction objects and for all reports, analyses, and documents.

Metrics are Business Measures

Metrics represent business measures and key performance indicators. Metrics consist of facts, attributes, or other metrics combined with a formulaic calculation. This formula may include aggregation functions such as "Avg", "Sum" or "Count," arithmetic operators (+, -, *, /) or other advanced functions (e.g. data mining, financial, mathematical, OLAP, statistical). A metric can contain 3 additional properties:

- Dimensionality

Metric Dimensionality determines the level of aggregation of a metric. The default aggregation level is based on the attributes contained in a given report. The aggregation level of a metric can be changed to force aggregation at a specified level. This flexibility is especially useful if business metrics need to be calculated at the same level regardless of the level of reports that use them.

- Filter

Metric Filters determine any restrictions on the data used in the calculation. A metric filter may be defined as an embedded component of the metric or as a link to existing filters, maximizing object reuse, and reducing administration.

- Transformation

Transformations may be associated with a metric to enable time series analysis as described in the next section.

Automatic Metric Creation

MicroStrategy has simplified the process of metric creation. Commonly used basic metrics or business measures such as averages, counts, maximum and minimums, are automatically created during the modeling process. This speeds up the metadata creation enabling users to spend time developing complex analytical metrics using the standard metrics.

Predictive Analysis Scoring Metrics

MicroStrategy's BI platform incorporates sophisticated data mining algorithms that define powerful predictive, segmentation, association, and time series metrics. Data mining is the process of using statistical analysis, modeling techniques, machine learning, and other technologies to identify hidden patterns and predictive information in a data warehouse.

Predictive models from third-party data mining tools can be imported using a Predictive Model Markup Language (PMML) representation of the model. A scoring predictive metric and confidence predictive metric is generated. These metrics are stored in the MicroStrategy metadata, and used like any other metric making them easy to deploy to all your users.

Transformations Add Flexibility to Facts

Transformations are reusable business abstraction objects that are embedded primarily in metrics for time comparison calculations. They automatically display information for changing dates and time periods without constantly needing to redefine metadata objects. For example, many reports display current period

information along with prior-period and period-to-date information. In this example, transformations handle the variable time periods automatically regardless of when the reports are executed within the same object definitions. Two types of transformations are supported in MicroStrategy:

- One-to-one transformations compare one period to another, for example, “Last Month” and “This Month Last Year”
- Many-to-many transformations define period-to-date and last-in-period metrics, for example “Month-to-Date” or “Last Four Weeks”

Hierarchies Align Attributes into Business Dimensions

A hierarchy is a logical group of attributes that defines a relationship between them. A hierarchy describes an ordered path through the attributes from a summarized level to more detailed levels. Hierarchies are used to browse data, and provide a navigation path when drilling from one attribute to another on a report.

In MicroStrategy, hierarchies also serve as placeholders for an attribute in a report, with the specific attribute being determined at run-time. There are two types of hierarchies:

- System Hierarchy

The system hierarchy is a direct map of the logical table structure stored in the metadata, and shows the true relationships between all attributes in the reporting application. There is only one system hierarchy for each MicroStrategy application.

- User Hierarchies

MicroStrategy allows a greater range of hierarchical paths than those found in the physical database. User hierarchies define relationships between attributes based on the business rules of the reporting application, and they guide user analysis or provide shortcuts to common analytic workflows. Attributes can be reused in different user hierarchies.

Drill paths can be defined for individual reports, over-writing the system and user hierarchies, and enabling specific workflows through the data.

Custom Groups Merge Data Elements into Virtual Attributes

Custom groups and derived elements define arbitrary groupings of attribute elements in a report, and represent data groupings that do not exist in the data warehouse itself. They are defined either by using filters or by specifying metric bands to group certain attribute elements together. User-defined descriptions identify each row of custom group elements.

Custom groups are commonly used for financial statements. For example, each row in a profit and loss report aggregates groups of ledger accounts and sub-accounts into a single value. Custom groups define these account groups for the various sections of the report: revenues, costs, and expenses.

Consolidations Extend the Definition of Custom Groups

Consolidations serve a function similar to that of custom groups in that they act as user-defined virtual attributes on reports, but their definition is restricted to filters on actual attribute elements that make up the group. With consolidations, additional groups can be created by performing arithmetic operations between any of the other groups in the consolidation. Derived elements can also be defined by performing arithmetic operations between groups of data.

4.5 REPORT COMPONENTS

Report components are dynamic, reusable metadata objects built with the data abstraction objects, business abstraction objects, and other report components. These three layers of metadata objects are combined to create new reports and analyses in both MicroStrategy Desktop and MicroStrategy Web. In all cases, the metadata objects retain their reusable and dynamic characteristics, and a change to a single object in the metadata is automatically reflected in all other dependent metadata objects.

Filters Focus the Data of a Report

A filter specifies the conditions that data must meet to be included in a report or metric. Filters limit the data that is extracted from the data source, and focus the data presented to the end user. A filter is essentially a logical expression that combines any metadata object, mathematical and comparison operators, and constant values and lists.

Filter expressions can be combined using Boolean operators and nested to create sophisticated conditional clauses. For example, to retrieve data for New York and San Francisco for the fourth quarter of 2012, the filter would be:

(City='New York' OR City='San Francisco') and Quarter='2012Q4'.

Prompts Give Users Control Over Report Content

Run-time prompts provide report input parameters that control most aspects of a report, and give the user significant ad hoc reporting capabilities. Prompts enable the personalization of reports from a single report definition, reducing the number of objects stored in the metadata repository. There are four types of prompts in MicroStrategy:

- Filter prompts provide filtering criteria at runtime, and qualify the data set from the data warehouse that will be included in the report results.
- Object prompts let users choose which attributes and metrics to include on a report at run-time. They can also be used to build metadata objects, particularly metrics, providing enormous flexibility from a single report definition.
- Value prompts allow the input of alphanumeric and date values that specify constants for filtering criteria and metric calculations.
- Level prompts dynamically control the aggregation level (dimensionality) of metrics.

SAP Variables are Imported as Prompts

SAP BW uses variables to specify filtering parameters at run-time. As part of MicroStrategy's certification with SAP BW, the MicroStrategy BI platform automatically translates SAP BW variables into reusable MicroStrategy prompts. Each MicroStrategy prompt contains a dynamic pointer to the SAP BW variable; changes to the definition of an SAP BW variable automatically ripple through to MicroStrategy metadata objects.

Templates Support Any Layout for a Report

A template specifies the layout and formatting of the data on a report. Attributes, metrics, custom groups, consolidations, and prompts can be placed in rows, columns, or pages on the template grid. There is no theoretical limit to the number of objects that can be placed on a template. Besides being able to define the presentation characteristics, such as font, color, alignment, and number formats, additional report characteristics such as subtotaling, metric thresholds, and graph properties are also stored in templates.

Autostyles

An autostyle defines a reusable report format. The MicroStrategy BI platform contains more than 35 autostyles and new autostyles can easily be added to match corporate standards, business reporting requirements or personal formatting preferences. New autostyles are immediately available to all MicroStrategy users. Default autostyles can be specified for projects, reports, and individual users.

4.6 REPORT AND DOCUMENT DEFINITIONS

MicroStrategy enables Business users to view and analyze business information in two different formats, Reports and Documents.

Reports

A MicroStrategy report is a specific business user question abstracted as a set of attributes, filtering conditions, metrics, and sub-totals, and sent to the data-warehouse for processing. The MicroStrategy Query Engine converts a business user question into a SQL or MDX query. The data retrieved in response to the business question is specific and formatted and may be obtained from one or many data-sources. MicroStrategy reports gather, consolidate, search, analyze, and convey information. MicroStrategy reports take on any form and role, including predictive analysis, cross tabular grids and graphs, operational reports, and transactional reports. A report definition includes all information necessary to calculate, place, and format the information. A MicroStrategy report is used for creating datamarts, as datasets for documents or as filters in a different report definition.

MicroStrategy reports consist of two components: a template and a filter. A template, which consists of attributes and metrics, is an abstraction of the fields requested by the user where as a filter is a condition imposed. MicroStrategy reports are fully interactive, allowing users to toggle between grid and graph mode, rearrange report layouts by pivoting rows, columns and pages, change the formatting, sort the columns, add subtotals, and create new calculations in the report.

Each report can access any of the other metadata objects at all levels. The reusable, dynamic metadata lets business users perform investigation and analysis on data that was not included on the original report. For instance, a weekly product sales report can be transformed into a list of customers most likely to respond to a direct marketing offer for the top 5 selling products by drilling to any attribute in the data warehouse and applying filtering conditions. These navigation paths do not need to be defined in the metadata at all. All calculations and relationships are handled automatically by MicroStrategy based on the object-oriented metadata objects.

At any time, a new report can be saved as a new metadata object. This new object stores dynamic links to all the data and business abstraction objects, report components, and formatting and layout settings.

At any time, a new report can be saved as a new metadata object. This new object stores dynamic links to all the data and business abstraction objects, report components, and formatting and layout settings.

Business User Question:
What is regional performance
for various brands in the
Electronic category?



Customer Region	Brand	Metrics		Revenue Rank by Region
		Profit	Revenue	
Northeast	Sony	\$69,110	\$377,111	1
	Sharp	\$58,026	\$322,835	2
	Hewlett Packard	\$38,000	\$318,978	3
	Panasonic	\$31,662	\$169,337	4
	Harman Kardon	\$20,854	\$123,470	5
Mid-Atlantic	Sony	\$77,619	\$432,360	1
	Sharp	\$73,347	\$410,180	2
	Hewlett Packard	\$50,275	\$279,093	3
	Panasonic	\$45,376	\$262,642	4
	Harman Kardon	\$26,376	\$132,695	5
Southeast	Sharp	\$21,603	\$119,714	1
	Sony	\$21,298	\$115,043	2
	Hewlett Packard	\$12,183	\$70,919	3
	Panasonic	\$11,568	\$68,672	4
	Harman Kardon	\$8,814	\$52,026	5
Central	Sony	\$89,401	\$449,141	1
	Sharp	\$82,096	\$446,987	2
	Hewlett Packard	\$51,510	\$281,070	3
	Panasonic	\$46,708	\$274,394	4
	Harman Kardon	\$30,183	\$164,350	5
South	Sony	\$34,403	\$186,794	1
	Sharp	\$29,925	\$169,337	2
	Panasonic	\$17,859	\$110,196	3
	Hewlett Packard	\$19,152	\$107,438	4
	RCA	\$8,924	\$50,161	5

Figure 4-4 A business user question is abstracted into a report consisting of attributes and metrics placed in rows, columns, and pages.

Documents

Documents are a compilation of reports or datasets, with a highly formatted display. Documents are built with attributes and metrics from different reports or datasets in a single interface. Multiple reports datasets combined with multiple formats in a single source serve to provide multifaceted and comprehensive insight into the business. A high degree of formatting and layout control further increases the flexibility of the MicroStrategy BI platform in meeting any reporting needs. While an individual document object is generic, it is a basic building block for several types of formats used by any organization to disseminate business intelligence to its business users. MicroStrategy classifies documents into 3 categories:

1. Mobile Apps

It is now universally accepted that Mobile will become the preferred device for users to receive information, look at business reports and conduct transactions. MicroStrategy provides organizations with a mobile app platform that enable them design 100s of new Mobile apps. Each mobile app uses a document object at its core, and is built on top of the one of the number of Mobile specific document templates. The document object provides developers the flexibility of creating three different types of mobile apps:

- Business intelligence App – An app that uses a document object to deliver critical business intelligence reports to business users. This app provides typical BI analytical functionality such as pivoting, drilling, page-by on a Mobile device.
- Transaction Apps – An app for conducting write-back transactions from a Mobile device. Such an app is based on a transaction-enabled document object. A transaction-enabled document contains grids or fields used to input data from the Web browser or Mobile device. Users type in data from a large number of data inputs such as text boxes, sliders, switches, steppers, toggles, calendar, camera, and GPS positioning.
- Information Apps – These apps are not related to business intelligence at all but enable organizations to deploy information sources such as brochures, white papers, videos and others through a MicroStrategy document object to its user base.

1. Analysis

- Mobile BI
- Exploration and Manipulation
- Grids and Graphs
- Data Visualizations



2. Transactions

- Write to Databases
- Write to operational systems
- Submit new data
- Update existing information



3. Information

- PDF, Video, ePub, web
- Graphic Design quality output
- Portals, Libraries, Boardbooks



Figure 4-5 MicroStrategy platform is designed to enable organizations to build Business Intelligence, Transaction or Information apps – each type built using a core document object.

2. Interactive Dashboards

Dashboards combine large numbers of reports into a document into a layout provides interactive, visual way to consume information about business performance. Interactive dashboards enable managers to visually identify anomalies, test ‘what-if’ scenarios, and get a view of the “at a glance” information of the business KPIs. As opposed to enterprise reports below, dashboards provide a high level view of the business, with interactivity to drill down to lower levels.

Interactive dashboards combine large number of reports, which act as datasets, into a single dashboard container that provide business users an “at a glance” view of the business. Designing a dashboard is very

easy; users simply define panels in the design interface, add grids and graphs to the panels, and choose a selector to control the display of the panels. To make a grid or graph interactive, users choose a relevant visualization, such as a time-series or heat-map visualization, which renders the data in a visually appealing format. The visualizations have built-in interactivity enabling business users to quickly assess business performance and identify abnormalities. Interactive dashboards use Flash and can be exported into a PDF or MIME HTML (MHT) format for offline analysis.

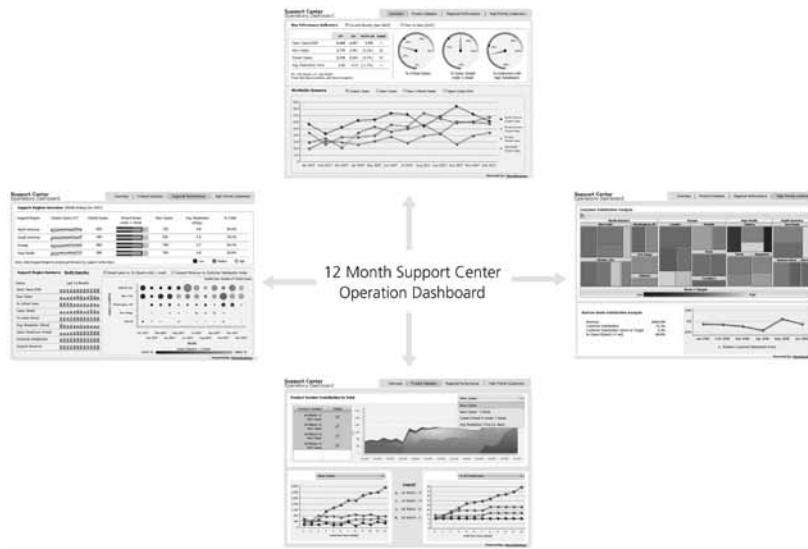


Figure 4-6 A sample dashboard showcasing the importance of different visualizations such as bullet graphs, microcharts, and heatmap.

3. Enterprise Reports

Enterprise reports are desktop-publishing-quality reports built on data from the MicroStrategy BI platform using an intuitive design interface. These reports are very detailed and typically have hundreds of pages, with different levels of subtotaling. Users create reports with pixel-perfect precision by simply dragging-and-dropping, and then formatting reports for boardroom-quality presentation. Enterprise reports can be delivered in Portable Document Format (PDF), ensuring printed reports have the same quality as on-screen reports. After placement and adjustments of document elements, formatting is defined in segments of bands and zones within bands, giving maximum flexibility in creating sophisticated presentations of data.

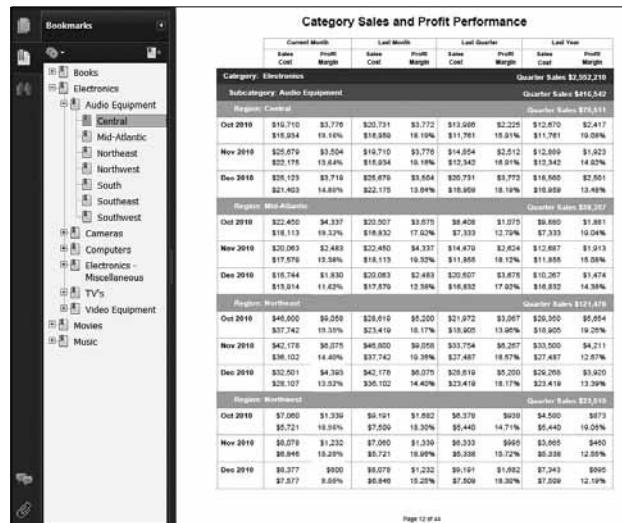


Figure 4-7 A 44-page detailed report documenting category and sales performance for various product categories in a pixel perfect format.

4.7 VISUAL ANALYSES

A Visual Analysis is a presentation of large data volumes as intuitive data visualizations². This format of information delivery is specifically designed for data discovery and is aimed to empower business users to get their business insights, spot outliers quickly without involving people to IT. Instead of looking through rows and columns of data, a user can see their data in a visual format and quickly derive insights by adding attributes and metrics onto the visualizations. MicroStrategy provides a large number of advanced visualizations to choose from such a Graph Matrix, Maps, Heat Maps, and standard charts. A visual analysis is fully interactive and provides standard analysis functionality such as multi-key sorting, pivoting, drilling, adding new metrics and aggregations, ranking, page-by, thresholding as well as filtering.

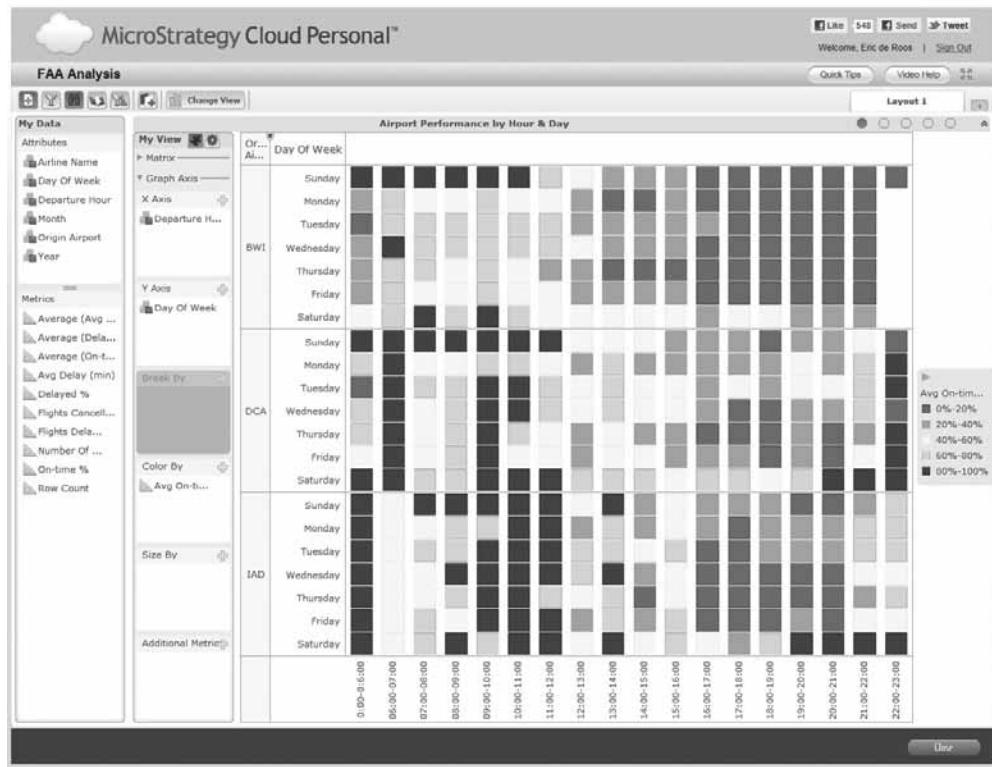


Figure 4-8 A highly interactive visual analyses layout with multiple visualizations, drop zones and filtering pane.

4.8 DELIVERY OBJECTS

MicroStrategy provides high-volume, automated distribution of reports, documents, dashboards and business performance alerts to E-mail, file servers, and network printers. Both administrators and users can define subscriptions to reports, dashboards, or documents to a variety of devices for a large audience in any desired format. All objects required to create a subscription are included in the MicroStrategy metadata repository making configuration and administration possible from a single interface. An administrator can manage delivery objects and subscriptions, monitor report deliveries, manage delivery formats and channels, and manage users and contacts. The object-oriented metadata layer simplifies delivery management by easily identifying subscriptions that are dependent on a particular report, document, or dashboard as well ensuring that subscribed reports, documents, and dashboards automatically inherit the security profiles set of users. All delivery objects transparently take advantage of Intelligence Server features, such as report and document caching, load balancing, system fail-over, and Intelligent Cubes.

²See Appendix E for a list of graphs and visualizations

Report Delivery Services are Automated and Highly Customizable

A report delivery service contains all the information required to deliver personalized and timely information to all business users. A service consists of four delivery objects:

- Publications – store the message content
- Schedules – frequency or exception condition for generating the content
- Subscriptions – users who receive the content
- Transmission Device – delivery method for the content

Delivery Channels

The MicroStrategy BI platform provides several channels to receive the delivery of subscribed reports, dashboards, and documents:

- E-mail – Subscribed documents can be sent to an E-mail addresses
- Print – Subscribed documents and reports can be sent directly to a printer
- File – Saved as a file in a location anywhere on the network
- History List – as a unread message in the history list provided in MicroStrategy Web

Delivery Formats

Subscribed reports and documents are delivered in the following formats. Additionally, subscribed objects can be distributed in a compressed format (to reduce size) or password protected (for security).

- HTML
- Excel
- PDF
- Flash embedded in PDF or MHT files
- Plain Text

Transmitters and Devices

Transmitters and devices together specify the communication medium and format used to deliver the requested publication, at the appropriate time and with any relevant personalization, to all the correct users. A transmitter is a medium (E-mail, file, Blackberry, iPhone, iPad and print) where as a device is the medium type (Microsoft Outlook E-mail, Outlook Express, Apple Push Notifications). A transmitter and device together define the Delivery Channel.

Time- or Event-Based Schedules Trigger Deliveries

A schedule sets the times or frequencies that a delivery service is executed, and represents a recurring pattern—not a fixed date—on which a subscription is executed. Schedules may be time or event based. Different schedules may be associated with a single subscription, and may be reused across any number of subscriptions. A time-based subscription will execute at a specific time, with a particular frequency (for example, every Monday Morning at 8:00 AM) while an event-based schedule will execute after a particular event takes place (for example, after new data is loaded into the warehouse).

Alert-based Proactive Delivery

Business users can set up delivery subscriptions that will execute only when certain conditions or thresholds in a report are met. This ensures that users are only alerted to any radical departure from a standard business condition enabling them to take prompt action. Any user can quickly set up a threshold on a report and create a subscription that will execute only when threshold condition is met. Alert deliveries can be triggered by one report and deliver a different report, dashboard, or document.

4.9 COMPREHENSIVE DOCUMENTATION OF METADATA

MicroStrategy provides comprehensive, interactive documentation of the metadata. From any object, all descriptions, definitions, formulae, object dependencies, and other properties are viewable. Extracts of the documentation can be published to ensure all users possess an intuitive understanding of the data model, reports, and other metadata objects.

Graphical Hierarchies Illustrate Relationships Between Logical Business Terms

MicroStrategy Architect's graphical hierarchy viewer displays the relationships between business attributes and underlying data elements. These graphical hierarchy views may be printed or saved, and published to business users so that they can easily understand the business model and attribute relationships.

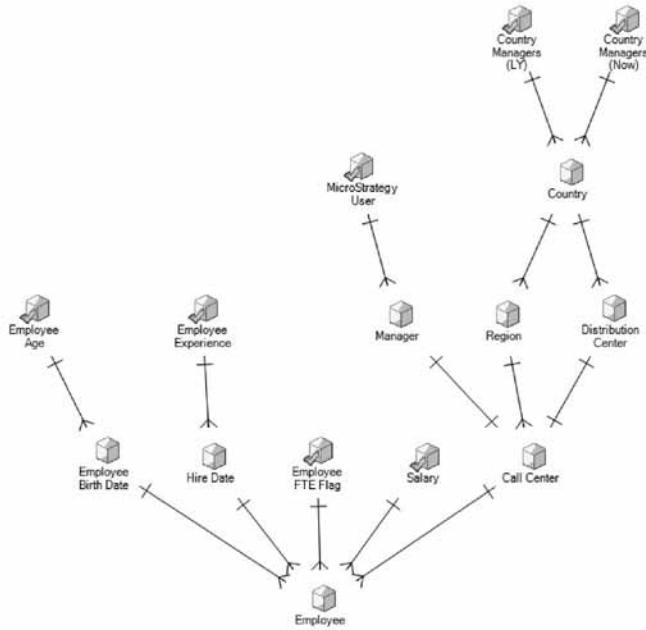


Figure 4-9 Viewing hierarchies graphically shows the relationships between the attributes in the hierarchy.

Interactive Project Documentation Gives a Detailed View of BI Applications

MicroStrategy Architect contains a Project Documentation Wizard that generates detailed documentation of the metadata. The wizard allows fine-grained control of the objects and the amount of detail to include in the documentation.

The wizard outputs HTML, enabling navigation through the complete metadata structure using the object dependencies. The output files may be printed, published on a Web server or shared drive, or stored in a configuration management system to track changes that occur in the metadata.

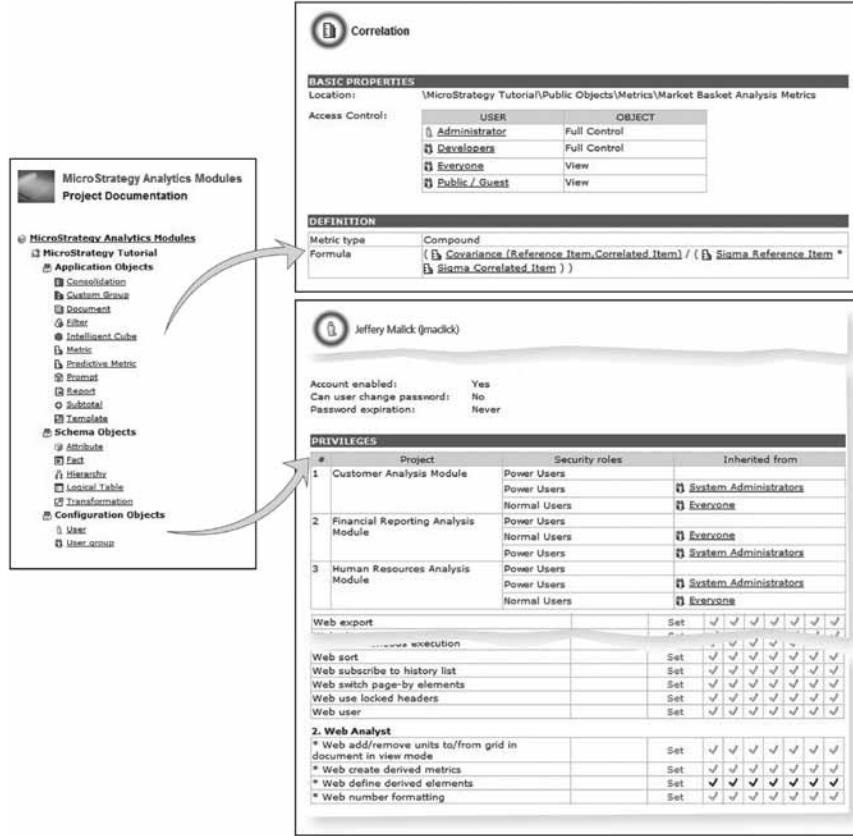


Figure 4-10 Metadata documentation includes important details about the metadata objects.

4.10 MULTI-LINGUAL BI APPLICATIONS

Many organizations have a global presence with workforces deployed in different countries using different languages. To support such a diverse workforce, it is important that critical business reports are made available in the language they prefer. That means that a business intelligence platform must support a single report definition that is automatically translated and presented to the business user in the language they expect while avoiding multiple copies of the same report. MicroStrategy BI platform enables organizations to develop multi-lingual applications by supporting internationalization of three important pieces, metadata objects, data translation, and interface translations.

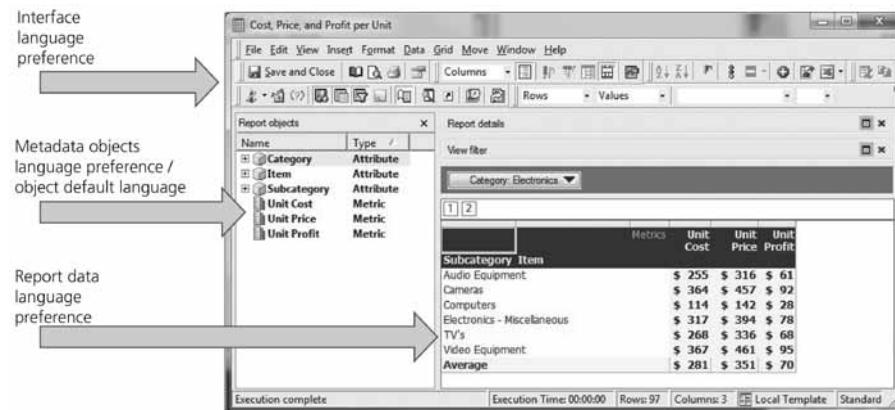


Figure 4-11 A sample report layout indicating the three components of a BI application that can be translated.

Multi-Lingual System Metadata

MicroStrategy provides translations for 20 languages³ and dialects out of the box for the metadata contained in a MicroStrategy. Translations are provided for user interfaces, system components, system messages, menus, and toolbars. A French user, who logins into MicroStrategy, would be presented with a French interface, would see MicroStrategy metadata objects with French names, and receive MicroStrategy Intelligence Server error and information messages in French.

Multi-Lingual User Defined Metadata Objects

With MicroStrategy, a single metadata object supports many translations of key properties, such a name and description. For example, a developer can create a single metric called 'Revenue' and translate its name, description, and long descriptions into any language or dialect. All translations are stored in the MicroStrategy metadata repository and the correct translation is used when a user is set up to use a particular language. When English users log into MicroStrategy, they see the metadata objects in English whereas French and German users will see the same metadata objects in their native language or dialect. There is a default language for each object to cover cases where translations have not yet occurred. Each metadata object has a translation editor that enables developers to easily enter translations for various languages. A bulk translation editor called the Metadata Repository Translation tool is available to extract and load translations for many metadata objects at the same time.

Name	Description	Modification Time	Default Language								
Revenue		3/4/2007 11:43:12 AM	English (United States)								
Object Name	English (United States)	Spanish (Spain)	German (Germany)	French (France)	Italian (Italy)	Portuguese (Brazil)	Swedish (Sweden)	Chinese (Simplified)	Chinese (Traditional)	Japanese	Korean
Revenue	Ingresos	Erltrag	Chiffre d'affaires	Entrate	Receita	Intakt	收益	营收	売上	수익	

Figure 4-12 A metadata object translation editor enables developers to add translations to the object names and descriptions.

Seamless Support for a Translated Data Warehouse

With a translated data warehouse, organizations can deploy a single report to their user base in different languages. Data internalization can be achieved in two different ways:

1. With Connection Mapping.

If the translated data is stored in different data warehouses, then the BI application can be configured to retrieve data from these warehouses depending on the language of the user. In this method, the report SQL will stay the same for each user but the translated data is retrieved from the language-specific warehouse via separate database connections.

2. SQL-based Data Internationalization.

When using SQL-based data internationalization, translated data is stored in a single data warehouse in separate translated tables or in additional columns of a lookup table. Both these techniques can be applied in a single MicroStrategy BI application. Depending on the user language, the MicroStrategy SQL Engine generates a SQL query that retrieves data for that language from the appropriate table or column.

When using separate tables, translated data is stored in the data warehouse using a specific table pattern. Patterns are generally in the form of a prefix or suffix added to the table name. For example, LU_ITEM_FR stores item level information in French and LU_REGION_FR will store region information in French. When a French user runs a report with Item on it, MicroStrategy SQL Engine will automatically uses the LU_ITEM_FR table to extract item descriptions in French.

When using separate columns, the translated strings for an attribute are stored in the same lookup table with additional columns storing the translations. The column names for the translations are identical except

³See Appendix G for a list of supported languages

for a prefix or suffix to identify the language. In the report, the MicroStrategy SQL Engine automatically queries the specific language column for the translated data and displays it to the user.

The table below shows the comparison of SQL-based and Connection Mapping-based Data Internationalization.

SQL – Based Data Internationalization	Connection Mapping Data Internationalization
All translated data in the same warehouse	Translated data in different warehouses
Different Report SQL for different languages	Same report SQL generated for the different language
Column based or table based data internationalization supported	Translated reports are generated via different ODBC connections

Figure 4-13 MicroStrategy supports localization of data by adjusting generated SQL or by connecting to different databases.

Interface Translations

In addition to metadata and data translation, MicroStrategy provides translations for interface menus, actions, and tool bars in MicroStrategy Mobile, MicroStrategy Web, and MicroStrategy Desktop, which use the language set in the Regional Settings on a user's computer or device. The language can be over-ridden by changing the language in MicroStrategy Desktop or in the MicroStrategy Web user preferences. The language also determines number formatting options for decimals and dates. When a French language user launches MicroStrategy Mobile, MicroStrategy Web or MicroStrategy Desktop, the language settings are used to provide French metadata translations and French report data seamlessly.

Metadata Repository Translation Tool

As the number of objects in the metadata increases, it is not scalable to provide translations for each object, for every language. To alleviate this problem, MicroStrategy BI platform provides a metadata repository translation tool to enable bulk translation of the metadata repository. This wizard-driven tool connects to the project metadata and extracts metadata objects into an external database, either Access or SQL Server, in a translation table. Translations of the metadata objects takes place independently in this database, and then imported back into the metadata repository.

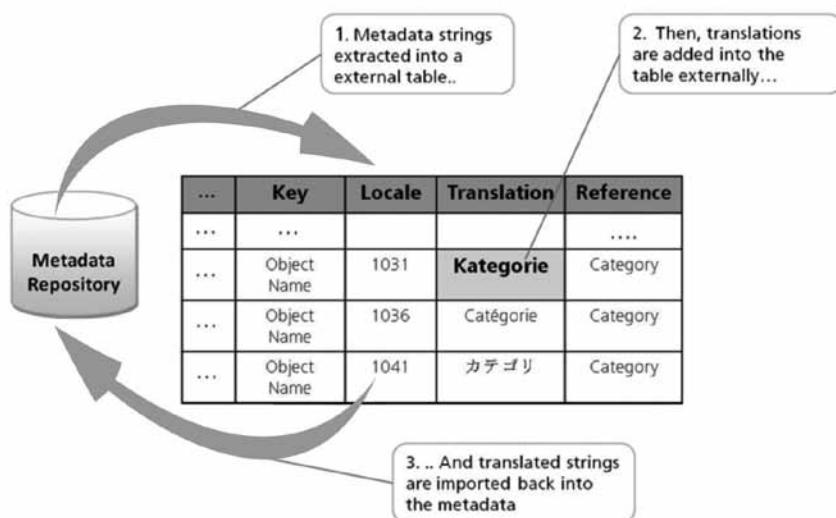


Figure 4-14 The Repository Translation tool extracts metadata objects for translation, and imports the translated strings once the translation is complete.

During the import process, the Metadata Repository Translation tool matches the metadata object from the string database to those in the metadata and populates the translations into metadata repository. It is important to note that MicroStrategy does not perform the translations themselves and they must be done independently.

4.11 OBJECT CHANGE JOURNALING

In any BI development environment, many objects go through several changes as business requirements change. Additionally, depending on the type of business requirement, these changes may involve several developers located in different locations. In such a dynamic environment, it is critical that each change is recorded into the metadata for auditing purposes. The capability to record each and every change is called object change journaling. Object change journals are stored in the metadata and are attached to the object.

A change journal stores when and by whom a change was made to an object. For every change, the critical information, such as user name, the type of change, time stamp, and comments, are recorded when the object is saved. Change journaling is integrated into the development workflow and requires no additional configuration.

A Permanent Record of Object Change History

An individual object change journal is a record of the object life cycle. It is available for every object in the metadata repository. Any MicroStrategy Desktop user with access to the object can right click on the object and view the object change history. Using object change journals, BI application developers and administrators can see the entire history of the object. In addition to individual object change journal, MicroStrategy has a monitor that presents a unified view of all changes across all applications in the BI system. As soon as a BI user makes a change in the system, it is recorded on the object and accessible in the change journal monitor. This provides the capability to conduct system wide audits of the BI environment.

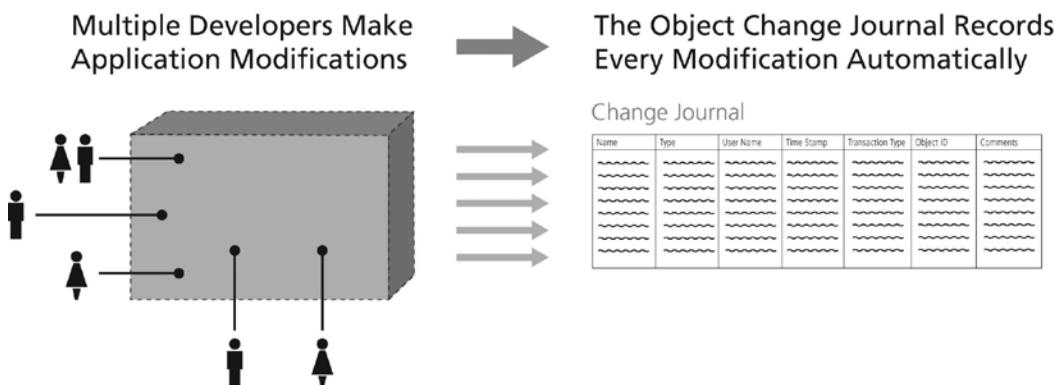


Figure 4-15 Every change made to a metadata object is automatically recorded into the change journal.

Change Journaling is Critical to Ensure Efficiency

Many organizations have distributed development teams that work on the same BI application to balance the development workload and ensure 24x7 support. These distributed teams must collaborate together to reduce lost development time. Change Journaling provides Business Intelligence teams the following advantages that radically improve their efficiency:

- Collaboration: A developer working on a metadata object can add comments to the Change Journal making it easy for other developers to know precisely the last change on the object. This eliminates any time that a developer otherwise would have to spend to manually identify the details of the last change.
- No External Change System: Since the change journal is stored in the metadata repository, there is no need to design or maintain an external documentation system.

- 24x7 Availability: Any BI user has access to the object's change history with a simple right click, as soon as they login into the BI system.
- Accountability: Since change journaling automatically records details such as user name, and timestamp of the change, it makes developers more accountable when they make changes to the object.
- Best Practices: Change documentation, an important yet ignored best practice, is embedded into the development workflow.

4.12 BENEFITS OF THE METADATA DESIGN

MicroStrategy designed and built its unparalleled metadata architecture based upon the following goals:

- Provide an abstraction of the physical data structures, and store these logical objects in a standard relational database
- Provide a single, centralized, and shared metadata for all reporting, analysis and monitoring
- Provide a fully reusable metadata
- Provide a truly dynamic and fully object-oriented metadata

These design principles result in tangible benefits for organizations that use MicroStrategy's BI platform, including:

- Intuitive reporting, analysis, and monitoring for all business users, and across all data sources
- The most efficient report development process
- Maximum maintainability and consistency across the entire enterprise

Abstraction from Warehouse Provides Intuitive Reporting, Analysis, and Monitoring for All Business Users Across All Data Sources

MicroStrategy's metadata abstraction translates the physical structure of a data source into easy-to-understand business terms, simplifying the deployment of reporting, analysis, and monitoring applications. There is no need to understand query languages such as SQL, be familiar with the underlying data model, or even be knowledgeable about the actual data source.

The different abstraction layers ensure that changes to objects in one layer do not materially affect other layers, and that changes are automatically propagated to all dependent metadata objects.

A comprehensive metadata is perhaps the single most important requirement for a true BI platform. It provides a way for a broad community of users to access valuable enterprise data using business terms, rules, and logic without requiring any technical skills or significant training.

Highest Number of Reusable Metadata Components

Most competing BI tools focus on developing individual reports as quickly as possible, neglecting the reusability of the report components, and sacrificing the ability to create a large number of reports efficiently. For true enterprise-class scalability, the BI architecture should allow each report to be developed more quickly than the last report, and should minimize the number of reports that need to be explicitly defined in the first place.

MicroStrategy has the broadest range of reusable objects in its metadata repository, from the data abstraction layer all the way to mobile apps, reports, dashboards, and documents. Even reports themselves can be re-used as filters for other reports, and as datasets for documents. Other BI technologies rely far less on reuse, and far more on each developer's ability to create the same components over and over again for use in each individual report.

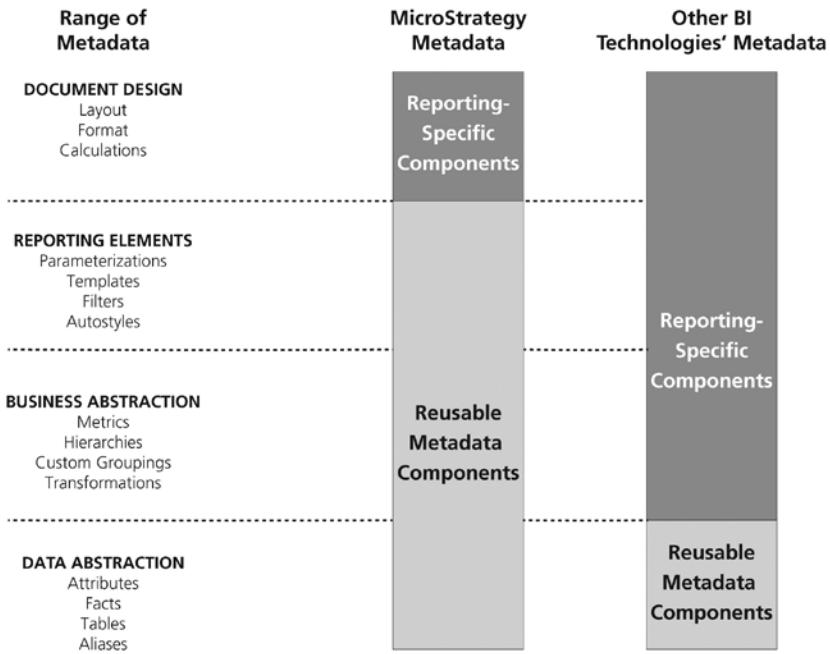


Figure 4-16 The MicroStrategy metadata architecture includes far more reusable components than any competing architecture, delivering much greater consistency across reports developed by different people.

Reusable Metadata Across All Styles of Business Intelligence

MicroStrategy's metadata object reusability spans all BI styles and applications. For example, a metric that uses a data mining algorithm is immediately available for mobile apps, alert notifications, operational reports, corporate dashboards, and ad hoc queries. With other BI tools, most of the business abstraction and report components need to be recreated separately each time a new application or style of BI is implemented. In addition to avoiding this extra development effort, MicroStrategy's reuse of objects ensures consistency across all BI applications and users.

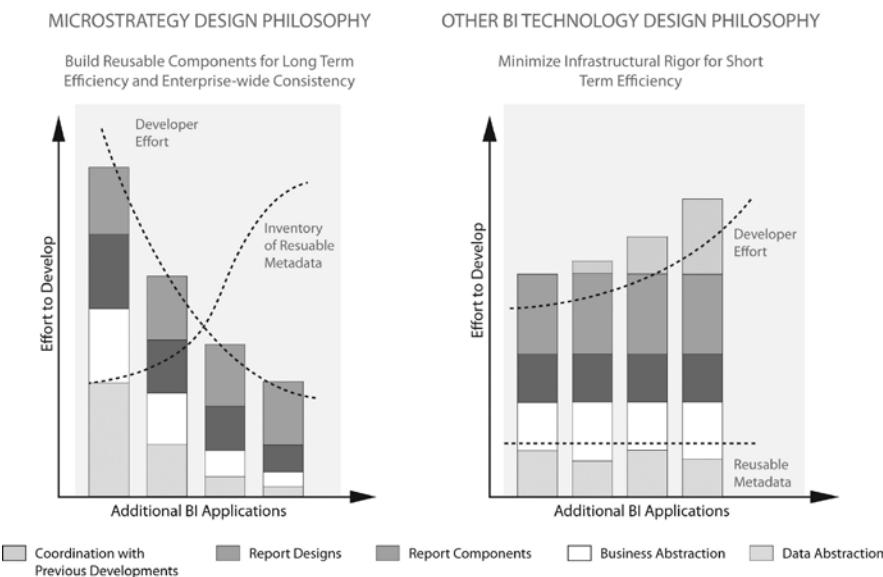


Figure 4-17 With MicroStrategy, new applications or styles of BI are deployed more quickly than the last because each application builds on all previous development. Other BI technologies actually require increasing effort to ensure consistency with all previous reports.

Dynamic Object Assembly Maximizes Maintainability

All MicroStrategy metadata objects are constructed using references to rather than copies of other metadata objects. When an object is used, all referenced objects are automatically accessed and merged into that object instance. For example, when a report is run, MicroStrategy dynamically assembles the latest versions of all underlying reusable objects into the report definition, and then generates optimized SQL or MDX syntax based on the assembled objects.

Any change to an object is automatically and instantaneously reflected in every metadata object that references that object. For example, if a metric is used to define a filter, and the filter is used in a prompt, then any change to the original metric will be automatically reflected in both the filter and prompt.

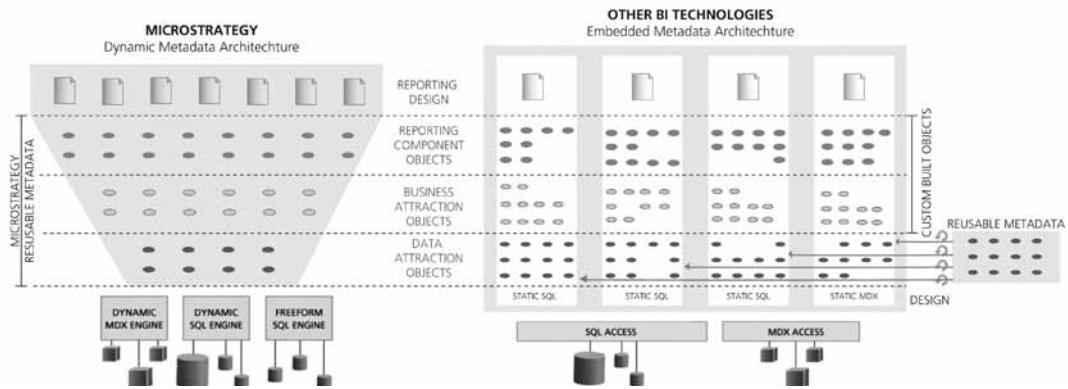


Figure 4-18 With MicroStrategy's Dynamic Metadata Architecture, each report definition contains "pointers" to metadata objects that are dynamically shared with other report definitions. By contrast, other BI technologies depend on embedded metadata where each report definition contains redundant copies of shared metadata.

Reusable Objects Ensure Consistency and a Single Version of the Truth

In the MicroStrategy metadata, there is only one definition of any object regardless of how many times it is used in different reports and objects. As a result, there are fewer objects to maintain. Additionally, since metadata objects map corporate data to the corporate business model, they ensure a single definition of any business entity in all BI applications deployed throughout the enterprise.

For example, a Top Stores filter, defined as "All stores whose sales exceed the average by 2 standard deviations," can be reused in many other objects: a top stores grid report, a top stores graph report, a metric, a filter, a prompt or other objects.

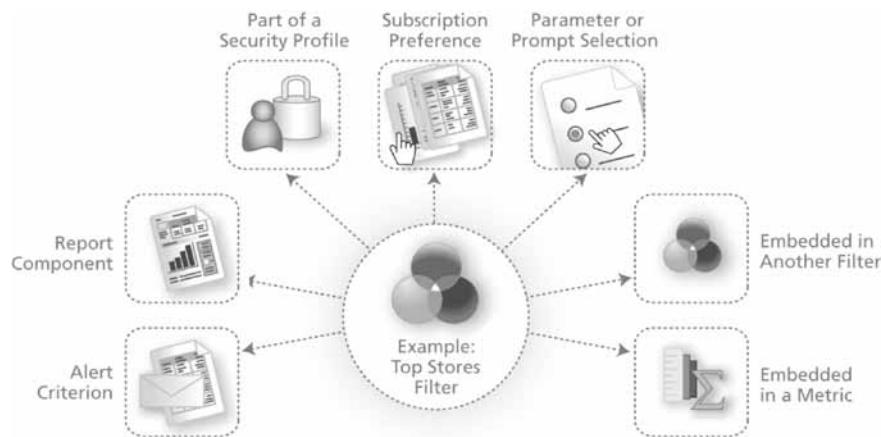


Figure 4-19 MicroStrategy's recursive object definitions ensure that object updates are necessary in only one place. All the dependent objects are updated immediately, transparently, and without any additional effort by the developer.

Ad-Hoc Querying with Highly Effective Drilling and Report Parameterization

MicroStrategy's dynamic assembly of report components delivers sophisticated, automated drilling anywhere in the data warehouse. A drill action incorporates the destination attribute and additional filtering criteria into the report definition dynamically before generating optimized SQL or MDX. In-depth investigative analysis is available from any starting report in a BI application.

Virtually any object can be parameterized and selected by the user at run-time using prompts. A single report definition caters for a large user population by allowing them to select the filter criteria, the attributes and metrics in the report, and even define dynamic calculations. This MicroStrategy capability reduces metadata sprawl, and speeds development as users are able to help themselves rather than rely on report designers to create custom reports.

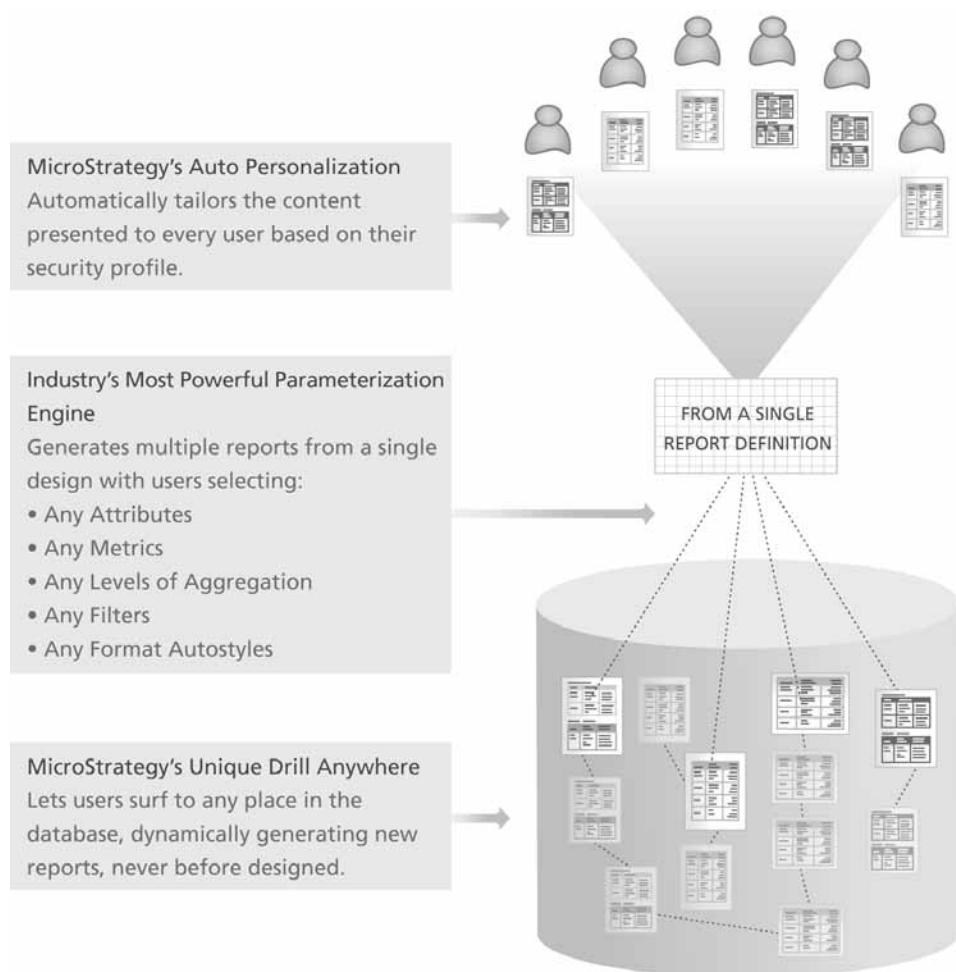


Figure 4-20 The MicroStrategy architecture routinely reduces the number of stored report definitions through automatic personalization, parameterization, and drill-anywhere.

Easy Metadata Exchange between different BI applications

MicroStrategy's metadata is designed for reusability and portability. A BI application developer can design a set of metadata objects such as attributes, metrics, facts, and hierarchies in one BI application and move them over to a totally different unrelated BI Application. All dependencies for metadata objects will be automatically migrated to the destination. This is especially useful when different unrelated BI applications

share a common metadata. For example, the finance and the marketing department may have different BI applications managed by different teams but may share a common set of Geography and Time dimensions and certain metrics. In this case, common metadata objects designed by one team can be quickly ported over into the BI application managed by the other team. There is no restriction on which type of metadata objects are ported.

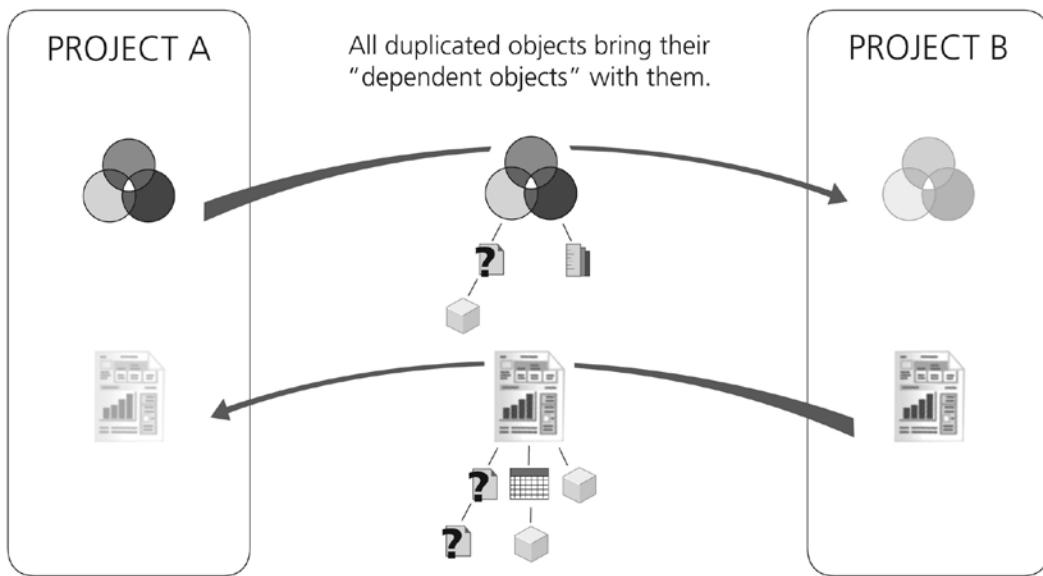


Figure 4-21 MicroStrategy metadata objects can be easily ported from one application to another.

4.13 SUMMARY

MicroStrategy's metadata architecture, built and refined over many years, and based on robust design principles, provides access to all enterprise data using familiar business terms, rules, and logic. The portable, flexible, reusable, object-oriented and dynamic nature of the metadata provides the most efficient BI application and mobile app development platform while maximizing maintainability and enforcing enterprise-wide consistency.

5

MICROSTRATEGY INTELLIGENCE SERVER ARCHITECTURE

The Business Intelligence (BI) standard for enterprises must provide a level of architectural and managerial sophistication that is equal to enterprise operational systems. The MicroStrategy BI platform service-oriented architecture is designed to power enterprise deployments while retaining ease and flexibility in satisfying all business needs. It offers the scalability, sophisticated analysis, extensibility, ease-of-use, performance, reliability, maintainability, and the 24x7 availability required by mission critical monitoring, reporting, and analysis applications.

The architecture is comprised of four main layers; a user interface layer, a server & data layer, a development layer, and an administration layer, with Intelligence Server as the backbone of the server & data layer that powers the entire platform. The design tenets for the MicroStrategy architecture meet extraordinarily stringent requirements for architectural integrity and manageability, including:

- Unlimited data scalability
- Unlimited user scalability
- Best-in-class performance
- Full range of analytical processing
- Any RDBMS or other data source
- Any user interface
- Personal, departmental, and enterprise deployments
- Highly manageable
- Single version of truth
- Automatic fault tolerance
- Open Standards conformance

To meet these challenges, Intelligence Server contains a group of well-defined, modular, and reusable components coordinated by a centralized management engine. This federated architecture within Intelligence Server provides maximum scalability, performance, and analytical power. Major highlights include:

- 64-bit technology
- Optimized for any database
- Sophisticated analytic power
- Multi-level secure shared cached and in-memory data
- Extensive governing parameters
- Adaptive, robust security
- Automatic resource allocation
- Flexible deployment options

The result is an offering unrivaled in the BI industry in offering enterprise-class BI capability along with the operational integrity of mission-critical ERP systems.

5.1 POWERING THE MICROSTRATEGY BUSINESS INTELLIGENCE PLATFORM

In the MicroStrategy Business Intelligence (BI) architecture, report processing is performed by Intelligence Server, which manages live and scheduled report requests, services these requests either from cache, in-memory cubes, or from the data sources, adds analytical insight and formatting to the results, and writes user input back to databases and Web services. Intelligence Server is the core processing and management tier for the MicroStrategy business intelligence platform, and is linked to the other platform components using a highly optimized communication framework.

The communication framework manages the information flow between different platform components, and provides secure user access, efficient report delivery, and rich data interactivity. Intelligence Server manages all report requests uniformly from any requesting source, including:

- MicroStrategy Mobile
- MicroStrategy Web
- MicroStrategy Office
- MicroStrategy Desktop
- MicroStrategy Distribution Services
- Third-party applications utilizing the MicroStrategy APIs

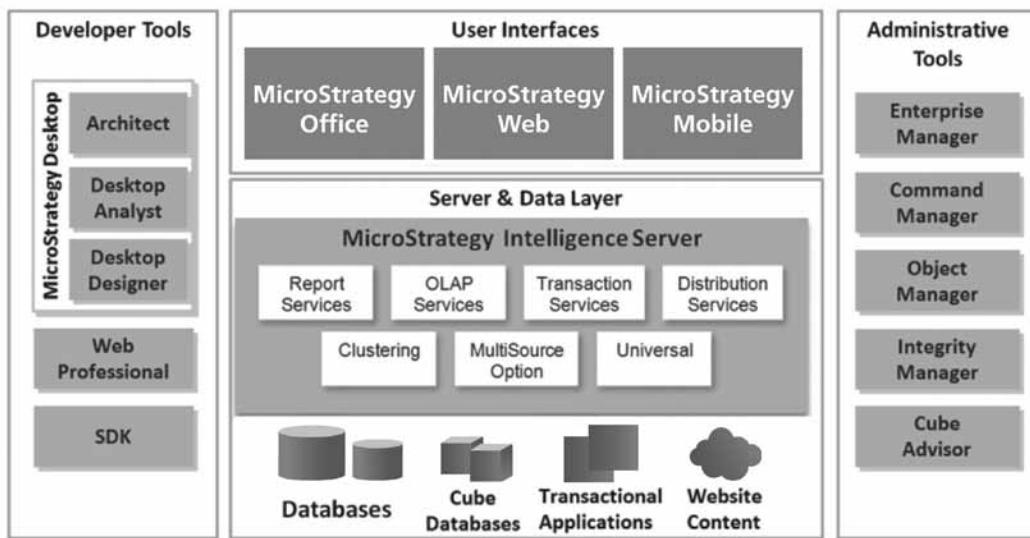


Figure 5-1 MicroStrategy Intelligence Server is the core of the MicroStrategy BI architecture.

Centralized BI Server Maximizes Productivity

This centralized architecture provides ease of maintenance, ease of management, integrated security, and one version of the truth to all users regardless of the interface. The entire BI system can conveniently be centrally controlled from Intelligence Server. In addition, Intelligence Server is reliable and fault tolerant, with automatic failover and fallback in the event of an unexpected fault. This architecture minimizes administration and maintenance, and maximizes security and performance of the business intelligence system.

5.2 INTELLIGENCE SERVER ENGINES AND QUERY FLOW

Although Intelligence Server is installed and administered as a single unit, it is built up from a group of federated, well-defined, modular, and reusable engines coordinated by a management engine. All the engines within Intelligence Server are multi-threaded for maximum throughput, and linked by a highly optimized communication bus. The main engines of MicroStrategy Intelligence Server are:

- Security Engine – manages user authentication, authorization, and encryption
- Object Browsing Engine – presents the folder navigation of the BI applications
- Prompting Engine – manages the presentation and resolution of prompts
- Resolution Engine – interprets business definitions for efficient execution
- Command Engine – manages all engine capacities and job queues
- In-Memory Data Engine – retrieves report data from Intelligent Cubes, caches, and history lists
- Dynamic SQL Engine – generates optimized SQL for relational data sources
- Dynamic MDX Engine – generates optimized MDX for multi-dimensional cubes
- Freeform SQL and XQuery Engine -- bridges business definitions with non-relational data sources
- Query Engine – directs data retrieval from the data sources
- Analytical Engine – performs additional calculations, cross-tabbing, and subtotaling of results
- Formatting Engine – adds rich formatting and converts results to the required output format
- Monitoring Engine – displays and logs current Intelligence Server activity
- Scheduling Engine – gathers and runs reports attached to time and event schedules
- Export Engine – offloads data intense data export operations
- Notification Engine – sends alerts and notifications to administrators
- Delivery Engine – ensures timely and reliable delivery of subscriptions and alerts
- Multi-Source Engine – merges data from multiple data sources into a single result set
- Transaction Engine – write user input to databases and Web Services

The query flow described below follows a typical job request to run a standard ad hoc report. To respond to this query, Intelligence Server dynamically processes multiple tasks with coordination provided by the Command Engine.

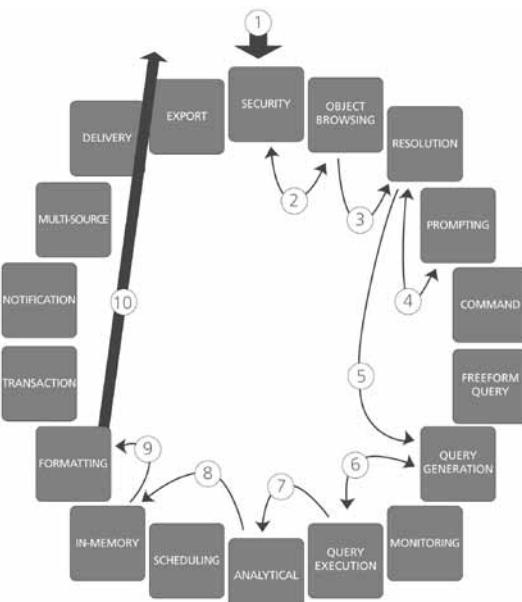


Figure 5-2 An ad hoc user request is processed by a number of engines as it passes through MicroStrategy Intelligence Server.

1. MicroStrategy Intelligence Server receives a logon request from any interface. The Security Engine authenticates the user, and allocates the appropriate authorization. If an external identity management system is used for authentication, the Security Engine receives authentication credentials from the third-party before providing the proper authorization information.
2. The user navigates through the available applications, folders, and business definitions to run a report. This activity is serviced by the Object Browsing Engine. The information is retrieved from the metadata repository or from caches via the In-memory Data Engine. If the metadata information is not cached, a connection is made to the metadata repository to retrieve the required object definitions. These object definitions are cached in memory, and managed by the In-memory Data Engine.
3. When the user runs a report, the information is handed over to the Resolution Engine for analysis. If the In-memory Data Engine determines that a valid Intelligent Cube or cache exists for this report, the cached result set is passed to the Analytic Engine and Formatting Engine before the final formatted report is sent to the user.
4. If prompts are included, the Prompting Engine manages the user input, resolves the prompt answers, and hands the information back to the Resolution Engine. Possible prompt answers are retrieved from cache if available. Otherwise, a connection to the data source is created via the Query Generation Engine.
5. Once all required information is gathered for query generation, the Query Generation Engine generates the optimized SQL for the specific data source. For multi-dimensional cube data sources, optimized MDX is generated.
6. The Dynamic SQL and MDX Engine hands the query text to the Query Engine for execution against the data source. The Query Generation Engine connects to the relevant data source, and runs the query. If a valid connection is cached, that connection is used to reduce overhead. Upon completion, the final report results are retrieved.
7. If intermediate calculations are required that are not supported by the data source, intermediate data is transferred back to the Query Generation Engine, and handed over to Analytical Engine for processing. The intermediate results are returned to the data source to complete the query. Steps 6 and 7 can be repeated many times if necessary. The report results are handed over to the Analytical Engine, which performs additional calculations, and applies cross-tabbing, page-by, and aggregation according to the report layout.
8. The final result set is added to the report cache, and indexed by the In-memory Data Engine. Other nodes in a clustered environment are notified that this report cache is available. This result set is now available for any matching requests from any user interface.
9. The Formatting Engine applies all the report formats – fonts, colors, images, thresholds, alignment, size, bands, graphs, and visualizations – and converts the data into the required output type – XML, Flash, PDF, Excel, CSV, and HTML. The XML and the final output are cached and made available for all other users.
10. The final formatted report is delivered to the user.

Throughout this process, the Command Engine allocates resources between different queries having different priorities among different engines. When multiple users execute multiple reports with multiple priorities, the Command Engine determines how each report will be processed. If a critical report is executed, the Command Engine can decide to suspend the execution of lower priority reports to service the critical report. In addition, when a particular engine comes under relatively heavier load, the Command Engine shifts processing threads to where they are required.

Engines Maximizes Flexibility and Code Efficiency while Maintaining Control

By bundling key processing logic into individual engines, Intelligence Server gains added agility to provide best-of-breed performance and data integrity. The benefits of this architecture include:

- Flexibility
- Code Efficiency
- Scalability
- Consistent Data
- Improved Monitoring

Resources available to these engines can be increased automatically to maintain optimum performance under changing system loads. Different engines provide the flexibility to interrupt processing to expedite higher priority requests. Engines are reused and revisited improving the efficiency of the system. With no redundant functionality among engines, consistency is enforced, the same security is always applied, and the same data is always presented.

5.3 AUTOMATIC RESOURCE ALLOCATION

MicroStrategy Intelligence Server is a fully multi-threaded application. Each engine in Intelligence Server is a discrete processing unit and contains a dynamic number of processing threads to perform its tasks. With multi-threaded processing units, each unit can support multiple users executing different reports asynchronously. In addition, each processing unit can leverage all available processors in a multi-processor system, enhancing the scalability and performance of Intelligence Server. At startup, the Intelligence Server assigns a specific number of processing threads to each engine. After startup, the Command Engine dynamically reallocates threads from one engine to another as the demand for individual engines varies depending on the nature of user activity and reports executed. This adaptive balancing of internal capacity ensures effective use of available resources, and allows the MicroStrategy BI platform to scale while maintaining excellent performance.

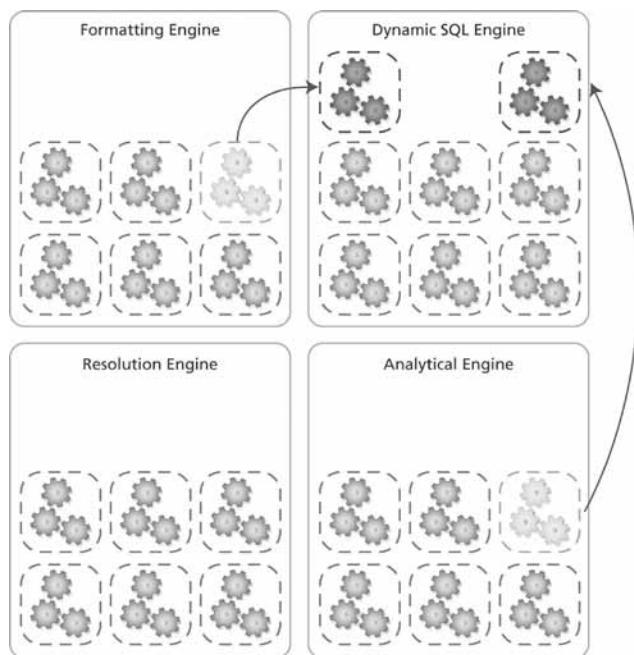


Figure 5-3 Threads are automatically moved to the SQL Engine from other engines to remove a SQL processing bottleneck.

For example, increased demand for reports that contain extensive analytical processing places a higher demand on the Analytical Engine. If the results for most of these reports are cached, the Dynamic SQL Engine will be under a low load. Intelligence Server may reallocate some of the Dynamic SQL Engine threads to the Analytical Engine to process more analytical calculations in parallel. Automatic thread balancing is enabled or disabled in MicroStrategy Desktop.

5.4 64-BIT BUSINESS INTELLIGENCE

The amount of memory a BI product can use is determined by the size of addressable space available. The change from 32-bit to 64-bit is a fundamental alteration. For a BI product to be fully 64-bit ready, it not only needs to execute on a 64-bit operating system, it also needs to support 64-bit datatypes. While other BI products claim 64-bit capable, they may simply execute on 64-bit operating systems when in fact the applications are still operating under 32-bit limitations. MicroStrategy Intelligence Server is a fully ported 64-bit BI application supported on 64-bit operating systems. With fully ported 64-bit Intelligence Server, the capacity of each Intelligence Server instance is increased multi-fold. The benefits include:

- Increased number of users
- Increased in-memory capacity and improved performance
- Richer BI applications through increased metadata capacity
- Reduced number of physical servers

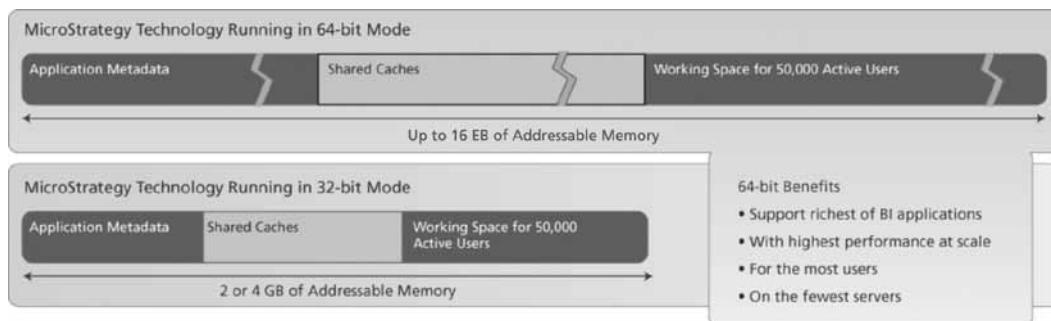


Figure 5-4 Native 64-bit technology in MicroStrategy supercharges BI.

True 64-bit technology increases the amount of data that Intelligence Server can store and access in memory. MicroStrategy 64-bit BI technology seamlessly scales with the growth in users and data sizes while delivering ultra-high performance.

5.5 CLUSTER-CAPABLE INTELLIGENCE SERVER

In clustering, a group of independent systems, called nodes, work together as a single system. In the MicroStrategy BI platform, multiple Intelligence Servers running on separate machines are clustered to work together as a single logical system. Each MicroStrategy Intelligence Server is cluster-capable, and requires no additional third-party software. It adheres to common requirements for hot, warm and cold failover configurations. Compared to the clustering architectures used by other BI products, the MicroStrategy Intelligence Server cluster is an all-active peer-to-peer configuration, and is built to minimize resource overhead.

This autonomous configuration minimizes the network traffic by requiring no central management service for clustering. The communication between each node is highly tuned, allowing for maximum information density per communication.

The features of Intelligence Server clustering include:

- Flexible Load Balancing
- Automatic Failover and Fallback
- Dynamic Asymmetric Clustering

Shared Data and Definitions Maintain a Single Version of Truth

Each node in a MicroStrategy Intelligence Server cluster connects to the same metadata repository as the other nodes, ensuring that BI users, applications, business definitions, security, data, and in-memory caches are identical for all nodes in the cluster. Report results are always the same regardless of which node executes the report. Even when new or updated definitions are added to a node, any changes are immediately available to all nodes within the cluster. This ensures data integrity, and provides a single version of the truth.

Adding nodes to a single Intelligence Server deployment or to an existing cluster requires minimal administrative effort. Aside from simply ensuring that each Intelligence Server node can access the other nodes via the network, a new node simply needs to load the same Server Definition in the MicroStrategy Metadata Repository. Modifying cluster membership is easy and straightforward using either MicroStrategy Desktop or MicroStrategy Command Manager.

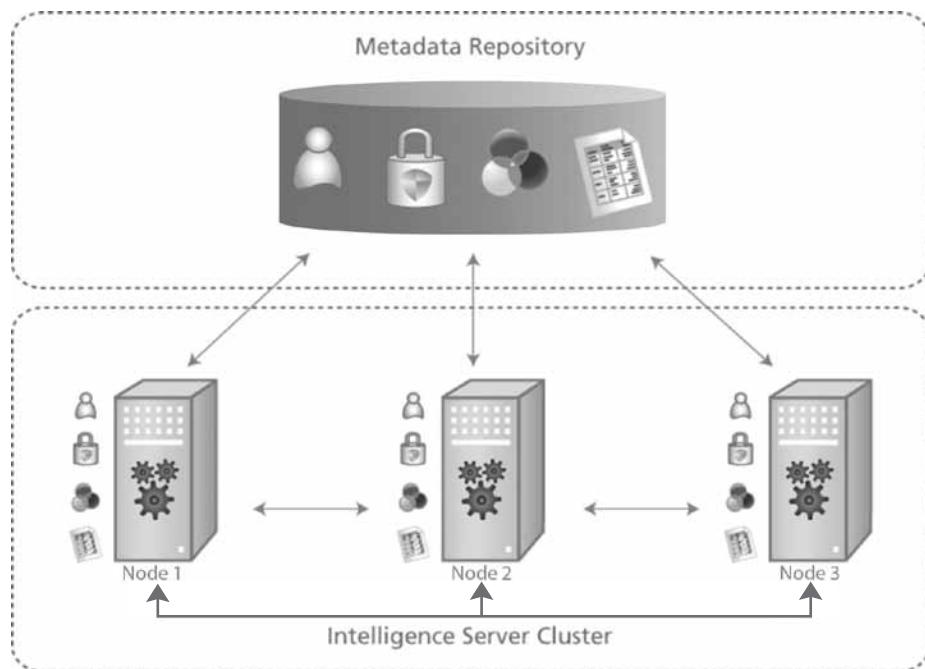


Figure 5-5 All nodes share the same security, business definitions, data, caches, and users, ensuring a single version of the truth for all business terms and measures.

5.6 OPTIMIZED FOR HETEROGENEOUS DATA SOURCES

Business needs often dictate that information is retrieved from many different data sources. A business intelligence server must be able to communicate with any data source and perform analysis that spans the boundaries of a single data source to fulfill these requests. However, different data sources have different connectivity mechanisms, optimization techniques, and result formats. Intelligence Server integrates with all major data sources, is optimized to extract data from these data sources, and can perform data analysis across different data sources into a single coherent data set.

This level of sophistication is reached by adhering to common industry standards that build a flexible, but reliable and well established foundation for data communication. Intelligence Server relies on Open Database Connectivity (ODBC) standard and the Structured Query Language (SQL) to query relational databases and uses extensible markup language for analysis (XML/A) over Simple Object Access Protocol (SOAP) based connectivity with the Multidimensional Expressions (MDX) query language for cube databases.

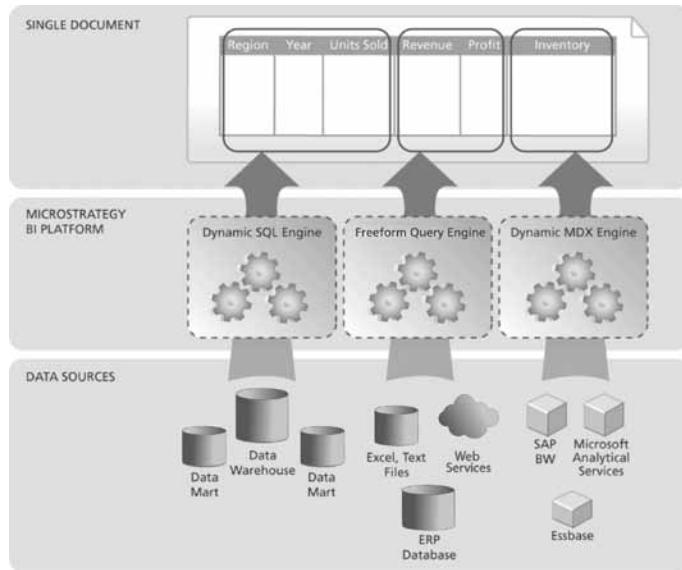


Figure 5-6 A single dataset contains data from heterogeneous data sources.

Open Database Connectivity (ODBC)

The MicroStrategy BI platform accesses many database systems using a common ODBC layer. MicroStrategy installs wire protocol ODBC drivers which communicate directly to the database through the database's own wire level protocol, eliminating the need for installation of the database's native networking drivers. This approach effectively removes an entire communication layer, further improving system performance and reducing system complexity.

The use of traditional ODBC drivers and the database's native networking drivers requires database client libraries for each version of database accessed by the BI application. By contrast, when using the MicroStrategy BI platform, only one wire protocol ODBC driver per database brand is required. This dramatically reduces administrative overhead and potential conflicts because one wire protocol ODBC driver services all database versions from the same vendor.

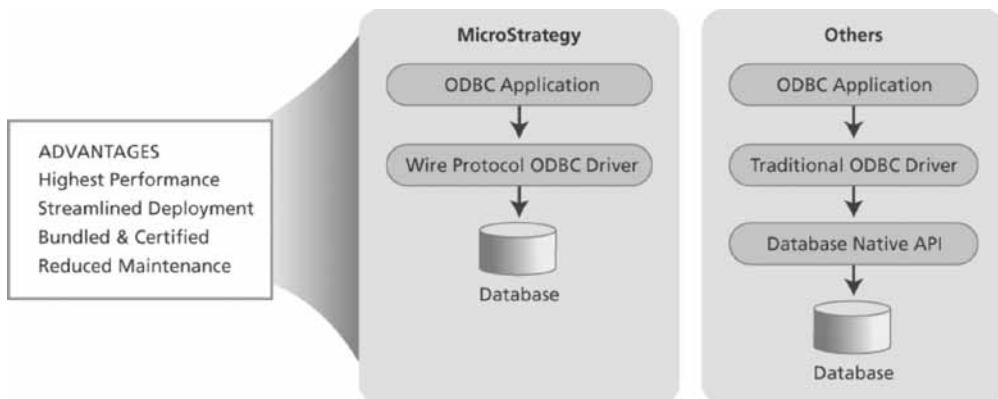


Figure 5-7 Wire protocol ODBC drivers used by MicroStrategy are more efficient than traditional ODBC drivers.

Dynamic SQL Engine

Database vendors invest heavily in optimizing their relational database management systems (RDBMS) to keep up with the proliferation of data. Each RDBMS provides many strategies to deal with this trend. The MicroStrategy BI platform, with its Relational OLAP (ROLAP) architecture, complements these strategies. MicroStrategy uses the RDBMS for much of its processing, making the most of the RDBMS investment. No time-consuming and costly additional processing is needed to load the data into a datamart or proprietary cube before users can access data.

Since each database uses its own optimized Structured Query Language (SQL) syntax, a generic SQL statement performs very differently on different database platforms. The performance difference can be orders of magnitude depending on the complexity of the query. Intelligence Server generates optimized SQL syntax specific to each database platform and version. The generated SQL can be modified through VLDB (Very Large Database) settings that control how certain SQL phrases are written. VLDB properties are available for projects, metrics, templates, Intelligent Cubes, and reports . MicroStrategy Intelligence Server optimizes performance in several ways:

- Multi-pass SQL
- Aggregate Awareness
- Very Large Database (VLDB) properties

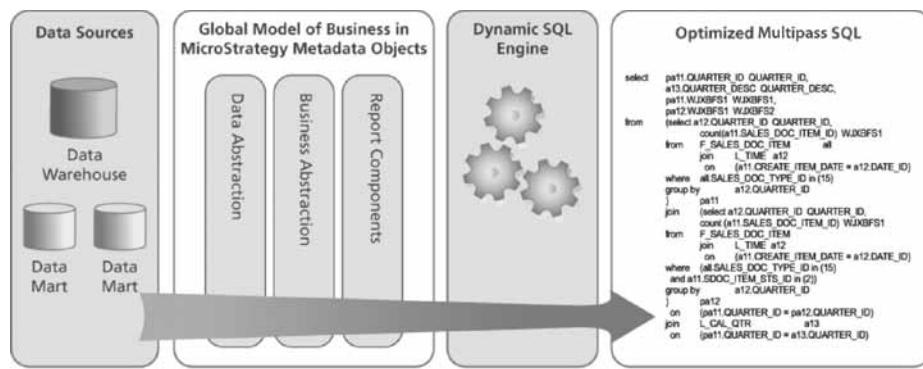


Figure 5-8 MicroStrategy automatically generates multi-pass SQL that is optimized for the specific database being used.

MicroStrategy SQL Engine automatically generates a query that fetches translated data from an appropriate database, column, or table determined by the internationalization settings on the metadata objects used in the query and the language setting of the user. This process is seamless to the end user.

Dynamic MDX Engine

MicroStrategy's dynamic Multidimensional Expressions (MDX) Engine generates optimized MDX syntax for multi-dimensional data sources. Firstly, MicroStrategy connects to SAP using high performance SAP Business Applications Programming Interface (BAPI®) via Java Connectors. For other cube data sources, the connections use extensible markup language for analysis (XML/A) as a Simple Object Access Protocol (SOAP) interface over Web services.

MicroStrategy's MDX is dynamically generated using the multi-dimensional models imported from SAP InfoCubes, QueryCubes, Operational Data Store (ODS) objects, and from other multi-dimensional cubes. End users can access all the information contained in these data stores directly through the MicroStrategy BI platform. All existing levels, characteristics, hierarchies, key figures, measures, variables, and structures in the cubes can be accessed by end users, ensuring that development efforts are not duplicated. Drilling back into

⁴See Appendix B for a full list of VLDB properties

the multi-dimensional cube for more data is automatic and transparent, without any programming or prior design of drill paths.

BAPI takes advantage of all SAP Business Information Warehouse (BW) capabilities, and provides an open, platform-independent interface to access any information that is available through the SAP BW OLAP engine. MicroStrategy is SAP Certified - Powered by SAP NetWeaver® and OLAP Business Applications Programming Interface (BAPI) for SAP BW solution.

XML/A uses XML to transmit MDX queries to other multi-dimensional cubes on any platform, allowing access to data stores that comply with these standards. Using XML/A, the MicroStrategy BI platform fully utilizes OLAP functionality within the multi-dimensional data sources.

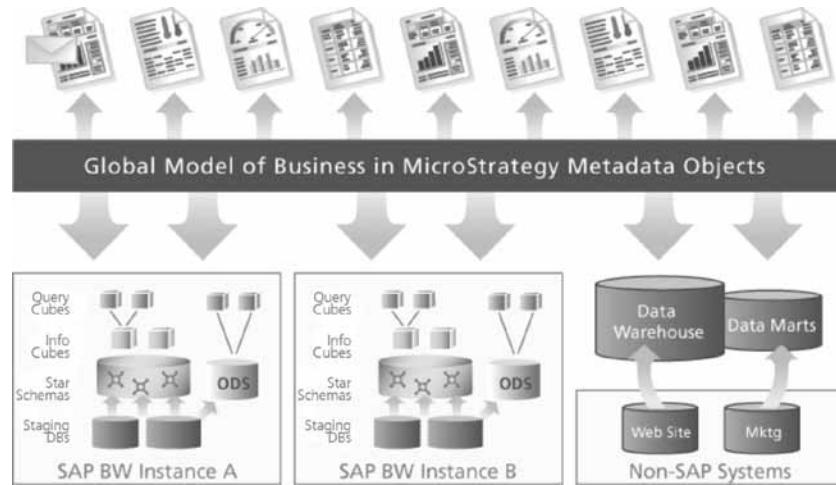


Figure 5-9 Accessing multiple SAP BW Instances together with other data sources.

Freeform Query Engine

The Freeform Query Engine in Intelligence Server provides access to data stored in any format accessible by ODBC and SQL, including Microsoft Excel files, CSV text files, and other operational databases of systems such as ERP systems, RFID applications, Web Site tracking, Call Center tracking, and CRM systems. It also accesses Hadoop clusters using HiveQL and PIG, and Web services using XQuery. Hand-crafted queries are embedded directly into the MicroStrategy BI platform to leverage existing investments such as proprietary stored procedures, SQL routines, and SOAP and RESTful Web services. Freeform query statements are subject to the full suite of security features in the MicroStrategy BI platform. Prompts can also be included, increasing the flexibility of the queries.

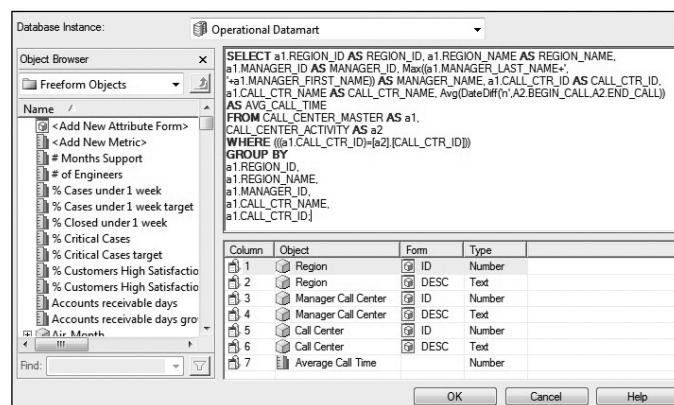


Figure 5-10 Freeform SQL provides the agility to inject prototype data prior to formal inclusion in the data warehouse.

Freeform queries also provide the agility to rapidly inject information from new data sources into the MicroStrategy BI platform. Prior to committing new processes to extract, transform, and load data from these new source into the enterprise data warehouse, new data can be retrieved and presented to business users through existing reports in an already familiar environment and format. In addition, data can be imported directly into in-memory Intelligent Cubes from files and databases from MicroStrategy Web.

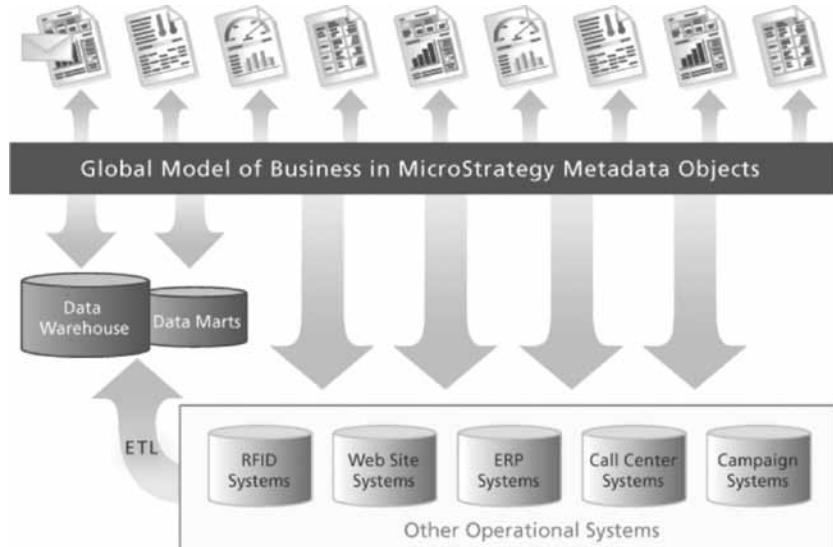


Figure 5-11 Direct access to operational data together with other data sources.

Joining Data from Different Data Sources with Federated Data Access

MicroStrategy MultiSource Option extends Intelligence Server allowing users to seamlessly report, analyze, and monitor data across multiple data sources through a single multi-dimensional view of the business. It employs a federated data architecture that pushes calculations and data joins down to the database level. This unique approach requires no additional layers as it leverages existing infrastructure to provide a single view across many data sources. Business users can seamlessly report, analyze, and monitor data across multiple SQL and MDX sources. Companies can get BI applications up and running in almost no time with minimum data engineering.

Most BI technologies join data in the BI server, thus requiring considerable hardware resources to move and analyze tremendous amounts of data. In contrast, MicroStrategy's federated data access technology employs a push-down architecture that utilizes existing database engines for complex calculations and data joins. This architecture is much more efficient than mid-tier joins as it:

- Requires much less server capacity
- Minimizes data movement across the network
- Leverages existing database infrastructure

After executing optimized queries against each different data source, the results from the data sources are merged before the MicroStrategy BI platform retrieves a single unified dataset.

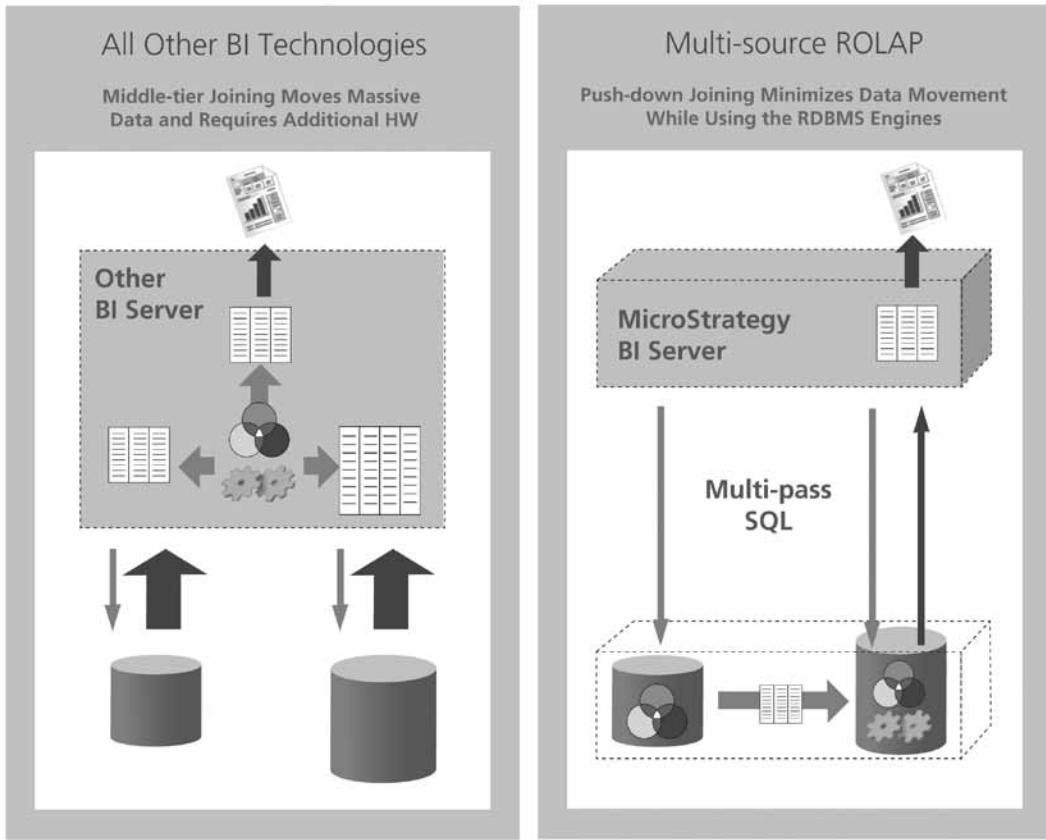


Figure 5-12 Federated data access provides the highest performance at a minimum cost by harnessing existing database infrastructure.

Intelligence Server also combines several datasets for display in the visualization layer. In this scenario a document is specified to display information from multiple datasets. When common data exist within the datasets, these are joined within the Intelligence Server. When there is no common data, Intelligence Server builds the relationships internally instead of simply performing a cross join. Unlike other BI products, this approach optimizes memory usage, reduces processing time, and enhances performance.

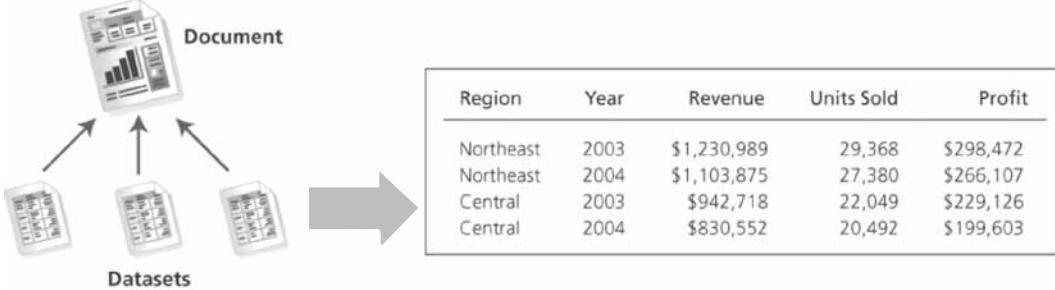


Figure 5-13 Joining data from distinct data sources into a virtual dataset is automatic and efficient.

5.7 SOPHISTICATED ANALYSIS

It is not sufficient for an enterprise-class BI platform simply to retrieve information from a data source. Business users must be able to apply analytical richness to the data to gain additional insight for accurate, data-driven corporate performance monitoring and decision making. MicroStrategy enables questions, such

as how the business is doing, what problems exist, how to solve them, and how to improve operations, to be asked at the enterprise level with a consolidated view. To achieve this, the MicroStrategy BI platform:

- Supports a wide range of analytical functions
- Applies these functions across all data sources in the enterprise
- Supports commonly used analytical techniques
- Performs predictive data mining
- Offers rich presentation methods for easy data consumption

Metrics and key performance indicators available in the MicroStrategy BI platform range from simple aggregations, and counts to advanced financial and statistical models and predictive data mining analysis. With more than 270 analytical functions and operators, MicroStrategy empowers organizations for better monitoring, reporting, and analysis.

$$\text{CallOptionPrice}_{\text{black-Scholes}} = S \text{StdNormalDist}(d_1) - X e^{-r\tau} \text{NormDist}(d_2)$$

$$\text{where } d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(r + \frac{1}{2}\sigma^2\right)\tau}{\sigma\sqrt{\tau}} \quad \text{and } d_2 = d_1 - \sigma\sqrt{\tau} \quad \text{Where}$$

$$\text{PutOptionPrice}_{\text{black-Scholes}} = X e^{-r\tau} - S + \text{CallOptionPrice}_{\text{black-Scholes}}$$

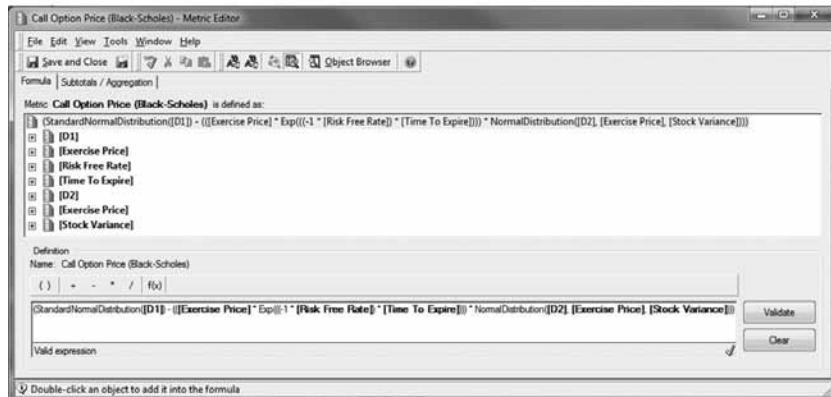


Figure 5-14 MicroStrategy analysis can be as sophisticated as required. This example shows the Black-Scholes option pricing model.

Prompts Control the Scope of Analysis

The Prompt Engine interacts with end users to manage all the prompts contained in a report. MicroStrategy prompting allows the user to dynamically control the complete scope of an analysis, not just the filtering criteria. An analysis can be made more comprehensive or more focused by answering simple questions like When, Where, Who, What, and How.

Prompts expand the analysis range of a single report definition by giving users the flexibility to decide which attributes and metrics to include at the time the report is run. Users have the flexibility to expand the scope of their analysis to any part of the enterprise data warehouse dynamically, without the need for new reports to be created. Prompts also focus the analytical process by providing filtering criteria that limit the cross-section of the enterprise data warehouse to include in the analysis.

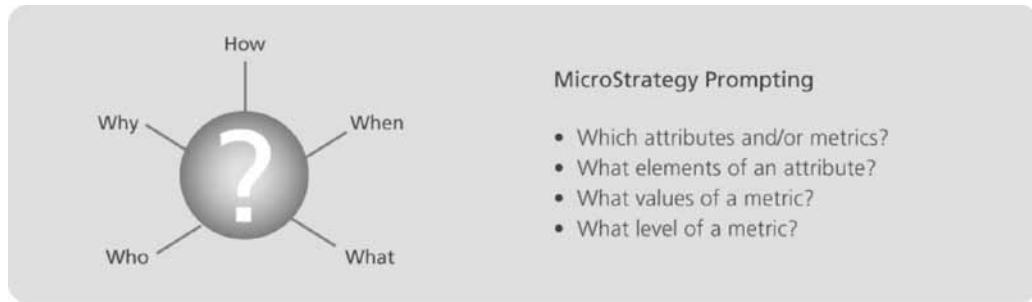


Figure 5-15 Control scope of analysis with MicroStrategy prompting.

Flexible Aggregation Defines the Level of Analysis

An important aspect of analysis is the level at which individual metrics are aggregated. In many cases, specific groups are compared against their entire peer groups. For example, following an advertising campaign targeted at Boston, New York, and Washington, D.C., the marketing analyst can use MicroStrategy to conduct cross strata analysis as follows:

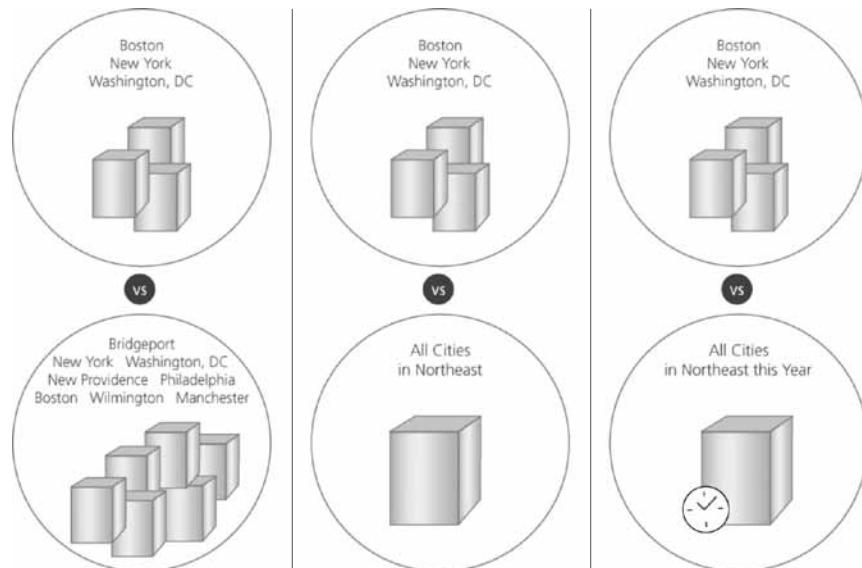


Figure 5-16 Flexible aggregation powers a single report for comparative analysis at different levels.

Each analysis above sheds new light on the impact of the campaign. With other BI products, multiple reports will be defined with complex custom SQL coding to answer each question. MicroStrategy prompts provide users the flexibility to choose, “on the fly”, the level at which a specific group should be compared. In all circumstances, data security continues to be enforced.

Additional Enrichment of Data in the Analysis

MicroStrategy Intelligence Server shares analytical processing with the database platform. When the database natively supports a function, it is included in the SQL query. If the database platform does not support a function, Intelligence Server first attempts to translate the function into a formula that the database can support. If a translation is not possible, the data is retrieved, and passed to the Analytical Engine, which uses its internal functions to calculate the metric. Intelligence Server dynamically performs these tasks without any intervention from the administrator or user. This automatic mechanism answers all user questions, regardless of whether the database supports the underlying analytical function.

Business users often need to define their own data groups or clusters of data elements, irrespective of how data is grouped in the database. The data schema within the database cannot always satisfy dynamic business needs. For example, users might want to find the Top 10% of the most profitable customers by different attributes displayed in separate rows on a report. Since this data changes every time new data is loaded into the data warehouse, maintaining this grouping in the database causes significant additional overhead. Intelligence Server solves this problem with Derived Elements, Custom Groups and Consolidations, in which attributes can be grouped into sets that define specific segments without modification to the underlying data. Complex segmentation SQL is automatically generated without end-user coding.

Standard set operations – union, intersection, exclusion, and difference – can be applied across more than one attribute set to refine the segmentation criterion. This is called Set Analysis. For example, in order to isolate a list of target customers for a promotional campaign, sets of attributes can be used to identify customers who have purchased the promotional items or products with a high correlation to the promotional items. Using these multiple sets, a marketing department can isolate target customers, and coordinate supporting activities with stores and suppliers to work towards a successful campaign.

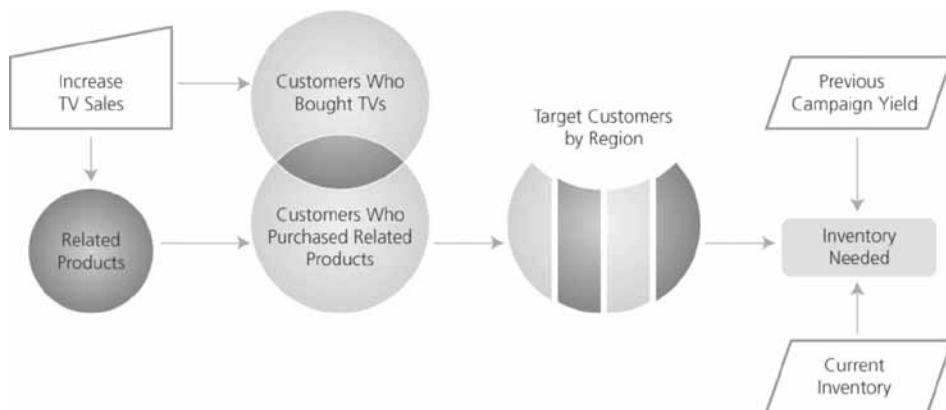


Figure 5-17 Set Analysis to segment a list of target customers for a TV promotion.

Iterative Analysis Technology

Iterative analysis is the dynamic mechanism that represents the continuous interaction and advanced analytical techniques between Intelligence Server and the database. MicroStrategy uses this approach when complex questions cannot be answered using only the database, or when a complex question that requires several steps cannot be answered by a simple SQL query. Not only does the Intelligence Server automatically invoke iterative analysis when required, it also dynamically decides the most optimal location to perform the calculations, based on its own strengths and the database strengths, while minimizing data transfer. The MicroStrategy BI platform is the only BI platform in the industry that performs iterative analysis.

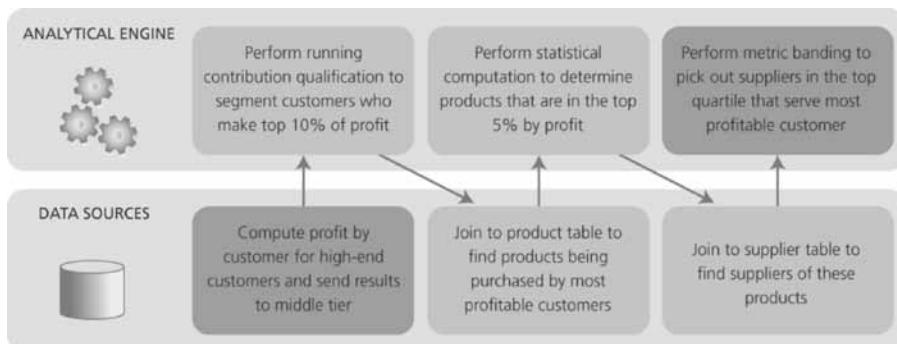


Figure 5-18 Complex iterative analysis is automated and simple without user coding.

Data Mining Integration into Enterprise BI

Data mining is the process of sifting through large amounts of historical data in search of hidden patterns and predictive information using statistical analysis, modeling techniques, machine learning, and database technologies. In other BI architectures, organizations are limited to historic analysis within a Business Intelligence solution, and are forced to perform predictive analysis in a separate system. With MicroStrategy, data mining training and scoring is fully integrated into mainstream business intelligence. This complete integration elevates the availability of predictive metrics to all decision makers and business users.

The integration of data mining models from other applications is accomplished by importing Predictive Model Markup Language (PMML) into the metadata repository, and automatically creating a predictive metric. PMML is an XML-based industry standard developed by the Data Mining Group (DMG) to describe predictive models. It supports a number of different data mining algorithms, including Neural Networks, Clustering, Regression, Decision Trees, and Association. MicroStrategy is the first Business Intelligence platform to support the PMML standard. If predictive models are still developed in specialized data mining applications, MicroStrategy manages the data scoring and distribution of the results across the enterprise.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?pml version="4.0"?>
<?xml-stylesheet type="text/xsl" href="http://www.dmg.org/PMML-4_0.xsl"?>
<?copyright "(C) 2011 by MicroStrategy, Inc."?>
<?application name="MicroStrategy Decision Tree Analytics" version="9.2.200.042"?>
<?timestamp "Tue May 05 17:53:40 2011"?>
</headers>
<?discretionary numberOfTables?"?>
<- <dataField dataType="string" name="Marital Status" optype="categorical">
<- <value property="valid" value="Married" />
<- <value property="valid" value="Previously Married" />
<- <value property="valid" value="Single" />
</dataField>
<- <dataField dataType="string" name="Age Range" optype="categorical">
<- <value property="valid" value="21 to 34" />
<- <value property="valid" value="35 to 44" />
<- <value property="valid" value="45 to 54" />
<- <value property="valid" value="55 and over" />
</dataField>
<- <dataField dataType="double" name="IsLostCustomer" optype="categorical">
<- <value property="valid" value="1" />
<- <value property="valid" value="0" />
</dataField>
</dataDictionary>
<?treeModel functionName="classification" modelName="Churn Predictor" splitCharacteristic="binarySplit"?>
<extension extender="MicroStrategy" name="k-fold" value="3" />
<- <miningSchema>
<- <discretize>
<- <discretize invalidValueTreatment="asIs" missingValueTreatment="Simple" missingValue="<Missing>" />
<- <discretize invalidValueTreatment="asIs" missingValueTreatment="asMode" name="Marital Status" outliers="asIs" usageType="active" />
<- <discretize invalidValueTreatment="asIs" missingValueTreatment="asMode" name="Age Range" outliers="asIs" usageType="active" />
</discretize>
<- <miningSchema>
<- <outputField dataType="double" feature="predictedValue" name="Retail Churn Predictor (Tree)" optype="categorical" />
<- <outputField dataType="double" feature="probability" name="Retail Churn Propensity (Tree)" optype="continuous" value="1" />
</output>
<- <discretize>
<- <univariateStats field="IsLostCustomer">
<- <Counts cardinality="2" missingReq="0" totalReq="9257" />
<- <descStats modalValue="0">
<- <array n="2" type="int">2730 6527</array>
</descStats>
</univariateStats>
<- <univariateStats field="Marital Status">
<- <Counts cardinality="3" missingReq="0" totalReq="9257" />
<- <descStats modalValue="Single">
<- <array n="3" type="int">3735 1443 4509</array>
</descStats>
</univariateStats>
<- <univariateStats field="Age Range">
<- <Counts cardinality="5" missingReq="0" totalReq="9257" />
<- <descStats modalValue="35 to 44" />
<- <array n="5" type="int">661 1634 3181 1492 2289</array>
</univariateStats>
</treeModel>
<Tree Style> Large Vertical <input checked="" type="checkbox" /> Use Display Names <input type="checkbox" /> Show Model View <input type="checkbox" /> Use Parentheses to Group Terms <input type="checkbox" /> One Term per Line <input type="button" value="Export" /> <input type="button" value="Close" /> <input type="button" value="Help" />
```

Figure 5-19 Predictive Model Markup Language creates MicroStrategy data mining metrics.

MicroStrategy Contains the Broadest Analytical Capabilities

The following table contains examples of the analytical capabilities available in the MicroStrategy BI platform. BI applications use these capabilities to add analytical richness to enterprise data.

Type of analysis	MicroStrategy Analytics	Description
Data Summarization	Sum, counts, variance, mean, mode, median	Summarization allows users to have an overview of their business performance. For example, users can view histograms of customer spending by product or age.
Time Series Analysis	Running and moving averages, TY-LY, MTD, YTD, transformations	Time Series Analyses allow users to dynamically compare data from various time periods. For example, users can accurately forecast future purchasing behavior using regression analysis or chart the 12-month moving average of the stock price.
Segmentation Analysis	Custom groups, consolidations, deciling, pareto analysis	Segmentation Analyses permit users to pull together data elements from any dimension in a database, and cluster them as one customized group. For example, enterprises want to segment their customers by many different attributes, such as age, gender or income, so that specific products can be promoted to the appropriate customer segments.
Conditional Analysis	Attribute comparisons, value and percentage ranking	Conditional Analyses permit users to analyze data through a set of "What If" scenarios. For example, users can compare ship and order dates to calculate order fulfillment performance.
Affinity Analysis	Correlation, market basket analysis, attribute affinity	Affinity Analyses allow users to discover hidden relationships between business variables. For example, users may discover that certain products sell better with other products during a specific day of the week.
Hypothesis Testing	Normal distribution, probability, confidence intervals, p-Values	Hypothesis Testing allows users to draw conclusions from sample data. For example, the following questions can be investigated: Do certain colors sell better in different cities? Do female customers respond better to direct mail campaigns?
Predictive Analysis	Association rules, clustering, regression, decision trees, neural networks	Predictive Analysis allows users to discover key business drivers, and analyze their trends for proactive decision making.

Figure 5-20 The MicroStrategy BI platform supports a broad range of analytical applications.

5.8 IN-MEMORY INTELLIGENT CUBES

Intelligent Cubes provide a seamless architectural extension of MicroStrategy implementing an In-Memory multi-dimensional caching layer. Because Intelligent Cubes reside in memory while their definitions reside in the global virtual cube of MicroStrategy metadata, this In-memory ROLAP delivers the best of both architectures. In-memory ROLAP instantiates parts of the global multi-dimensional metadata model in memory in form of Intelligent Cubes with actual data of selected portions of the data sources. These Intelligent Cubes are seamlessly interwoven in the ROLAP architecture providing high performance, analytical insight, and economic savings to a modern BI system.

High Performance

Successful businesses depend on swift analysis for optimal business decisions. Maintaining fast query performance and analytical interactivity is a dominant challenge for BI applications, especially those serving thousands of users. Intelligent Cubes deliver speed-of-thought analysis and lightning performance through an adaptive caching technology called In-memory ROLAP. MicroStrategy In-memory ROLAP has demonstrated query performance improvements up to 50x faster – giving businesses information as quickly as they need it.

Analytical Insight

Reports are sometimes a starting point to begin answering business questions. Having a comprehensive set of data for analysis should equate to better decision making. Unfortunately, business users often feel bombarded with incomprehensible data. Intelligent Cubes provide the analytical tools necessary to interactively sift through data to understand the drivers of business performance. Intelligent Cubes help

business users focus their analysis by applying business conditions to filter the data to return only the information required. Intelligent Cubes also let business users create new business calculations. Many frequently used calculations such as variances, ranks, and time comparisons are only one click away. Intelligent Cubes offer on-the-fly access to a complete range of more than 270 arithmetical, statistical, and financial functions for use in new metrics and analysis. Furthermore, business users can create new analytical groups, called Derived Elements, which consolidate similar business attributes together to identify important business characteristics not considered by the original report designer. In short, Intelligent Cubes empower a new level of analytical insight and self-service for the business person.

Economic Savings

Intelligent Cubes provide economic savings in two ways. First, increasing self-service for business users means IT can create and maintain fewer reports. Second, MicroStrategy OLAP Services does not incur the immense IT burden of cube farms found in traditional multidimensional OLAP (MOLAP) products. In contrast, Intelligence Server implements its multidimensional functionality as a virtual ROLAP cube, called an Intelligent Cube, which does not suffer from the problems of traditional cubes. Intelligent Cubes require little to no maintenance, and are a seamless in-memory extension of the MicroStrategy ROLAP architecture.

ROLAP Complementing Intelligent Cubes

Most BI vendors use custom-made proprietary cube databases that are complicated to build but only hold relatively small data volumes for cube analysis. As the demand for more BI applications, more users and more sophisticated analysis grows, hundreds, even thousands, of overlapping cubes are needed to cover all combinations of data subsets, summarization levels, and security profiles for different users across multiple applications. These ever-growing collections of cubes, or “cube farms”, create an immense burden on IT departments that generate the cubes, pre-calculate the summarizations, distribute them to users, and retire them when their data become outdated.

By contrast, MicroStrategy uses a ROLAP architecture that models the structure of the data sources as relational multi-dimensional structures inside the metadata repository. This approach provides the same standard OLAP functionality found in cube-based solutions – cross-tab, page-by, pivot, sort, filter, and drill – but the entire breadth and depth of the data warehouse is available for analysis with no complex steps needed by IT. No further extraction of data into a limited and proprietary structure is needed. The previous tradeoff for a ROLAP-only approach (somewhat slower response-times due to the vast amount of data that can be processed) is eliminated with Intelligent Cubes.

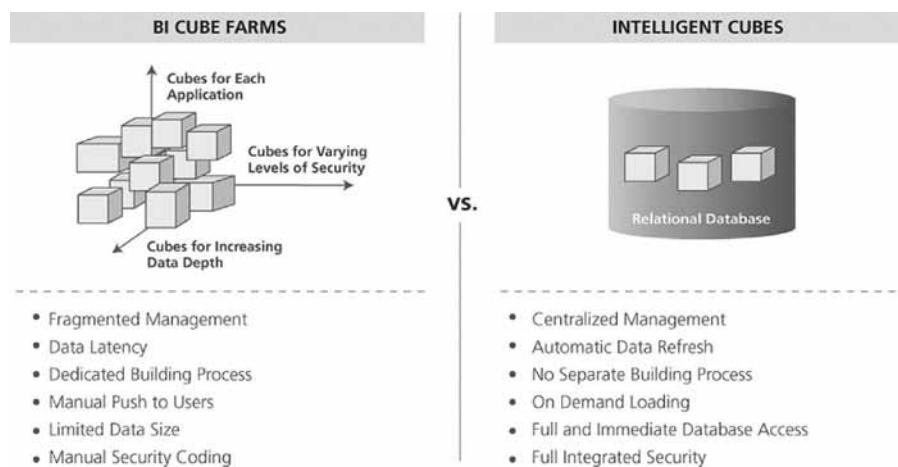


Figure 5-21 Intelligent Cubes enhance performance while providing break-through access to data with high performance.

No Separate Cube Creation Process

IT administrators of other BI software products spend most of their time managing and updating cubes in various locations, such as user desktops, Web servers, and file servers. Cube data is duplicated across all these locations, and this quickly becomes impossible to manage and keep synchronized with the latest data. MicroStrategy OLAP Services delivers fast OLAP by automatically creating and populating Intelligent Cubes in report cache files for use across the entire enterprise.

Intelligent Cubes are designed by the same end-user report creation process. Designers place related business definitions together, insert any required analytics, and scope out the dataset using filtering. New Intelligent Cubes are either manually created by simply executing the source report or automatically when the report is scheduled. Once created, the Intelligence Cubes are loaded into memory automatically when needed and discarded when not.

Intelligent Cube definitions are stored in the centralized metadata, and are automatically created and removed – with cube data always up-to-date and synchronized – without an administrator's involvement. Given that Intelligent Cubes are part of the entire cache refresh system, refreshing Intelligent Cubes can made an automatic part of the data loading process. Updating Intelligent Cubes is seamless, injecting flexibility, simplicity, and speed into the MicroStrategy BI platform, and leading to a high performance analysis environment.

Speed-of-Thought Analysis

Intelligent Cubes access data stored directly in memory in Intelligence Server. Full analytical reporting is performed without accessing the data warehouse with every request. This provides a dramatic boost in performance, as the responses to data manipulation occur almost immediately.

Even though the dataset within the cached subset is bounded by the report definition, users retain access to the entire data warehouse when needed. Unlike other BI products, MicroStrategy Intelligence Server's OLAP Services automatically detects queries outside of the cached subset, and generates a highly optimized query against the specific data source to retrieve the additional data. This process works behind the scenes, and is seamless to the end-user. From the end-user's perspective, accessing data from an Intelligent Cube, from a report cache or reaching through to the data source is transparent. With Intelligent Cubes from OLAP Services, the MicroStrategy BI platform embeds speed-of-thought analysis.

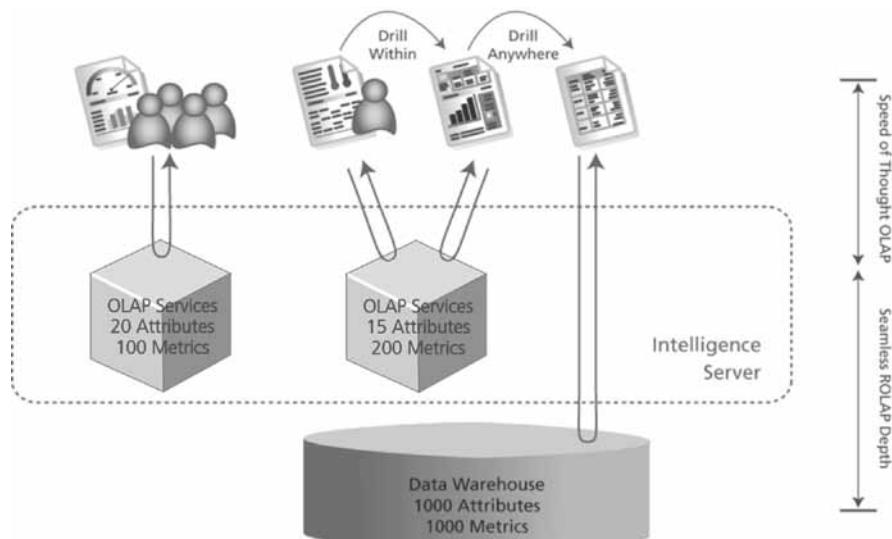


Figure 5-22 Drilling within and outside of Intelligent Cubes is seamless, embedding speed-of-thought analysis.

The following report manipulations are extremely fast because they are executed directly against Intelligent Cubes, without generating additional SQL queries:

- Drilling
- Pivoting
- Page-by
- Sorting
- Subtotals
- Adding derived metrics and calculations
- Grouping several data set elements
- Modifying column labels (column aliasing)
- Adding/removing information from the report view
- Filtering report cache results
- Formatting the report view
- Outline mode
- Banding
- Reset data to original report

Dynamic Sourcing

Intelligent Cubes can serve as a transparent multi-dimensional cache for any query thanks to a feature called Dynamic Sourcing. With this technology the Intelligence Server automatically accesses In-Memory Cubes whenever relevant without any user intervention directing queries from any data source to be served from an Intelligent Cube instead. Intelligence Server analyzes the structure of the report as well as its filtering criteria to determine if a query can be directed to an existing Intelligent Cube. The result is a dramatic performance improvement for user created reports and ad-hoc queries.

Dynamic Sourcing provides a dramatic shift in user analysis. Without bounds to a limited aggregate of data, users on the MicroStrategy BI platform are free to navigate the entire data warehouse for comprehensive analysis. IT departments no longer have to build different cubes to serve different business requirements. Instead, most frequently requested data is bundled once, and users self serve ad hoc requests. Administrators can leverage powerful tools to determine the set of cubes that will optimally support user requests. MicroStrategy Cube Advisor in conjunction with historical usage data from MicroStrategy Enterprise Manager analyzes the structure of the most frequently used reports to suggest a list of optimal cubes to build. Administrators can even use the tool to create the new cube definitions after a thorough what-if analysis showing the expected benefit for each suggested cube.

5.9 MULTI-LEVEL SHARED REPORT CACHING

Caching is the process of temporarily storing data in the BI platform in memory or on disk, closer to users, to reduce the processing time of job requests. Rather than going to the database for each query, a BI system can leverage cached data of previous job requests for subsequent requests. These job requests include browsing metadata, providing prompt answers, and running or creating reports. As the number of business intelligence users increases, the chances that two users will request the same information increases. As more users enter the system and create more caches, average system response time decreases.

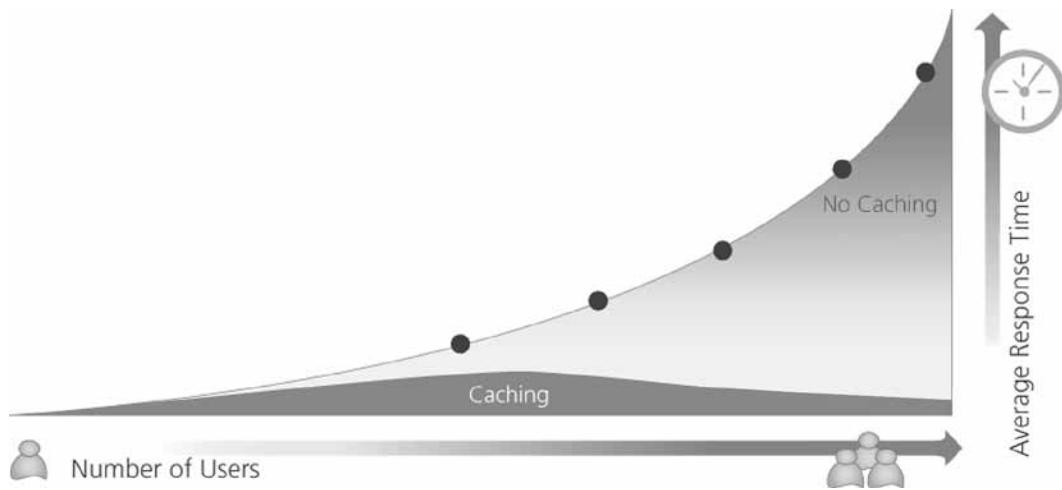


Figure 5-23 Caching dramatically reduces average response time.

While other BI products may simply provide report caching, the MicroStrategy BI platform implements multi-level shared caching. At each strategic point in a user interaction, key data is cached to boost performance. The result is a significantly improved end-user experience and best-of-breed BI performance. The types of MicroStrategy BI platform caching are:

- Metadata object caching
- Attribute element caching
- Database connection caching
- Result caching
- Document caching
- Dashboard caching
- Mobile App caching

In addition, within a clustered Intelligence Server configuration, cache availability status is synchronized across clustered nodes without massive data transfer between nodes, and users connected to one Intelligence Server node benefit from the caches created and stored by other Intelligence Server nodes.

Dashboard, Document, and Report Cache Creation and Delivery

The most important requirement for caching is to serve the correct data. As a part of the MicroStrategy BI platform's integrated security, Intelligence Server ensures that only the right cache is served to the right users. MicroStrategy uses a sophisticated cache-matching algorithm that matches not only the components of caches or Intelligent Cubes, but also the versions of those components and the user's security context. The following checks are carried out to identify a valid cache:

- Does a cache for the requested report or document exist?
- Does the cache contain the requested data: for example, answers to prompts used in the report?
- Does the cache satisfy the users' security constraints?
- Has any part of the report (document) definition been modified since the cache was created?

- Does the cache contain data in the correct language?
- Was the cache data collected with the right security credentials?
- Is a cache file available with the required formatting?

Upon receiving a report request, Intelligence Server verifies that the user is sufficiently authorized through privileges, permissions, and the access control list of the objects requested. If caching is enabled for the report request, Intelligence Server proceeds to verify if there is a valid cache for the request using a well-tuned hash table. The use of a hash table compresses all subsequent verification steps into a single fast and accurate step.

The details of the logical steps are shown in the following diagram. Only after a number of verification checkpoints, including verifying the security context, are cached data returned to the user. The security context includes the user's security filter, user's database login, and the user's connection mapping. If a cache is not available, Intelligence Server automatically generates a highly optimized query against the data source on behalf of the user. This implementation dramatically speeds up cache matching and performance.

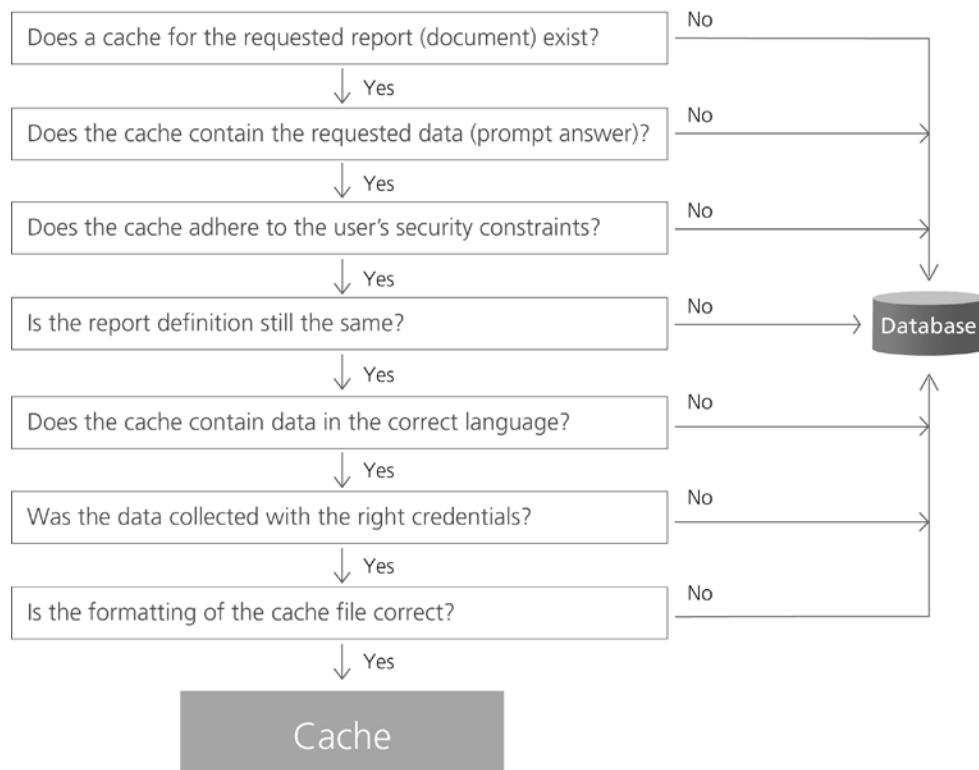


Figure 5-24 Intelligence Server cache matching algorithm ensures correct caches are used.

Cache Management is Automatic and Transparent

MicroStrategy Intelligence Server automatically optimizes and maintains all caches. The Cache Management Engine in Intelligence Server automatically tunes in-memory cached data and controls the amount of memory resources used for caches. When this exceeds the configurable value, caches are off loaded onto disk using a Least Recently Used policy. In this manner, the most number of most frequently used caches are maintained in memory. Compared to on-disk cached data, in-memory cached data provides a much higher degree of performance boost, and Intelligence Server ensures that the most popular cached data remain in memory for maximum benefit.

Caches are automatically updated whenever any underlying object of a particular cache is changed. For example, if the definition of a metric is altered, the new definition is propagated to all reports and other objects using that metric, and all caches affected by the change are automatically invalidated. These caches are automatically recreated the next time the reports or prompts are run.

When Intelligence Servers are clustered together, every cache created by any node in the cluster is available for all clustered machines. The cache hash lookup table on all the nodes is automatically synchronized, making every cache available to users on all nodes.

Intelligence Server statistics provide information on the most frequently used reports, reports with the largest result sizes and reports with the longest query execution times. Caches for these reports can be created during off-peak hours using Intelligence Server scheduling. Administrators can schedule report cache creation using time or event triggers. For example, whenever new data is loaded into the enterprise data warehouse, an event can be triggered for Intelligence Server to update related caches.

As the number of reports in a BI application grows, ensuring that the latest data is cached and used becomes more difficult. Some BI products simply purge all cached data, and rebuild all caches from scratch when more data is added to the data warehouse. MicroStrategy Intelligence Server provides a means to integrate with different data loads, especially if they update different tables. Only cache files that use these tables are purged and recreated, while the remaining caches are left intact. This maximizes the utilization of existing caches by minimizing redundant cache updates.

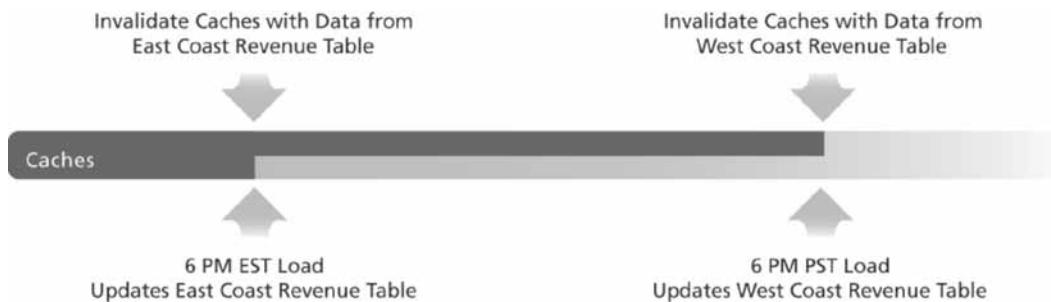


Figure 5-25 Table-based cache invalidation synchronizes caches with data sources.

5.10 SCHEDULING REPORTS AND ADMINISTRATIVE TASKS

Scheduling is the process in which tasks are automatically executed at a specific time or after an event occurs. The MicroStrategy BI platform is designed for both scenarios. This integrates the MicroStrategy BI platform into existing business processes and automates tasks within Intelligence Server. While other BI products only provide for the scheduling of report executions, Intelligence Server also enables the scheduling of administrative tasks. The following tasks are available for scheduling within Intelligence Server:

- Dashboard and Report Cache Management
- Intelligent Cube Management
- History List Message Management
- Project Status Management

Frequently used dashboards and reports can be scheduled to automatically execute after each new data load or at specified times. As part of a regularly timed data load, the Intelligence Server automatically executes reports at specified times. In scenarios where the data load process is less predictable, the data load process can fire off an event which automatically triggers report executions in Intelligence Server. Similarly, various

administrative tasks can be scheduled. For example, when business usage cycles are demarcated by time or by event, Intelligence Server can automatically load or unload projects, invalidate caches, or delete user History List messages.

Distribution Services Option

The Distribution Services Option of Intelligence Server provides high-volume, automated distribution of reports, dashboards, and business performance alerts. Extending the standard capabilities of Intelligence Server, users can manage the distribution of reports and documents to E-mail, file, and printer. Distribution Services monitors your business data like a personalized radar and alerts users to critical business exceptions and trends for faster, fact-based decisions. Intelligence Server sends E-mail notifications to administrators for failed deliveries.

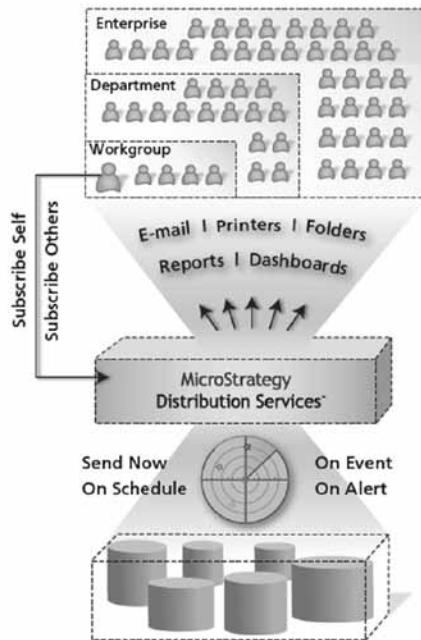


Figure 5-26 Distribution Services monitors your business data like a personalized radar and alerts users to critical business exceptions and trends for faster, fact-based decisions.

5.11 EXPORTING DATA

While other BI products provide limited options to export data, the MicroStrategy BI platform offers numerous options. Within the MicroStrategy BI platform, users can export their report results into various formats:

- HTML
- Microsoft Excel (No formatting, with formatting, or CSV files)
- PDF file
- Adobe Flash
- Microsoft Word
- Text file (Comma, tab, tilde, colon, semi-colon, space, or custom delimited)

These different formats are useful for different purposes. Adobe Flash dashboards are interactive mini-applications that can be run from a browser or sent out via E-mail. PDF documents are used for presentation-quality summaries or briefing books suitable for executive review. Excel documents are useful for offline data

analyses, and are also strongly preferred by finance departments who are skilled at using Microsoft Excel. Using sophisticated analysis within the MicroStrategy BI platform, users can build data subsets for users in other systems. Delimited data formats or datamarts are often used to handle large data sets such as address lists where a customized subset of the data is needed for use in third-party applications, such as a campaign management tool.

When exporting data into Microsoft Excel workbooks, reports can be associated with the MicroStrategy BI platform using MicroStrategy Office; when underlying data is updated, the report results in the Microsoft Excel workbooks are also updated. In addition, with graph reports, grid data can be exported in addition to the graph representation. This functionality is called “Live Charts”. By the inclusion of the raw grid data within Microsoft Excel, a dynamic workbook is generated, giving users opportunities to further analyze the data while offline.

Datamart and Bulk Export

With access to the entire data warehouse along with sophisticated analysis from the MicroStrategy BI platform, users often use analysis results in other applications within the enterprise. When result sets are large, writing the results back into database tables as datamarts provides a convenient path to making the data accessible to different applications. To optimize this process, the Intelligence Server utilizes parameterized queries, and provides a list of user configurable options. This flexibility allows for maximum customization to any specific database environment, ensuring maximum performance. As datamart reports can also be scheduled just like any other report, this workflow can be triggered along with data loads.

```
CREATE [Table Qualifier] TABLE [Table Descriptor][Table Prefix] <table_name>
[Table Option] ( <column_expressions> ) [Table Space][Create Post String]
```

Figure 5-27 Customizable parameters within [] brackets provide maximum flexibility and performance.

A datamart report stores its resulting data set in a pre-defined database table. Similarly, a bulk export report streams its result set into a CSV formatted file in a pre-defined network location.

5.12 FORMATTING DATA

Once all report data processing is complete, the Intelligence Server formats the data using a highly sophisticated formatting engine. The formatting is applied both at the data cell level in a grid, and as a variety of charts and graphical visualizations.

Data Cell Formatting

Each cell within the result data is formatted according to specifications given by the report designer. The following are available cell formatting options within the MicroStrategy BI platform:

- Number Format – General, Fixed, Currency, Date, Time, Percent, Fraction, Scientific, Special, Zip Code, Telephone, Custom
- Alignment – Text Alignment: Vertical or Horizontal, Wrap Control
- Font – Type, Language Script (Character Sets), Size, Bold, Italic, Strikeout, Underline, Color
- Border – Line Style, Location: Top, Bottom, Left, Right, Color
- Background – Fill Color, Pattern Color, Pattern

As each cell represents data at an intersection of business definitions, the formatting engine combines information from these business definitions to create the final report presentation layout. The final format of a single cell is built from five different formatting layers, with the lowest level overriding higher levels. The levels are listed below, with the highest level listed first:

- Pre-defined formats in Autostyles
- Metric Formatting
- Report Banding Formatting
- Custom Group Element / Consolidation Element Formatting
- Report Thresholds Formatting

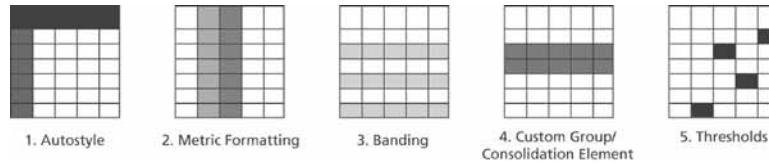


Figure 5-28 Five layers of formatting definition in MicroStrategy grids

An AutoStyle is a set of pre-defined format styles provided by the MicroStrategy BI platform. It consists of a collection of formatting definitions for different sections in a report. These include row-level headers, row-level values, column-level headers, column-level values, metric headers, metric values, subtotal headers, and subtotal values. Within the MicroStrategy BI platform, users are able to create their own custom-defined Autostyles. The other formatting layers are controlled by formatting information within specific business definitions such as metric, custom groups, consolidation, and reports. These 5 layers of formatting information enable the MicroStrategy BI platform to provide highly sophisticated formatting for final report presentation within MicroStrategy Mobile, Web, and Office using XML or any of the export formats listed in the exporting section above.

Charts and Visualizations

Charts and visualizations help display a large number of data points in a visual context for fast and easy comprehension, making it quick to identify data trends or outliers that do not adhere to the norm. Charts are graphical images of the data with layers of “hotspots” that provide interactivity, such as drilling. Visualizations are Flash renderings of the data and are fully interactive. Intelligence Server supports an ever growing list of charts and visualizations that are available in different delivery formats. All visualizations are available as individual widgets. New widgets can be created using the Visualization SDK⁵.

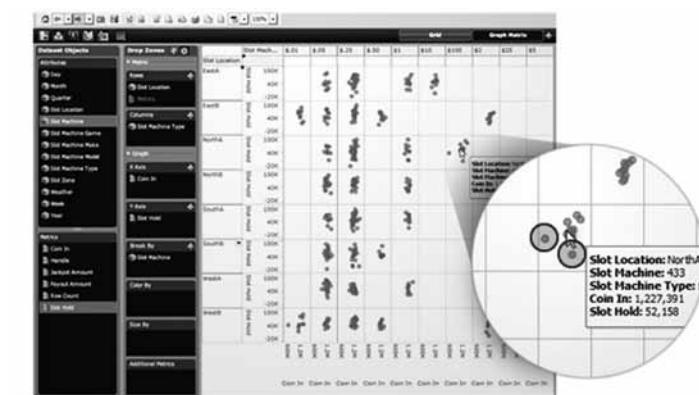


Figure 5-29 Visualizations such as a graph matrix make it easy to compare many charts side-by-side on one page and quickly spot outliers.

⁵See Appendix D for a full list of charts and visualizations

5.13 SIMPLIFIED CENTRAL ADMINISTRATION

Administration of the MicroStrategy BI platform is performed centrally through Intelligence Server. Unlike other BI platforms, the Intelligence Server components and services were built organically. As a result, there is a single unified administration interface within MicroStrategy Desktop, with full-featured, integrated scripting support from MicroStrategy Command Manager. The administration of Intelligence Server is comprised of two main components:

- Monitor the behavior of the system
- Control the resources and usage

Together, these two components provide full oversight into the MicroStrategy BI platform.

Monitoring Gives Instant Insight

MicroStrategy Intelligence Server provides real-time system information in all aspects of the platform. From MicroStrategy Desktop, various information feeds such as the following are available:

- Jobs
- Projects
- Users
- Database Connections
- Scheduled Jobs
- Caches
- Clusters

Each of these monitors provides details on the number of items and a snapshot of their statuses. For example: The Job Monitor below shows 4 jobs currently executing. The user executing the report “Top 10 Profitable Products” is currently providing prompt answers. A document called “Customer Analysis Scorecard” is currently executing with two children reports named “Q3 Top 10 Most Profitable Products” and “Q3 Top 10 Customers by Profit”. All of these reports are executed by the user “Administrator” connected to the project “Customer Analysis Module”. The reports are submitted within a short time of each other and from the same client machine.

Job...	User	Status	Description	Project ID	Project Name	Priority	Creation Time	Network Address
126	Administrator	Waiting for autoprompt	Running report Top 10 Profitable Products	0	Customer Analysis Module	Low	9/6/2005 1:52:49 PM	10.15.200.32.10252
170	Administrator	Waiting for children	Running document Customer Analysis S...	0	Customer Analysis Module	Low	9/6/2005 1:54:04 PM	10.15.200.32.2856
171	Administrator	Executing	Running report Q3 Top 10 Most Profitab...	0	Customer Analysis Module	Low	9/6/2005 1:54:05 PM	10.15.200.32.2856
172	Administrator	Executing	Running report Q3 Top 10 Customers by...	0	Customer Analysis Module	Low	9/6/2005 1:54:05 PM	10.15.200.32.2856

Figure 5-30 The Job Monitor provides real time details on report executions.

Similarly, the Cache Monitor provides real time insight into the Intelligent Cubes and caches available and their states. Below, there are 12 Intelligent Cubes. The cache monitor reflects the projects in which their reports reside, the cache sizes, statuses, and creation times, and details on how many times each cache was used.

Intelligent Cube Report Name	Project Name	Status	Last Update Time	Hit Count	Size (KB)
Affinity	Food and Beverage (iPad)	A, L, F	11/30/2010 1:30:01 PM	0	202526
Private Loan Statistics Cube	Loan Stats	A, L, F	1/23/2012 7:31:20 PM	0	124741
Private Loan Statistics Cube	Loan Stats	A, L, F	3/20/2012 10:55:05 AM	330	98416
ISPs	SpeedTest	A, L, F	11/15/2011 11:11:44 PM	0	55288
ALL-Years-World Bank Facts Cube	iPad Standard	A, L, F	12/20/2010 9:45:14 AM	0	50652
Cube converted from KPIs over time for graphs (Cube)	Banking Performance	A, L, F	1/13/2012 8:01:06 AM	6	31611
Top Drinks About	Food and Beverage (iPad)	A, L, F	11/30/2010 1:33:05 PM	20	30822
Cube converted from Key Financial Metrics (Cube)	Banking Performance	A, L, F	1/13/2012 8:00:47 AM	6	27184
Drinks Summary (Graphs) Cube	Food and Beverage (iPad)	A, L, F	11/30/2010 2:28:02 PM	0	25495
Drinks Summary Cube	Food and Beverage (iPad)	A, L, F	11/30/2010 2:29:08 PM	0	22573
Adherence Cube	Care Management	A, L, F	2/20/2011 3:11:12 AM	0	22159
Cities	SpeedTest	A, L, F	11/15/2011 1:11:22 PM	0	17888

Figure 5-31 The Cache Monitor provides real time details on available caches.

The status of each cache is also listed in the cache monitor. This provides additional insight into how the caches can be used. The following is a list of cache statuses

- Ready: Cache can be used.
- Processing: Cache is being created.
- Invalid: Cache cannot be used because the report definition has changed.
- Expired: Cache cannot be used because it is too old.
- Filed: Cache is written to disk.
- Loaded: Cache is in memory.
- Dirty: Cache in memory is newer than cache on disk.
- Updated: Cache has been updated.

In addition to real time data, the Intelligence Server also logs historical usage statistics, allowing comprehensive analysis of system usage. MicroStrategy Enterprise Manager implements a turn-key BI solution analyzing these statistics providing powerful insights into all parts of a MicroStrategy implementation.

Monitoring with Health Center Improves System Uptime

MicroStrategy achieves the lowest TCO through comprehensive administration products. Health Center provides a centralized console that monitors the entire BI environment, checking for and anticipating potential errors before users are affected. When Health Center detects any problems, it alerts administrators of issues and how to fix the problem via e-mail and in the Health Center console. Health Center streamlines the technical support process by collecting all the diagnostic information, such as log files and configuration settings required to quickly resolve an issue. This diagnostic package provides a comprehensive view into the issue and helps reduce the tech support resolution time up to 50%. The result is a BI system with higher availability that can be run with fewer IT personnel.



Figure 5-32 Health Center improves system uptime by resolving the most common issues and by facilitating diagnostics gathering.

Take Action with Controls for Maximum Efficiency

By default, the Intelligence Server is already preconfigured, however, it is also designed with various levers and controls at numerous levels and components, allowing the BI platform to be configured and optimized for any deployment scenario. Complementing the extensive amount of monitoring information available,

the Intelligence Server provides full control into the business intelligence system to maximize the system's efficiency. Control of the Intelligence Server can be broken down into three main categories:

1. Server Workload

2. Memory

3. Prioritization

Server workload refers to managing the users logged onto the BI platform and the report information they request. Administrators control the following in order to govern the server workload:

- Number of network threads
- Maximum number of executing jobs
- Maximum number of user sessions and per project
- Maximum number of jobs per project, user or session
- Number of database connections
- User idle time
- SQL cancel
- Maximum report execution time
- Maximum recent reports for manipulation
- Project failover latency
- Configuration recovery latency

Memory management is a key requirement for an enterprise class platform. MicroStrategy Intelligence Server governs memory usage on multiple levels to control the total amount of memory consumed by Intelligence Server, as well as to optimize resource utilization by its various components. Governing is performed through the following parameters:

- Report execution size
 - Number of rows returned
 - Number of elements returned
 - Maximum number of XML cells
 - Maximum number of XML drill paths
 - Maximum memory consumption for XML, PDF, and Excel formatted file generation
 - Maximum memory consumption for SQL generation
 - Maximum memory consumption for data fetching
- Maximum RAM usage for object cache
- Maximum RAM usage separately for result cache, document cache, and Intelligent Cubes
- Maximum number of result caches, document caches, and Intelligent Cubes
- Maximum RAM for report cache index
- Maximum RAM usage for element cache

- Maximum RAM usage for Intelligence Server process
- Minimum reserved RAM for external processes
- Minimum machine free RAM
- Maximum Intelligence Server use of virtual address space
- Maximum single allocation size
- Memory request idle time
- Maximum RAM usage for working set
- Cache lookup cleanup frequency
- Maximum number of messages per user
- Maximum message lifetime
- Maximum attribute elements to display
- Maximum file size to import, quota per user

Prioritization of report requests is critical to ensure that users receive the level of service appropriate to their positions. Without this capability, a CEO would need to wait at the end of the queue before his or her requests are processed. To avoid such scenarios, all requests can be prioritized, and routed through Intelligence Server. Job prioritization is based on:

- Report cost
- User groups
- Projects
- Request type
- Application type

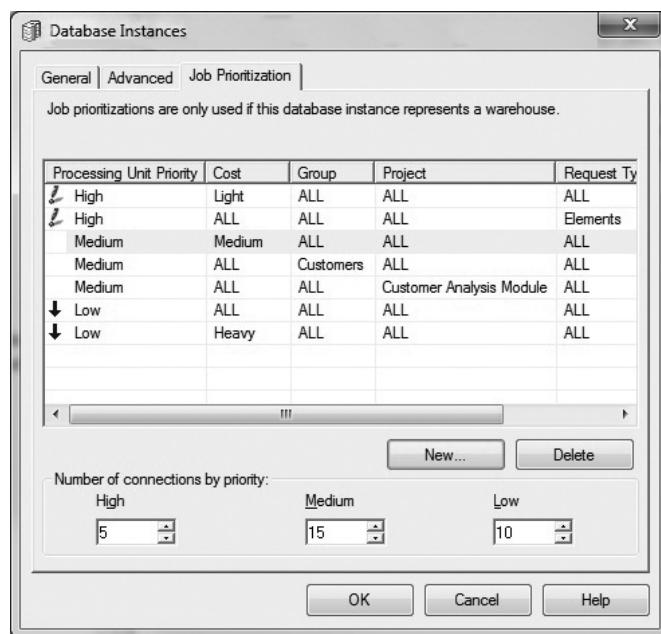


Figure 5-33 Parameterized processing threads provide flexible report request prioritization.

By combining the wealth of information available via monitoring and near real time statistics, together with the capability of scripting control actions, MicroStrategy Intelligence Server provides the foundation for sophisticated system administration.

5.14 SUMMARY

MicroStrategy Intelligence Server provides enterprises with one central location for all corporate querying, reporting, and advanced analytical needs. With optimized query generation for different data sources, multi-level caching, Intelligent Cube architecture, and a fully optimized 64-bit technology, end-users can easily perform the most sophisticated analysis on data reaching through to the entire enterprise data warehouse while experiencing great performance and report interactivity. In addition, Intelligence Server's clustering, caching, monitoring, and control enable enterprises to scale to multiple terabytes of data and thousands of users as well as ensure a 24x7 fault tolerant environment.

MICROSTRATEGY WEB ARCHITECTURE

MicroStrategy Web is the primary business user interface of the MicroStrategy BI platform. MicroStrategy's Web architecture provides a single, consistent interface to all users whether the BI application is departmental and internal, or an extranet application deployed to hundreds of thousands of users. MicroStrategy Web allows business users of any skill level to move fluidly between all styles of BI to satisfy their reporting, analysis, and monitoring needs.

Across all industries, styles of BI and deployment profiles, customers demand a BI Web architecture that is:

- High performing and scalable
- Secure, yet extremely simple to deploy and maintain
- As intuitive and interactive as the best desktop applications
- Easily integrated with existing systems, software and hardware

6.1 CORE DESIGN PRINCIPLES

With over 20 years of experience supporting the most demanding BI applications, MicroStrategy delivers a unified BI platform and Web architecture that is proven to meet these business requirements. MicroStrategy ensures customer success by rigorously adhering to following fundamental design principles:

- Be a part of a Service-Oriented Architecture for BI
- Use open standards in a layered architecture
- Build a platform-independent architecture on a single code base
- Ensure data security for all users
- Provide robust user interactivity with a no compromise, zero footprint interface
- Provide full-featured BI functionality
- Easily integrate with enterprise portals
- Support all available data formats, grids, graphs, and maps

6.2 OPTIMIZE PERFORMANCE WITH A SERVICE-ORIENTED ARCHITECTURE

MicroStrategy Web is the primary user interface of the broader MicroStrategy BI platform. Before discussing the Web architecture in detail, a brief review of how MicroStrategy Web fits into MicroStrategy's multi-tier BI architecture is useful. A Web-based deployment of MicroStrategy's BI platform consists of four primary tiers:

1. Data sources, such as relational databases, ERP system, MDX cubes, flat files, web services or combinations of these.
2. MicroStrategy Intelligence Server, operating on a wide variety of hardware and operating system combinations.

3. MicroStrategy Web, operating on a wide variety of application server, web server, and operating system combinations.
4. A web browser operating on an end user's computer.

MicroStrategy's BI architecture optimizes the distribution of work across all four tiers in order to provide business users with the best possible performance. Jobs are distributed to the tier which can perform them most efficiently, minimizing the hardware and database resources required for the BI deployment.⁶

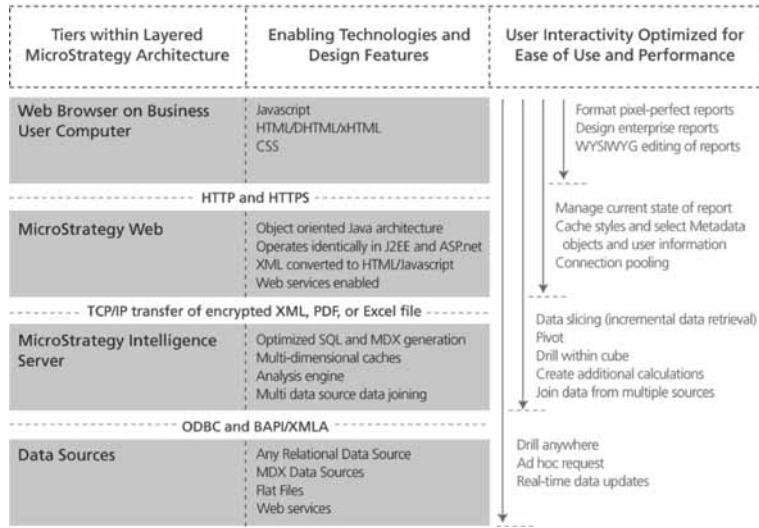


Figure 6-1 MicroStrategy's multi-tier BI architecture optimizes performance by distributing work to the tier which can process it most efficiently.

MicroStrategy Web's Role in the Multi-Tier Architecture

MicroStrategy Web translates end user actions into requests for information from Intelligence Server and presents end users with all the information, formatting, and interactivity provided by Intelligence Server. The following high-level steps describe MicroStrategy Web's role in processing a user request:

1. Business users access MicroStrategy Web through a Web browser via a standard HTTP(S) connection to a Web server. These user actions generate requests that are transmitted back to MicroStrategy Web.
2. MicroStrategy Web translates user actions into XML requests that are transmitted via TCP/IP to Intelligence Server. These requests may be for report results, metadata object definitions, or ad hoc calculations.
3. Intelligence Server retrieves data from cache, or from data warehouses, operational systems, cubes, and the metadata repository, adds analytical richness to the data, and formats it.
4. Intelligence Server returns report results, formatting, security privileges, and other information to MicroStrategy Web in an XML format.
5. MicroStrategy Web translates the XML data into Web pages with all the appropriate information, formatting, layout, and BI functionality.

⁶See Appendix G for a list of operating environments

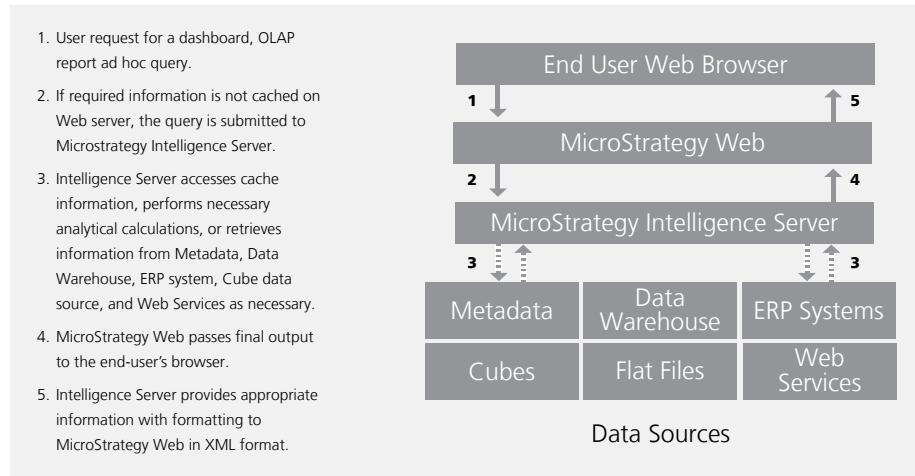


Figure 6-2 MicroStrategy Web handles all communications between end users and MicroStrategy Intelligence Server.

Many requests can be satisfied by MicroStrategy Web without communicating with Intelligence Server, and most user interactions and job requests occur within seconds for hundreds or thousands of users simultaneously.

6.3 OPEN, LAYERED DESIGN OPTIMIZES PERFORMANCE, INTEGRATION, AND CUSTOMIZATION

MicroStrategy Web's internal architecture consists of JavaBeans, Java classes, XML files, a servlet or ASP.NET container, and a thin layer of either ASP.NET pages or JSP pages. Moreover, MicroStrategy Web functionality is exposed from a task framework for third party applications to invoke features without additional coding. With this architecture, MicroStrategy Web operates on virtually any operating system, application server, and Web server combination.

The layered architecture separates the data access logic from the presentation and control logic, providing a more efficient internal workflow. Additionally, interface and application customizations are much simpler, since the cleanly layered architecture isolates modifications to a single layer of the architecture.

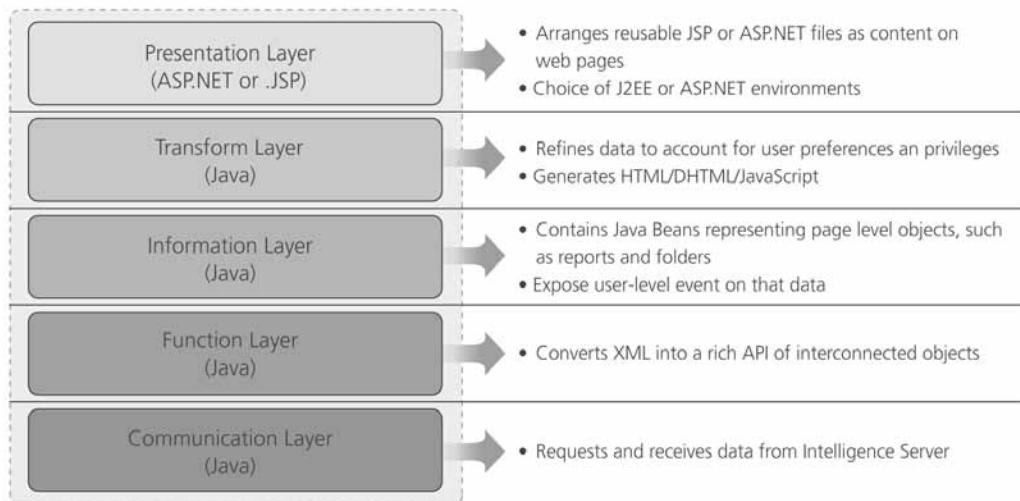


Figure 6-3 The MicroStrategy Web architecture provides distinct functional layers for more efficient processing of requests, and easy customization and integration.

The Communication Layer Interacts with Intelligence Server

The lowest layer of the Web architecture is the Communication Layer that manages the communication between MicroStrategy Intelligence Server and MicroStrategy Web. All messages are passed using an efficient XML structure that maximizes the throughput of Web requests.

The Function Layer Translates Requests into XML

The Function Layer sits above the Communication Layer and performs two primary jobs:

- Converts the XML returned by Intelligence Server into interconnected objects for use by other layers.
- Translates object interactions (for operations such as pivot, page-by, prompt, drill, and search) into requests for data from Intelligence Server submitted through the Communication Layer.

The Information Layer Instantiates All Web Data and Functionality

The Information Layer gathers the data and available functionality for a report, and forms self-contained MicroStrategy Java Beans, called “Web beans”. A Web bean for a report provides the report data as well as the methods and properties necessary to pivot report grids, sort the data, and expose other report interactions. In the context of a Model-View-Controller design paradigm, a Web bean is a miniature, combined Model and Controller object. It is a Model object in that it represents a high-level data concept (such as a report). It also fulfills the role of a Controller by coordinating the use of several Web objects to fulfill the collection of XML, and by managing the transform objects which are the View objects.

The Transform Layer Converts Web Bean Information into Output Formats

The Transform Layer combines BI information contained in the Web beans with transform styles to create report output formats. This layer also manages user preferences, and enforces user security privileges on the Web server.

The Presentation Layer Constructs the Final Web Page

The Presentation Layer defines the final Web page layout that is presented to the end user. It organizes the page sections such as header, page-by, grid, left panel, right panel, and footer, and identifies where the data from the Web bean should be rendered in those sections. A single Controller, used by either an ASP.NET or Java servlet, controls overall page execution, and manages all interactions between the presentation layer and the objects in the underlying layers.

6.4 MICROSTRATEGY WEB IS BUILT FOR INTEGRATION AND CUSTOMIZATION

Changes to the way MicroStrategy Web presents reports and BI functionality occur in external XML configuration files. No ASP.NET, JSP or Java coding is required for most customizations. The configuration files also provide a ready catalog of all changes made to the default configuration of MicroStrategy Web, resulting in customizations that are easy to maintain, extend, upgrade, or roll back to their default configuration. It is important to note that the configuration files do not need to be modified at all to fully deploy MicroStrategy Web.

MicroStrategy Web report results, functionality, formatting, and user interactivity are easily integrated in any commercial or homegrown portal and with other IT applications. MicroStrategy's layered Java architecture is compatible with industry standards such as:

- J2EE
- ASP.NET
- XML
- xHTML
- JSR 168
- WSRP
- SOAP

MicroStrategy Web Functionality Wrapped into a Task-based Framework

In MicroStrategy Web, specific features and functionality are available as a service through the architecture's task-based framework. Each specific service is called a task and, once defined, can be used from many different applications regardless of the how this task is accessed. Once MicroStrategy functionality is encapsulated a task, an HTTP request, a Web Service, COM, or .NET frameworks or any other application can use it without any extensive coding.

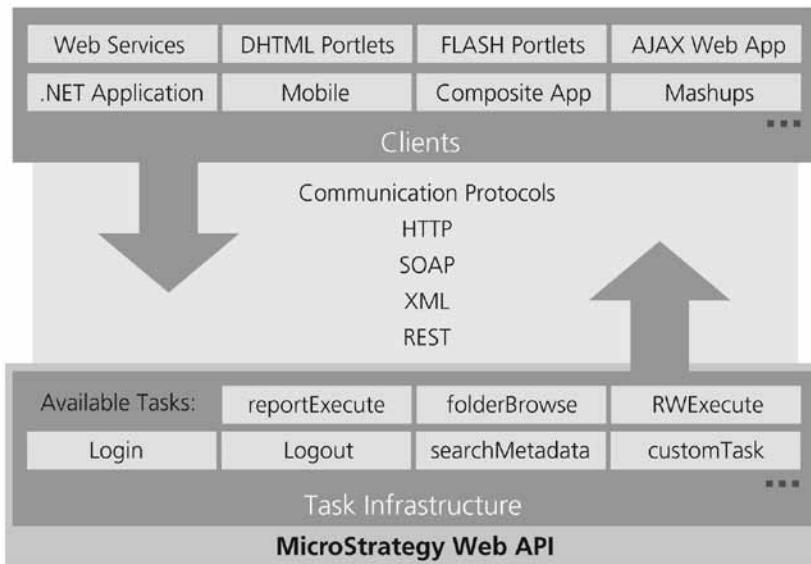


Figure 6-4 The task Infrastructure facilitates MicroStrategy Web functionality to be accessible from any client or protocol.

Style and Layout Parameter Files Determine Web Page Content

Modifications to Web page content are stored in the Style Catalog and Layout files using an xHTML format to embed MicroStrategy-specific tags into a familiar HTML structure. Only simple changes to file parameters are needed to perform common formatting and layout customizations such as rearranging or removing menus, toolbars, grids, and other GUI elements that compose the report page. An Eclipse plug-in simplifies this process significantly.

The Page Configuration File Catalogs All Web Pages

MicroStrategy Web contains Web pages specifically tailored to the activity performed on each page. Examples include the Login Page, Report Execution Page, Shared Reports Page, My Reports Page, Create Report Page and Create Document Page. The Page Configuration XML file catalogs all the pages available to the Web application in a format that is easy to understand and manage. Each JSP or ASP.NET file represents an independent, pluggable component that is not dependent on other JSP/ASP.NET files. The Page Configuration file externalizes the logic for combining these files into complete Web pages linked together for navigation.

Web Services Allow Loosely-Coupled Integration with Other Applications

The MicroStrategy SDK includes a Web Services Development Kit, which is composed of sample code and documentation describing MicroStrategy's Web services API's. The Web Services Development Kit is used as a starting point to deploy MicroStrategy reports as Web services for integration with other Web services applications. It utilizes the object-oriented layered MicroStrategy Web architecture and Apache SOAP Toolkit.

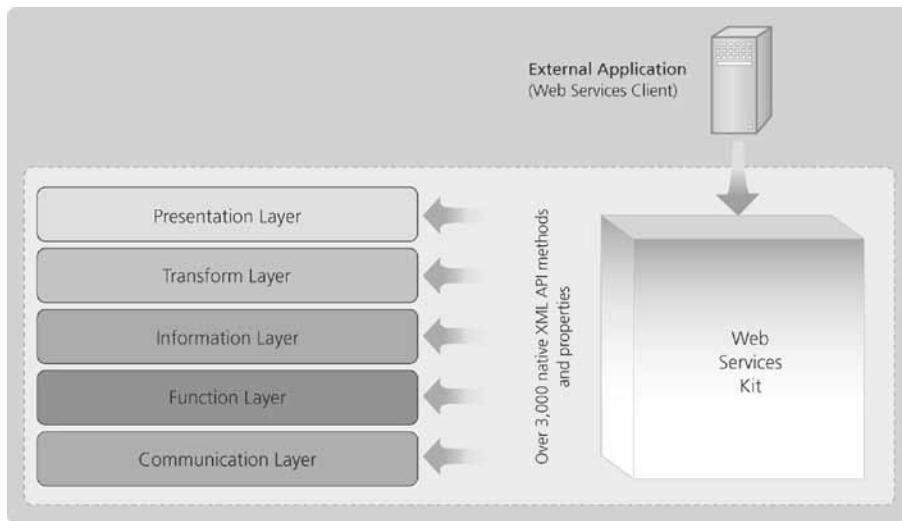


Figure 6-5 With the Web Services Development Kit, application developers use Web Services to integrate MicroStrategy BI applications with external applications or systems.

6.5 A SINGLE, PLATFORM-INDEPENDENT CODE BASE PROVIDES FLEXIBILITY AND CONSISTENCY

MicroStrategy Web uses a single code base that is truly platform independent and ensures that updates and upgrades occur on all supported environments simultaneously. The exact same reports, functionality, and user experience is delivered regardless of the choice of application server, operating system, hardware, or ASP/JSP front-end. MicroStrategy Web supports all leading of application servers, Web servers, and operating systems.

Increases Flexibility and Reduces Future Risk

Many IT environments contain a variety of hardware and software, and require the flexibility to mix and match hardware and software across applications. Some organizations even choose to operate one hardware platform, operating system, and application server combination in their development and testing environments, and a different combination for their production systems. MicroStrategy's Web architecture enables this flexibility today.

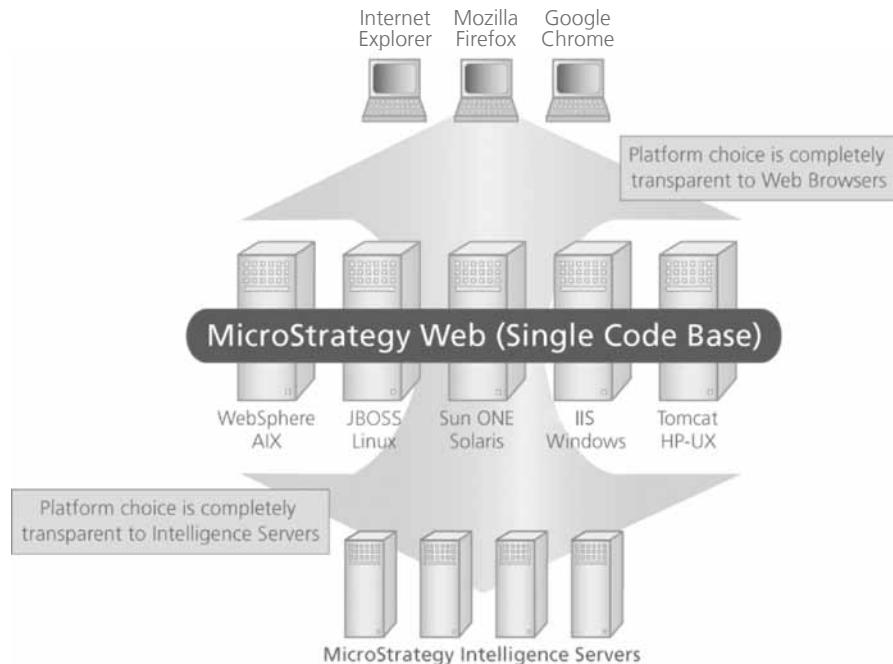


Figure 6-6 MicroStrategy Web's platform-independent, single code base provides maximum deployment flexibility.

Fast Response to Customer Issues and Enhancement Requests

For 10 years, MicroStrategy's Web architecture has consisted of a single unified code base across all supported operating systems, application servers, and Web servers. Other BI tools have different Web architectures for different styles of BI applications. These tools may have one Web interface and architecture for dashboards, and another interface and architecture for OLAP analysis. Compounding this problem, each interface is often compiled from separate code bases in order to run on different operating platforms. The same enhancements and issue fixes must be programmed many times into several different code bases. As a result, some fixes or enhancements appear in different interfaces or on different operating systems at different times, if at all.

6.6 ALL DATA IN THE BI APPLICATION IS SECURE

MicroStrategy addresses security requirements with an elegant architectural design that is optimized for high performance and high scalability while providing information security. Eight primary design principles ensure the integrity of the MicroStrategy Web architecture:

1. Secure communications across firewalls

The MicroStrategy platform is normally installed on more than one server to distribute the BI workload. Secure communication across these servers is often governed by layers of firewalls constructed into Demilitarized Zones (DMZ). With three firewalls, two distinct DMZ's are created with one DMZ protecting the Web server and the second DMZ securing the infrastructure of the data source and Intelligence Server.

2. No database connection from the Web Server

Web servers are the most common targets for hackers. It is vital that there is no direct access to the data warehouse from the Web server. MicroStrategy's BI architecture avoids this potential security problem by routing all communications with databases or other data sources through Intelligence Server. In this configuration, a hacker who gains access to the Web server does not also gain access to the data sources.

3. Single port control for data access
Firewalls protect corporate information assets by limiting applications to certain computer network ports. The ports used by MicroStrategy Web and Intelligence Server for inter-server communication and for data access are fully configurable. As a result, intrusive port scans can be blocked by firewalls while letting users access critical business information.
4. No external Remote Procedure Calls (RPC) or Remote Method Invocation (RMI) calls
RPC and RMI calls are hazardous because they allow hackers to access, and control remote and distributed computer processes. More importantly, these calls often allow anonymous access through separate, open ports in the firewall. MicroStrategy Web uses only XML to communicate with Intelligence Server eliminating the need for RPC or RMI calls completely.
5. No reliance on ActiveX, Java applets or a specific Web Browser
Many corporate firewalls prevent incoming ActiveX or Java applets as a standard security procedure. Unlike other BI products, MicroStrategy's BI platform does not make use of ActiveX, Java applets, downloaded compiled code, or a specific Web browser to deliver complete BI functionality to business users.
6. Secure session handling
MicroStrategy Web prevents the disclosure of the session IDs and sets a secure session cookie, which when deployed over an HTTPS request, protects it from being passed over to unencrypted requests. Additionally, MicroStrategy Web creates a new HTTP session upon a successful login. This ensures that there is no session theft, masquerading, or fixation from MicroStrategy Web server.
7. Cross-site request forgery protection
MicroStrategy Web issues a random token under every POST and GET transaction that is validated to prevent cross-site forgery attacks.
8. Secured against Injection attacks
MicroStrategy Web is equipped to prevent denial of service attacks. With format validation on user input fields and encrypting SQL requests, MicroStrategy Web is designed to prevent unwanted large scale intrusions.

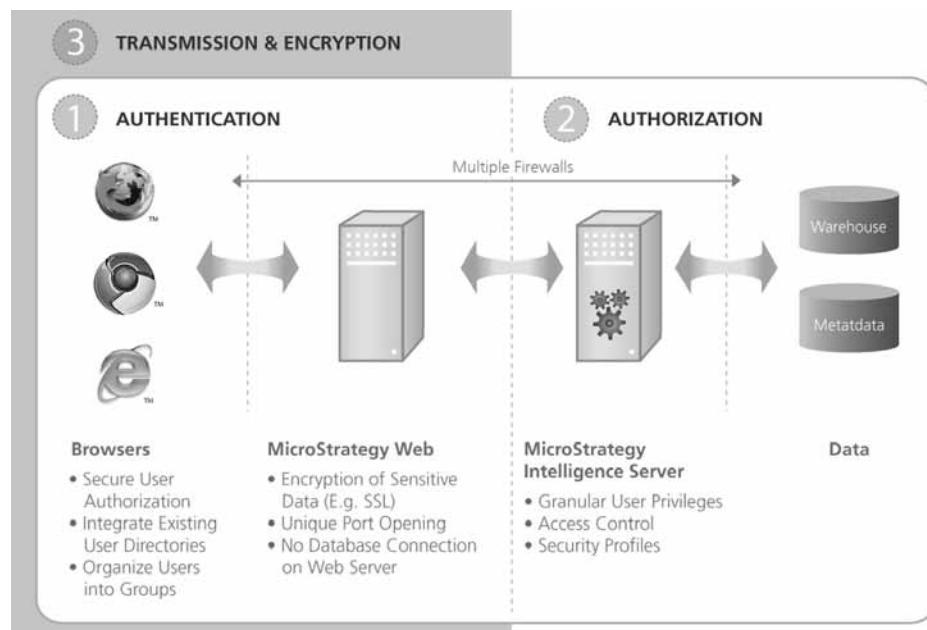


Figure 6-7 The MicroStrategy BI platform supports three layers of security: authentication, authorization, and transmission/encryption.

6.7 EMPLOY LEADING EDGE TECHNOLOGIES TO MAXIMIZE EASE-OF-USE AND INTERACTIVITY

Zero Footprint Interface Maximizes User Adoption

MicroStrategy provides all BI functionality through a zero-footprint interface unlike other BI architectures that offer less functional zero-footprint interfaces alongside full-featured ActiveX or Java applet interfaces. Providing a full functionality, zero-footprint Web interface has been a guiding design principle for MicroStrategy Web since before 2000. MicroStrategy's zero-footprint Web architecture is well designed to satisfy common deployment requirements.

Requirements	MicroStrategy Web's features support these requirements:
Access BI applications using any Web browser.	No dependence on a specific browser. Supports Internet Explorer, Mozilla Firefox, Google Chrome, and other browsers.
Deploy BI applications quickly and safely.	Users only need a URL address, which can be included on a corporate website or internal portal.
Restrict installation of additional software, applets, and plug-ins or ActiveX components on client machines.	No need to install any compiled code, or use plug-ins and ActiveX already embedded in Web browsers.
Minimize impact of Web browser security setting on the usability of the BI application.	User sessions are managed in cookies on the client or without cookies on the application server.
Conduct worry-free upgrades of BI Web server software.	MicroStrategy Web is only installed on the application server or Web server, and is updated to the latest software version within minutes.
Provide an interface compatible with the US Government's Section 508 accessibility requirements.	An HTML-only view of the interface is compatible with the US Government's Section 508, and designed to work with accessibility technologies such as screen readers.
Deliver Superior Performance.	Express mode available in MicroStrategy Web was a radical re-write of MicroStrategy Web using JSON technology.

Figure 6-8 MicroStrategy Web's zero footprint interface satisfies all user and administrative requirements.

MicroStrategy provides a single Web interface for all business reporting, analysis and monitoring applications. The same Web URL accesses all reports, dashboards, scorecards, analyses, and ad hoc queries. Navigation in the interface, running reports, and interacting with the results is identical for all styles of BI; users do not need to learn how to use different interfaces.

Groundbreaking Interactivity and Ease of Use

In 2002, MicroStrategy pioneered the use of XML, DHTML, JavaScript, and asynchronous communication in a zero-footprint Web interface that looked and acted like a traditional desktop application. In 2005, the term "AJAX" (Asynchronous JavaScript and XML) was coined and popularized as a description for a category of applications that extensively use XML, DHTML, JavaScript, and asynchronous data loading and formatting for creating interactive Web applications. Although the term, "AJAX", is relatively new, its component technologies and characteristics evolved over multiple years into a general paradigm for Web development. For many years MicroStrategy has pioneered innovative uses of AJAX technologies such as DHTML and JavaScript in order to provide business users with the most intuitive and interactive BI experience on the Web. In 2008, Flash Mode was added so that business users render their data in visualizations to quickly derive insights.

Superior Performance

In 2010, MicroStrategy introduced Express mode to provide users with superior performance using JSON (JavaScript Object Notification) technology. With this approach, MicroStrategy Web content is bundled

into light-weight JSON coded objects, and the entire HTML is rendered on the client browser. This radical approach drastically reduces the amount of content sent from the web server to the client delivering high performance while retaining regular BI functionality such as sorting, drilling, pivoting, and page-by.

Designed for the Business User with “Windows-on-the-Web” Simplicity

Business users of all skill levels find MicroStrategy Web intuitive and simple to use as they recognize familiar tools, techniques, and formats similar to any desktop application. With no prior training, MicroStrategy Web users easily access vital reports, perform insightful analysis, and monitor key business metrics using familiar techniques and Windows-like paradigms such as:

- Drag and drop actions
- Right click menus
- Drop down menus
- One-click functionality in toolbars
- Undo/redo buttons

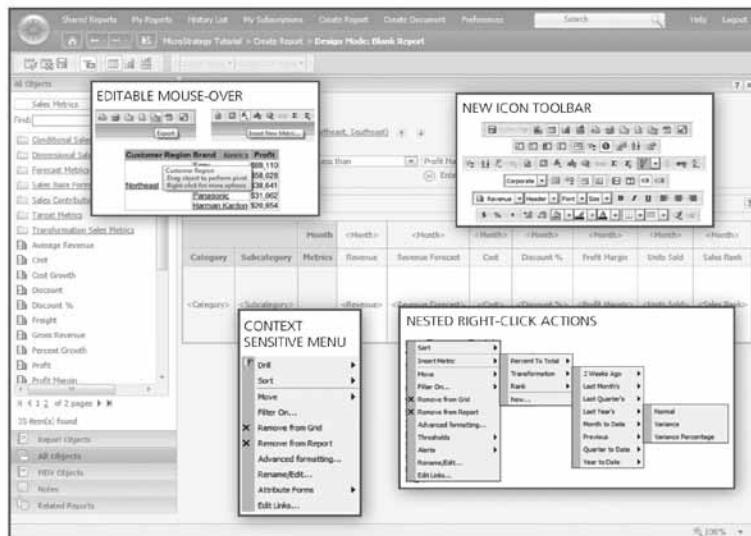


Figure 6-9 The Web interface is easy to learn with toolbars, tooltips, menus, and context-sensitive right-click actions.

Familiar Web Navigation Techniques

With MicroStrategy Web, all reports are organized into folders and subfolders with a navigation bar clearly indicating the location within the folder hierarchy. Reports and other objects can be found quickly using a familiar “Search” box. A single click of the mouse accomplishes common tasks like running a report, exporting a report, or drilling to additional detail. New reports can be saved in existing or new shared folders, or in private folders. Favorite reports can be bookmarked in a user’s personalized “My Reports” area.

Floating Dialogs Instead of Intrusive Panels

Many BI tools provide input dialogs using panels that occupy valuable on-screen real-estate, and partially obscure the report. These panels cannot be moved, and restrict the business user’s ability to interact with a

report while still viewing the full context of the report. In contrast, for user input, MicroStrategy Web utilizes floating dialogs that are formatted to have a similar appearance to standard Windows dialogs. These dialogs can be moved anywhere on the screen, ensuring that business users never have to sacrifice important report context for interactivity.

Toolbars Provide Powerful One-Click Interactivity

The vast majority of BI interactions are available as toolbar buttons in the Web interface. The single-click actions are described by tooltips, which are displayed when the mouse pointer hovers over the toolbar button. The buttons are grouped into a series of toolbars, which can be hidden or shown with a simple mouse click. Only users with the corresponding security privileges can use a given toolbar button.

Context-Sensitive Menus

A menu shows all available BI actions for a piece of data, an object, or an area of a report by simply right mouse-clicking on the data, object or area. MicroStrategy Web only displays the options that are permitted by the user's security profile, and that are appropriate in the current context. For example, when right-clicking on a report or folder name, common file-management tasks such as copy, move, rename, or create a shortcut are available. When a business user right-clicks on a scorecard, dashboard or enterprise report, actions such as drilling, renaming, changing displays, copy, paste, and format are presented. This makes MicroStrategy Web as easy to use as an office productivity application.

6.8 SEAMLESS SUPPORT FOR ENTERPRISE PORTALS

MicroStrategy enables organizations to embed BI content into Enterprise portals. Typically, an Enterprise portal consists of different views, each view presenting a certain information, all view hosted on a single page. Portals provide business users a unified view of different, critical pieces of content, each piece of content displayed in a single portal window called "portlet". With portals, organizations can provide their users a 360 degree view of their business ensuring in-time information for accurate decision making.

Recognizing that BI content is an integral part of the business user experience, MicroStrategy provides out of the box port-lets for the following enterprise portals and content management systems,

- IBM Web Sphere Portal
- Microsoft Sharepoint Portal
- Oracle WebLogic Portal
- SAP NetWeaver Portal
- Liferay Portal
- Apache Pluto
- Drupal Content Management System
- DotNetNuke Content Management System

All out-of-the-box portlets are designed to integrate seamlessly with enterprise portals removing the need for IT to write enterprise portal specific code. Each portlet is designed to take advantage of the storage and repository mechanisms of its portal product.

Figure 6-10 MicroStrategy Web content embedded in three portlets, alongside third-party content feeds in a Microsoft SharePoint portal.

Advantage of Using MicroStrategy Portlets

Each portlet offers a wide range of features that are common to the four major enterprise portal servers and offers a large number of benefits:

- BI content: MicroStrategy portlets enable organizations to combine BI content into one or more of the portal windows.
- Portlet to Portlet communication: Out of the box portlets are designed to communicate with each other as well as with non-MicroStrategy portlets. Users can make selections in one portlet that controls content in other portlets.
- Single Sign On: MicroStrategy portlets leverage the authentication mechanisms of the portal, portlet, and MicroStrategy Web to give portal users seamless access to MicroStrategy BI content—providing secure access to data that conforms to business requirements and environment without requiring unnecessary additional log-ins.
- Intra-portlet interactivity: Out of the box portlets ensure availability of all MicroStrategy Web functionality such as drilling, pivoting, page-by, sorting, and adding calculations and data groupings.
- High Performance: Developed using AJAX/Web 2.0 technology, redundant CPU cycles are no longer needed.
- Extranet capability: MicroStrategy portlets can easily be used in portal environments with firewalls.

6.9 DATA INTEGRATED WITH GEOSPATIAL INFORMATION SYSTEMS

MicroStrategy provides out of the box integration with leading Geospatial Information Systems (GIS) enabling organizations to visualize their data in a map. Two GIS connectors, one for ESRI maps and the other for Google maps, are included with the platform. The GIS connectors act as a translation bridge between MicroStrategy BI platform and GIS Systems by mapping reports and attributes to geo-objects enabling business users to view the reports in a spatial format. Popular BI functionality such as drilling, thresholds, or selectors controls are available on maps making it a consumable third format, in addition to grids and graphs, to visualize data. The out of the box GIS connectors support common map specific functionality such as bubbles, markers, or color-coded regions to identify data points, as well as zooming into and panning left and right on the maps. Maps are displayed using either Flash or DHTML.

A Generic Mapping Services Layer

The mapping services layer provided with the MicroStrategy architecture is designed to support integration of map functionality from any GIS vendor. It can be easily extended to any GIS product. GIS connectors are designed to be plug-ins, which when deployed into the MicroStrategy Web plug-in folder, automatically map the geo-codes to MicroStrategy attributes and renders BI data in a map visualization. MicroStrategy can connect to mapping products hosted in the Cloud or on a local server.

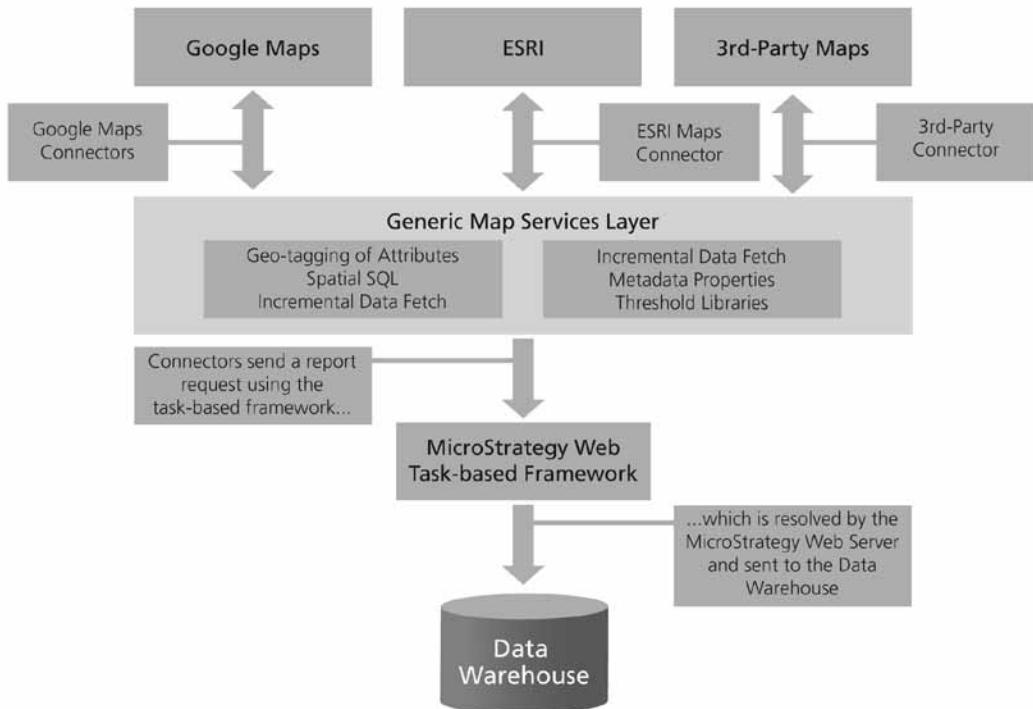


Figure 6-11 MicroStrategy Web merges information from 3rd-party mapping tools with business intelligence data.

Comparison between ESRI Maps and Google Maps Connectors

Connectors for both mapping services are provided out of the box with MicroStrategy Web and offer business users the capability to view data either in an ESRI or a Google map. The table below compares the features offered by the MicroStrategy GIS connectors to ESRI and Google maps.

GIS Integration Feature	ESRI	Google
Areas on the map	X	X
Bubbles on the map		X
Markers on the map	X	X
Color-coding using thresholds	X	
Drilling	X	X
Map as selector control	X	X
Map as target for selector controls	X	X
InfoWindows on a map	X	X
Customize InfoWindows	X	X
Zooming and Panning	X	X
Dynamically change metric displayed	X	X*
Map in DHTML	X	X
Map in Flash	X	X
Map in Visual Insight analysis	X	X
Satellite maps		X
Map with terrain		X
Map as hybrid map		X**

* Available in Visual Insight only

** Available in Visual Insight and Flash

Figure 6-12 Table showing a comparison of MicroStrategy's integration with ESRI and Google maps.

6.10 SUMMARY

For organizations planning to deploy their first BI application, and for those organizations consolidating tens or hundreds of applications into a common BI platform, MicroStrategy Web provides:

- An efficient, scalable, and high performance Web architecture
- Independence from hardware, software, and operating environments
- An extremely intuitive and easy-to-use interface
- A single BI interface that supports all of an organization's users, BI applications, and data sources
- Easy integration with major Enterprise portals
- Support for displaying BI data in maps
- Unparalleled security architecture

7

MICROSTRATEGY MOBILE ARCHITECTURE

Computing has entered its fifth generation with desktop internet applications giving way to mobile applications. With the improvement of wireless networks, availability of more bandwidth, and plethora of smartphones and tablets, business users are increasingly relying on mobile devices for receiving information and making decisions. By being able to provide critical insights on mobile devices, businesses are shortening lead times by enabling executives to make crucial decisions at the moment's notice.

As Mobile devices become a common interface for conducting business, organizations will demand a BI platform capable of satisfying a set of critical conditions:

- High performing to large user populations
- Secure, yet extremely simple to deploy and maintain
- Capable of delivering business intelligence, write-back transactions, and multi-media content
- Make the best use of the convenience and interactivity of mobile devices

7.1 CORE DESIGN PRINCIPLES

MicroStrategy Mobile inherits and builds on the MicroStrategy Web architecture to provide a unified BI platform and Mobile architecture that delivers not only business intelligence dashboards and reports, but fully-fledged mobile applications. MicroStrategy ensures successful mobile deployments by rigorously adhering to the following fundamental design principles:

- Utilize a Service-Oriented layered architecture
- Build a platform-independent architecture on a single code base
- Ensure data security for all users
- Provide BI functionality through native mobile apps
- Support all available data formats, grids, graphs, and maps
- Take full advantage of mobile devices' native controls such as cameras and GPS

7.2 DELIVER MOBILE APPS IN A MULTI-TIERED ARCHITECTURE

The MicroStrategy Mobile app is the primary user interface of the MicroStrategy BI platform on mobile devices. MicroStrategy Mobile is fully integrated into the MicroStrategy BI platform and shares the same

metadata, data sources, and Intelligence Server as the other user interfaces. A Mobile-based deployment of MicroStrategy's BI platform consists of four primary tiers:

1. Data sources, such as a relational database, ERP system, MDX cube, flat files, web service or a combination of these.
2. MicroStrategy Intelligence Server, operating on a wide variety of hardware and operating system combinations.
3. MicroStrategy Mobile Server, operating on a wide variety of application server, and operating system combinations.
4. The Mobile App installed on the mobile device.

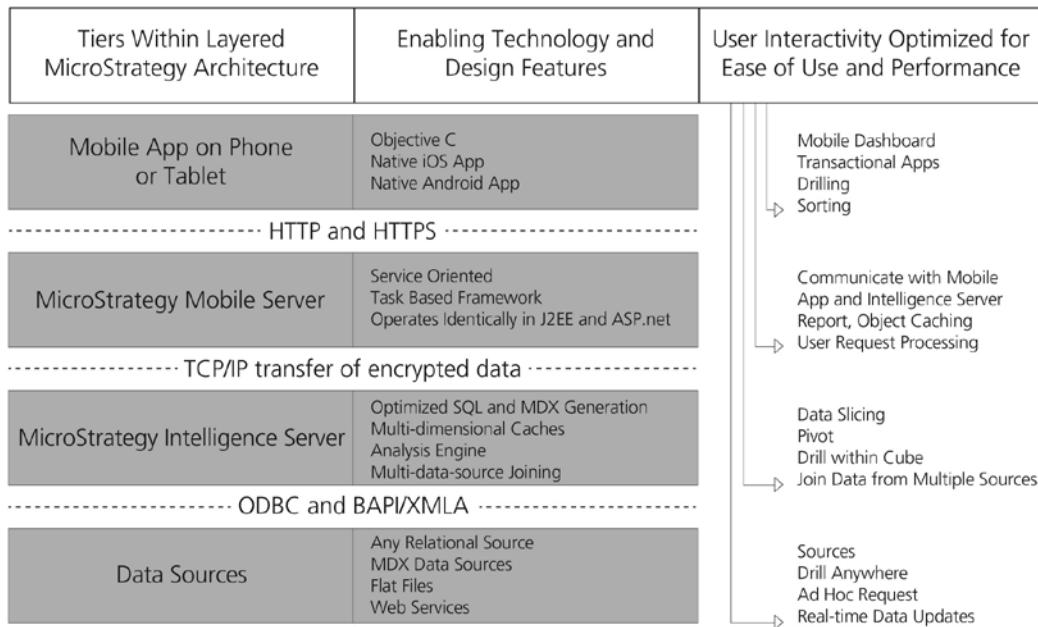


Figure 7-1 MicroStrategy's multi-tier BI architecture ensures that enterprise-class BI capabilities are available on mobile devices.

MicroStrategy Mobile's Role in the Multi-Tier Architecture

As described above, MicroStrategy Mobile is made up of two components, the Mobile App and the Mobile Server, that connect to the rest of the MicroStrategy BI platform. The Mobile App is installed on the mobile device and performs two major functions:

- Display app and report information, with full formatting and interactivity, on the mobile device
- Translate end user actions into requests to change the information displayed, or to retrieve new information. Some requests occur immediately on the Mobile App, while others require communication with the Mobile Server.

The Mobile Server's role is to act as a communication medium between the mobile device and MicroStrategy Intelligence Server. The following steps describe how a user request for a report flows through the MicroStrategy tiers:

1. The Mobile App connects to the Mobile Server via wireless or cellular networks using an HTTP or HTTPS connection and sends user requests for reports to the Mobile Server.
2. The Mobile Server translates user actions into XML requests that are transmitted via TCP/IP to Intelligence Server. These requests may be for report results, metadata object definitions, or ad hoc calculations.

3. The Intelligence Server retrieves data from cache, or from data warehouses, operational systems, cubes, web services, and the metadata repository, adds analytical richness to the data, and formats it.
4. The Intelligence Server returns report results, formatting, security privileges, and other information to the Mobile Server.
5. The Mobile Server sends the data to the Mobile App, which renders the information on the mobile device.

Note that many user requests are satisfied directly by the Mobile App using the device cache, without communicating with Mobile Server, and most user interactions and job requests occur within a few seconds for hundreds or thousands of users simultaneously.

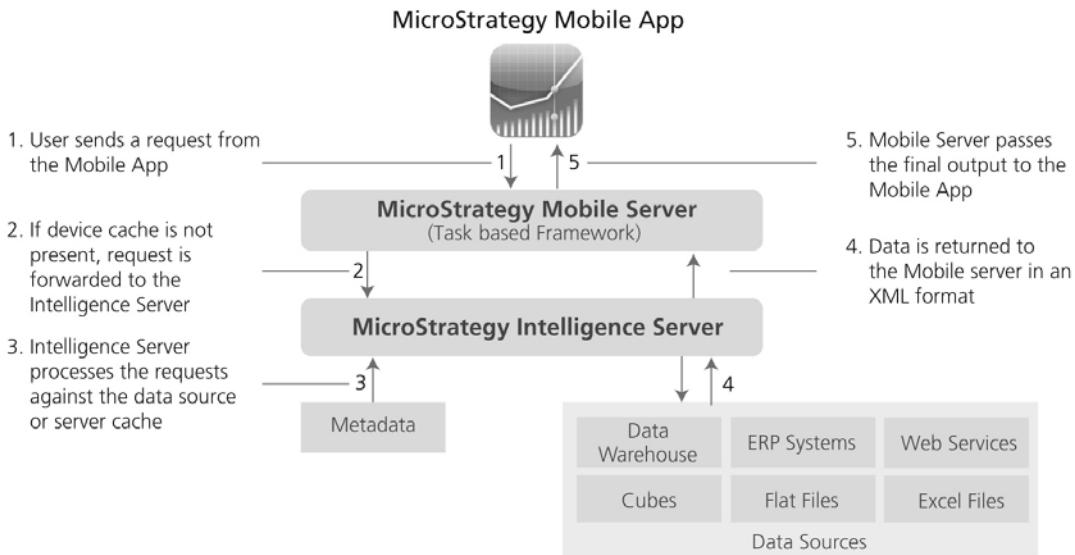


Figure 7-2 The MicroStrategy Mobile Server handles all communications between end users and MicroStrategy Intelligence Server.

7.3 MICROSTRATEGY MOBILE APP ARCHITECTURE

The MicroStrategy Mobile App is installed directly on the mobile device and performs a number of functions:

- Displays report information in grids, graphs, visualizations, maps, and multimedia containers
- Interacts with reports by scrolling, paging, sorting, drilling, and changing views and data slices
- Interfaces with the mobile device features such as location services and the camera
- Captures user input, while online or offline, and send the data to the BI platform
- Stores configuration information and report caches

The MicroStrategy Mobile App uses a layered architecture that separates the data access logic from the presentation and control logic, providing an efficient internal workflow. It is written using the most proficient code for each supported operating system – Objective C for iOS and Java/HTML 5 for Android – ensuring an effective data retrieval and on-device data management. Additionally, interface and application customizations are much simpler, since the cleanly layered architecture isolates modifications to any single layer.

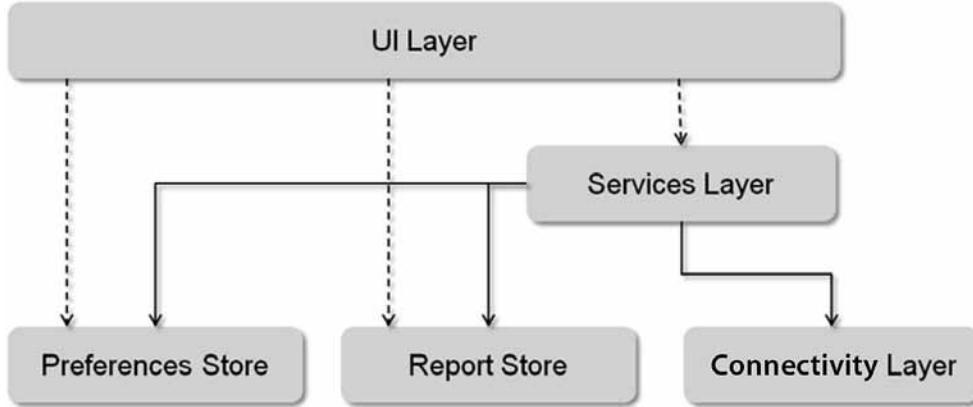


Figure 7-3 The Mobile App architecture provides distinct functional layers for efficient processing of requests, and easy customization and integration.

The UI Layer is the Visible, Interactive Layer of MicroStrategy Mobile

The UI layer sits on top of all MicroStrategy Mobile layers. It passes user requests to underlying layers, displays data using a number of different visualizations, and interprets gestures input by the user. A large number of actions such as a login, report reconciliation, report requests, and report manipulations are initiated in the UI layer which passes the action requested to the Services Layer for processing.

The Services Layer is a Gateway for Processing User Requests

The Services layer serves as the control mechanism between the UI layer and other layers. Its main function is to facilitate the retrieval of report results for the UI layer. For each data request, a service request checks if the report results are stored in the report repository or whether it should invoke a network request to the BI platform to retrieve the data. The Service layer also processes the other actions invoked by the UI layer and Connectivity layer, and interacts with the mobile device's camera, sensors, and location services.

The Preference Store Manages the Configuration Settings

The Preference store manages the MicroStrategy Mobile app configuration parameters, such as connection settings, server configuration settings, project settings, caching settings and other application-specific settings. These parameters are used to connect to the BI servers and applications.

The Report Store Ensures the Integrity and Consistency of the Report Data

The Report store contains report lists and report data caches in a single, encrypted repository on the mobile device. All data requests made from the services layer are processed by the Report store. If the data exists in the cache repository, the cache is sent to the services layer. The report store is designed to service multiple report requests, clean up caches, and reconcile report lists, while ensuring high performance of the MicroStrategy Mobile app.

The Connectivity Layer Handles the Communication Between the Mobile App and the Mobile Server

All external communication requests to the Mobile Server by the Services layer are performed by the Connectivity layer. These communication requests are for lists of available reports and report data. The Connectivity layer manages the security of the connection and incrementally retrieves the data to enhance performance.

7.4 MICROSTRATEGY MOBILE SERVER ARCHITECTURE

Since the MicroStrategy Mobile Server was built using MicroStrategy Web, its internal architecture consists of JavaBeans, Java classes, XML files, a servlet or ASP.NET container, and a thin layer of either ASP.NET or JSP pages. With this architecture, the Mobile Server operates on virtually any operating system, application server, and Web server combination, and serves reports and dashboards to both iOS and Android mobile devices.

Configure MicroStrategy Mobile Implementations

The MicroStrategy Mobile Server acts as a conduit between the mobile devices running the MicroStrategy Mobile App and the underlying MicroStrategy BI platform. There are two sets of configuration parameters that can be set in the Mobile Server:

1. Connections to MicroStrategy Intelligence Servers

The Mobile Server routes requests from mobile devices to the BI platform. Each Intelligence Server connection used by the Mobile Server can be configured to specify which TCP/IP port, number of connection threads, a load balancing factor for clustered Intelligence Servers, timeout settings, and whether encryption is used for data transmission.

2. Connection parameters for mobile devices

The Mobile App has many configuration parameters that describe how the app connects to the Mobile Server, which MicroStrategy applications are available, what the home screen of the Mobile App contains, how the users are authenticated, and different governing settings. Although these parameters can be entered directly in the Mobile App, it is much easier to define them once on the Mobile Server, and have all mobile users apply them through a URL link.

Increase Flexibility and Reduce Future Risk

The Mobile Server uses a single code base that is truly platform independent, and ensures that updates and upgrades occur on all supported environments simultaneously. The same reports, functionality, and user experience is delivered regardless of the choice of application server, operating system, hardware or ASP/JSP front-end.

Many IT environments contain a variety of hardware and software, and require the flexibility to mix and match hardware and software across applications. Some organizations even choose to operate one hardware platform, operating system, and application server combination in their development and testing environments, and a different combination for their production systems. MicroStrategy's Mobile architecture allows and enables this flexibility today.

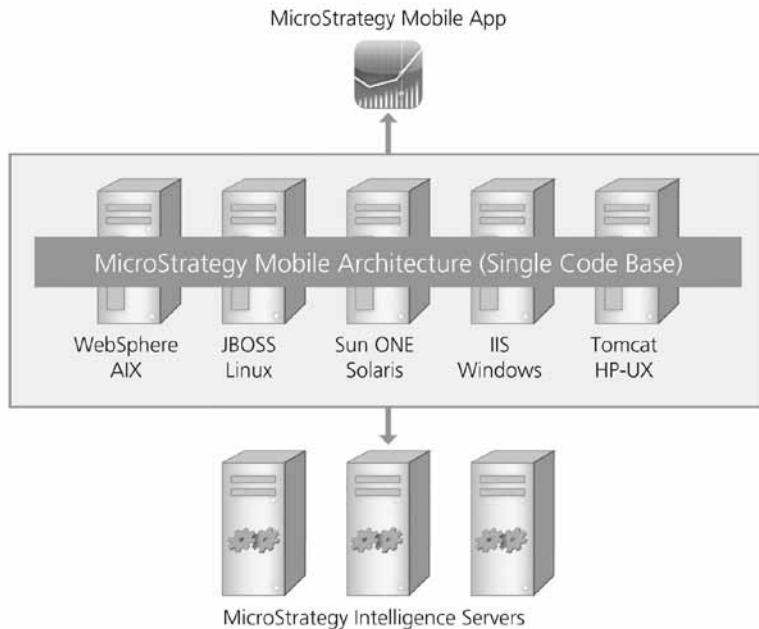


Figure 7-4 MicroStrategy Mobile's platform-independent, single code base provides maximum deployment flexibility.

7.5 ALL DATA IN THE MOBILE BI APPLICATION IS SECURE

MicroStrategy recognizes that a secure environment is vital to successful adoption of mobile devices, and provides a completely secure platform which addresses the unique challenges of mobile security. The platform is designed to protect data access, data transmission, and data storage ensuring a secure application for business users. Securing mobile BI applications and sensitive data can be abstracted as follows:

1. Security of the Mobile App

MicroStrategy Mobile takes full advantage of Apple's iOS and Android features to secure the Mobile App running on a mobile device. Since the MicroStrategy Mobile App is "sandboxed", data from MicroStrategy Mobile cannot be shared across any other app installed on the device. MicroStrategy Mobile also uses a secure encrypted keychain of storage application commands extending this functionality with its own authentication options and policy controls.

2. Data Protection

It is critical for any app to protect data from unauthorized use. MicroStrategy Mobile is designed to ensure that data stored locally on the device is secured by strong data encryption on mobile devices. Data encryption is achieved by leveraging each user's unique device pass-code with hardware encryption of the device, and generating a strong encryption key which encrypts data to AES 256 standards.

3. Data Wiping

If a device is lost or stolen, all data can be removed from the device through a remote wipe command, which also deactivates the device. Additionally, devices can be configured to initiate a local wipe of the data after several failed password attempts. Finally, through the Mobile configuration interface, administrators can specify the maximum cache size for users and configure the app to clear all local data when the app is closed.

4. Confidential Project Authentication

Organizations may want to keep certain Mobile BI applications limited to only a select group of users. In

these cases, MicroStrategy Mobile provides the capability to set certain projects as confidential. To access a confidential BI Application, a user must provide a user name and password before accessing it.

5. Wireless Transmission and Public Network Security

MicroStrategy Mobile leverages the inbuilt support for VPN on mobile devices providing the strongest security available. MicroStrategy Mobile also supports HTTPS communication between the mobile device and the MicroStrategy Mobile Server. This ensures that all communications are encrypted and authentication is based on an X.509 digital certificate. Additionally, MicroStrategy supports mutual authentication supported through a Certificate Server

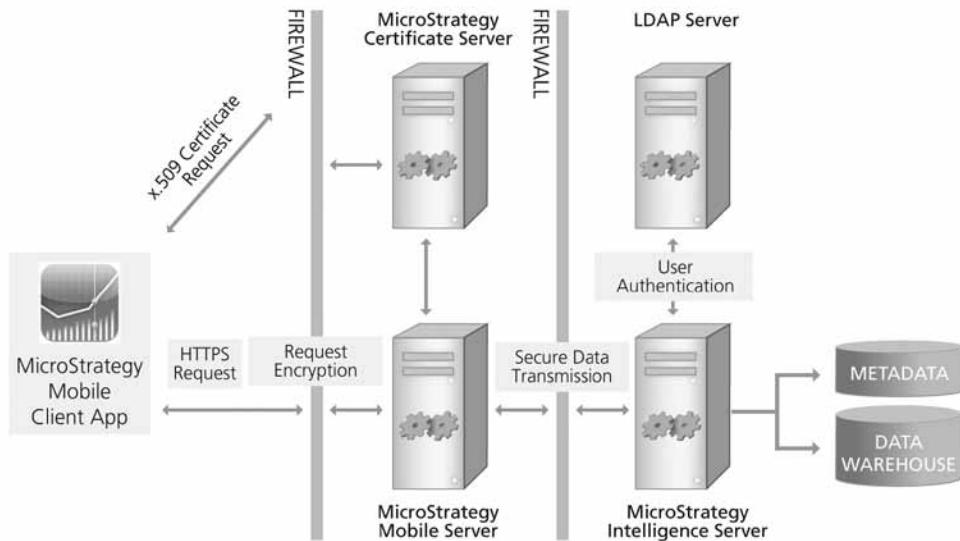


Figure 7-5 The MicroStrategy BI platform manages three layers of security: user authentication, user authorization, and data transmission/ encryption.

7.6 LEADING EDGE TECHNOLOGIES MAXIMIZE EASE-OF-USE AND INTERACTIVITY

Self-Contained App Maximizes User Adoption

- MicroStrategy provides all BI functionality through Mobile Apps for Apple iOS and Google Android smartphones and tablets. Designed with native controls in mind, the MicroStrategy Mobile App uses gesture-based actions such as tapping, pinching, and swiping available on the mobile devices. These touch-based actions are used to execute reports, change page-by selection, and drill or sort from BI reports and dashboards.

Requirements	MicroStrategy Mobile's feature supporting these requirements:
Access BI applications using Mobile devices.	Native Android and iOS apps are supported.
Deploy BI applications quickly and safely.	Users only need MicroStrategy Mobile app on their phone. The app can be automatically configured to access BI applications.
No additional coding or configuration	All functionality is provided from within the MicroStrategy Mobile app.
Conduct worry-free upgrades of Mobile Apps.	Latest version of the app can be downloaded or updated from the App Store or Android Market.
Deliver Superior Performance	Device caching ensures that reports results are cached locally on the mobile device.

Figure 7-6 MicroStrategy Mobile satisfies all user and administrative requirements.

7.7 SUMMARY

For organizations planning to deploy mobile applications for BI, transactions, and multimedia content, MicroStrategy Mobile provides:

- An efficient, scalable, and high performance architecture fully integrated in the MicroStrategy BI platform
- Independence from hardware, software, and operating environments
- An iOS and Android Mobile App that enables mobile users to view critical reports and dashboards on mobile devices
- Unparalleled security architecture
- Easy support for thousands of Mobile users

REPORT SCHEDULING AND DELIVERY

Organizations are reliant on business users having the latest information at their fingertips. Constantly growing and changing data creates the need for an automated monitoring system, searching out exceptions and delivering important business performance data to business users without having to manually seek the information out themselves.

MicroStrategy monitors business data and automatically sends reports and dashboards to users when the latest information is needed. Business rules, schedules, and events define the content that needs to be delivered, the frequency to send the information, and the list of recipients to whom the data will be sent. Built on an Industrial-Strength Business Intelligence platform, MicroStrategy provides performance, stability, and guaranteed delivery of the right information to the right person at the right time.

8.1 DESIGNED FOR HIGH-VOLUME AUTOMATED REPORT DELIVERY

Users can control individual alert parameters, and create data-driven alerts off of any metric on any report. MicroStrategy Distribution Services scans the data, looking for these business parameters, turning the system into a personal “information radar” for each business user.

Design Tenets

MicroStrategy designed its report scheduling and delivery functionality to efficiently and automatically deliver personalized information to any user in a timely manner. The underlying goals are to:

- Scale from departmental solutions with tens or hundreds of users, to enterprise-wide systems that support millions of users and messages per day
- Deliver information on a preset times and events, on exceptions found in the data, and on ad hoc or on-demand initiation of report content
- Allow a centralized rollout of information delivery or allow a self-service content and delivery process managed by business users
- Support the widest variety of delivery methods, with optimal formatting for each device and application
- Compile the broadest range of message content; from small alert messages containing links for further analysis to self-contained interactive dashboards
- Provide multi-locale support for international rollout using the same metadata objects

This exceptional performance and flexibility is made possible by a combination of enterprise-caliber technical concepts that allow for easy setup, tremendous scalability, and low maintenance overhead:

- Self-service and IT managed subscriptions
- Exception Reporting using Dynamic Subscriptions

- Alerts and Proactive Notifications
- Performance advantages through Bursting
- Superior scalability using Segmentation
- Efficient processing with Job Bucketing

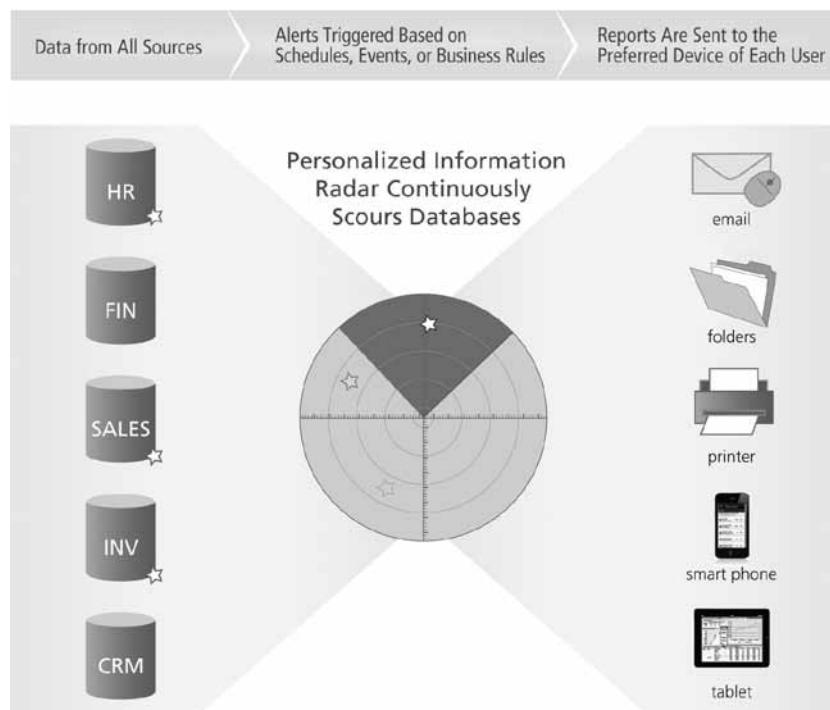


Figure 8-1 MicroStrategy's "Information Radar" lets organizations stay on top of business parameters.

Information scheduling and delivery expands the MicroStrategy BI platform with automated content delivery capabilities. Enterprises benefit from flexible configuration options that support many different use cases such as:

- Batch reporting – sending out personalized operational reports such as invoices, account statements, and business reports to employees, partners, or customers
- Business monitoring – automatically delivering scorecards, dashboards, and threshold-driven alerts
- Offline analysis – delivering reports, including the underlying data, in business application formats such as interactive Flash documents, Excel workbooks, or CSV data files

This automated delivery is controlled via subscriptions that contain all the information needed to deliver content to a recipient.

- Who: This identifies the list of people that are subscribed to receive information from the MicroStrategy BI platform. Every subscriber is identified in MicroStrategy either as a user, with his/her own login and security profile or in a simpler form for information consumers as a contact. Users and contacts can have one or more delivery addresses, e.g. E-mail addresses, file locations, or printers.
- When: This specifies the frequency that MicroStrategy will process the subscription. This can be set up as a schedule or an event. While scheduled subscriptions are triggered automatically by MicroStrategy on a periodic basis, an event must be triggered manually or using an external process, such as a Command

Manager script. Subscriptions containing alerts will only deliver content when the alert threshold is exceeded. One-off information delivery is possible using the Send Now feature, which does not create a permanent subscription.

- What: This contains the documents and reports that must be run and delivered to the subscriber. The content can be personalized with a unique set of prompt answers for every subscriber or subscriber group and subscriptions containing multiple reports and/or documents are possible, including defining the behavior when one or more parts of a multi-subscription fails or returns empty results. With E-mail subscriptions, the message subject and body can be customized and different file formats can be specified for each report or document as:
 - Adobe PDF
 - Microsoft Excel
 - Adobe Flash (embedded in a MIME HTML/MHT file or Adobe PDF file container)
 - HTML
 - Plain text or Comma Separated Value (CSV)
 - Zip



Figure 8-2 MicroStrategy delivers reports and dashboards to E-mail, print, and file server devices.

A single subscription to a report or dashboard can deliver local-language content to each subscriber using each user's language preference. Every report or dashboard is automatically personalized according to each user's security profile, prompt answers, and reporting preferences. MicroStrategy applies all security features of the MicroStrategy BI platform, such as security filters, privileges and object access control, appropriately to each subscriber in a single subscription.

8.2 SELF-SERVICE VS. CENTRALLY-MANAGED SUBSCRIPTIONS

MicroStrategy supports both centrally-managed and self-service subscriptions in the same BI application implementation as required to meet corporate needs. End users can manage their own subscriptions, power users can create and manage subscriptions to other users, and administrators can centrally manage bulk subscriptions for large delivery based deployments.

Centrally-Managed Subscriptions in MicroStrategy

The centrally-managed approach is used when information recipients do not have direct access to MicroStrategy to set up their own subscriptions. All the subscription settings – content, frequency, and recipients – are set up and maintained by a central group. The centrally managed approach delivers invoices or account statements to customers, and sends executive dashboards, scorecards, and briefing books to senior management and corporate executives in time for their meetings. The following capabilities enable efficient centrally-managed deployments:

- Bulk subscription management within MicroStrategy Desktop for rapid subscription creation
- Scripted import of user and subscription information for automated subscription management
- Dynamic Subscriptions integrates external recipient lists with MicroStrategy report delivery
- Bursting personalized information to diverse user communities efficiently

Simplified Self-Service in MicroStrategy Web

Self-service deployment, also referred to as self-subscription, gives control and personalization capabilities to the recipient directly. Business users can create a subscription to any report or document by selecting the desired schedule or event from the available drop-down list and by answering any existing prompts. Optionally, the delivery schedule, format, method, and address, along with a subject and message, can be specified with each subscription. Additional recipients can be added as well. The report content for these additional recipients will be personalized individually according to their security profiles. A subscription is automatically created in the metadata repository containing all of this information. If other recipients are included in the subscription, these users can see the subscription and modify their preferences from the My Subscriptions page in MicroStrategy Web.

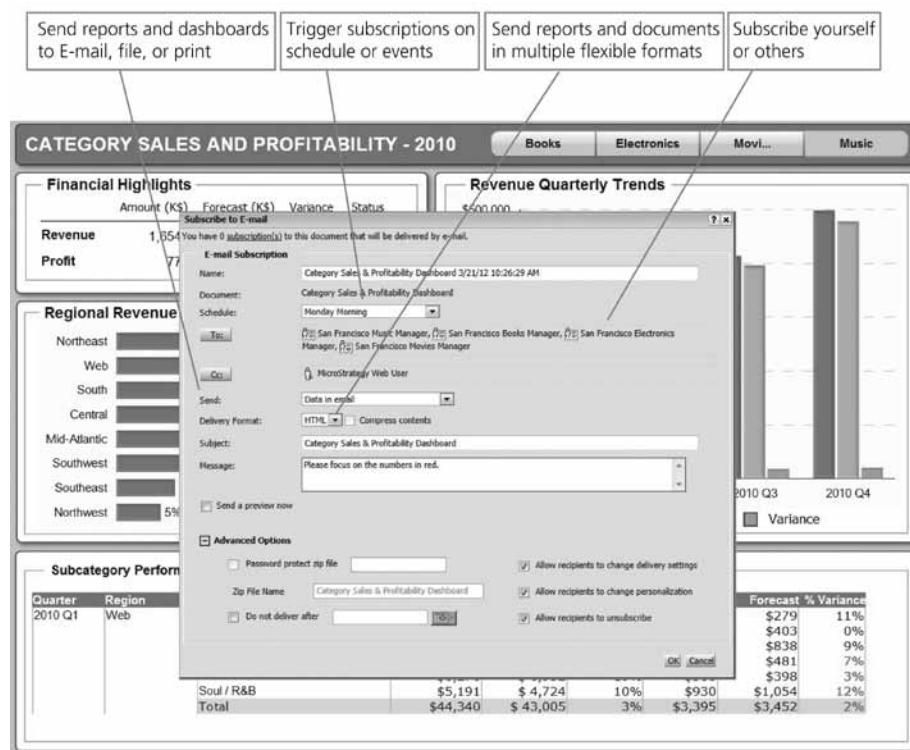


Figure 8-3 Users can subscribe themselves and others to automated delivery of reports and dashboards ensuring the latest information reaches key decision makers.

Self-service is possible using the MicroStrategy Web interface, which lets:

- Business users subscribe to any report or dashboard for E-mail, printer, and file deliveries
- Power Users manage department level subscriptions on behalf of other subscribers
- All users manage any of their subscriptions, modify and delete them

8.3 EXCEPTION REPORTING

Exception reporting in MicroStrategy is a way to only deliver information to recipients when they need it, instead of sending out standard reports to a large, fixed number of people simply based on a regular schedule. MicroStrategy can automatically deliver information to a customized recipient list by evaluating the content of a report.

Dynamic Subscriptions

Report delivery is taken to a new level with Dynamic Subscriptions which automatically determines who should receive critical information. Instead of sending information to a static list of recipients, the recipients are gathered based on the content of a subscription report. This report collects recipient and address details from a separate data source. Individual personalization of parameters is possible for each recipient. The report definition can contain complex analytics, e.g. list all store managers who experience low inventory conditions on at least 3 items.

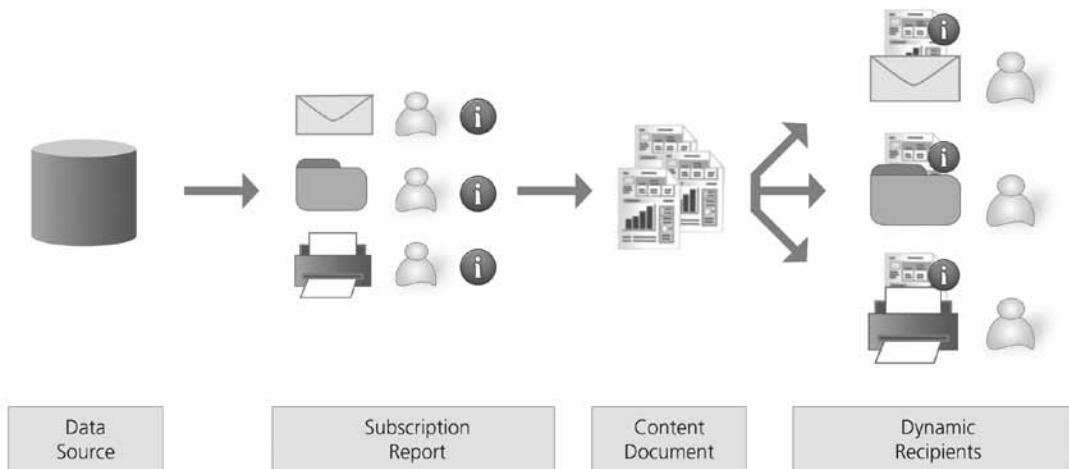


Figure 8-4 Dynamic Subscriptions generate a list of recipients at runtime from the content of a subscription report.

Dynamic Subscriptions provide the following benefits.

- Maintenance of subscription lists is reduced: For example, a weekly delivery service sends account details to the sales team. The content that each team member receives changes over time – new hires are added, members leave the team, and accounts change hands. A Dynamic Subscription Set picks up these subscription changes directly from the data warehouse or sales force automation system.
- Efficient implementation of exception reporting: Only relevant information is sent to recipients, eliminating unnecessary deliveries. Dynamic Subscription Sets are generated with the analytical capabilities of the MicroStrategy BI platform. The filtering criteria used in the report determine the list of subscribers. For example, an inventory shortage alert is delivered only to certain store managers for products with low stock levels or outages. The list of store managers who receive this regular alert is created every time the service is run.

Alerts and Proactive Notifications

Information consumers and analysts use BI applications to monitor corporate and departmental key performance indicators. High and low watermarks determine when these KPIs display problematic behaviors, spurring more detailed investigations into the causes of the values displayed. Business users can employ MicroStrategy's alerting capabilities to monitor the values of KPIs when new data is available, and send detailed reports or alert messages if the values cross the threshold boundaries.

Threshold conditions on a metric are defined either by using fixed value comparisons or by comparing one metric with another one. MicroStrategy supports a wide range of comparison functions to set up meaningful threshold conditions. An alert subscription then scans and evaluates these threshold conditions on a fixed schedule or triggered by an event, and sends proactive notifications to the subscribers if the alert conditions are met.

Alert subscriptions often evaluate threshold conditions in one report, but send a different report or dashboard that will include more details and insights to the cause that triggered the alert conditions. This allows efficient alerting from light-weight reports containing only leading indicators.

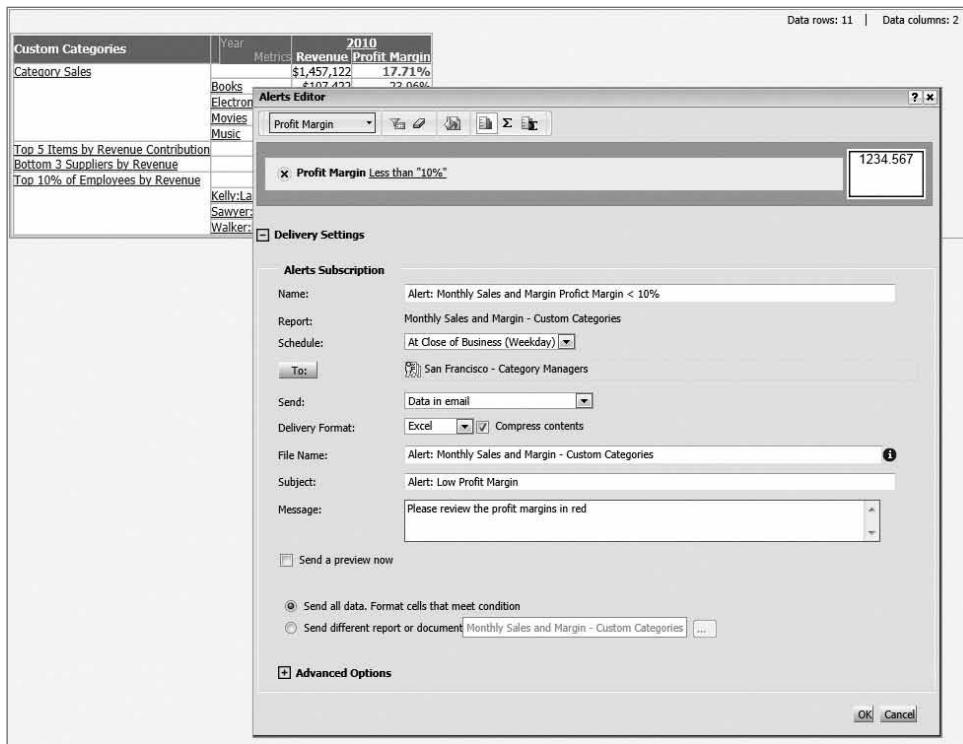


Figure 8-5 MicroStrategy Web allows Business Users to define flexible Alert conditions.

8.4 HIGH PERFORMANCE REPORT DELIVERIES

One characteristic of report delivery applications is the ability to distribute thousands to millions of reports within a specified time frame. MicroStrategy uses several techniques to ensure that a high volume of reports can be delivered to large numbers of recipients inside a certain batch window.

Report Bursting

People are often required to monitor business performance within predefined constraints. For example, store managers review sales, inventory, and labor data just for their store, while regional sales managers view the

same data for all the stores in their regions. In effect, this can be represented as a number of personalized slices of data sourced from the same data set.

Bursting, also called mid-tier slicing or personalized page-by executions, allows a single report extracted from a data warehouse to be parsed into sub-reports for individual subscribers. Rather than running different reports for each recipient, the individual report requests are automatically consolidated into a single report request that is sent to the data warehouse. This single report is then sliced into personalized subsets for delivery to the recipients. Thousands of personalized subscriptions can be delivered while ensuring a minimal load on the data warehouse.



Figure 8-6 Bursting distributes individual reports from a single database query.

Superior Scalability Using Segmentation and Job Bucketing

Segmentation is a mechanism to manage resource utilization in large scale report delivery deployments. When MicroStrategy processes large volumes of subscriptions, it divides the work into smaller sets called segments. By managing how many segments are processed in parallel, it is possible to effectively optimize resource utilization when processing subscriptions. Segmentation effectively controls job submission rate. Too few processing segments might not fully utilize the existing CPU, memory and database resources, too many will overload these resources.

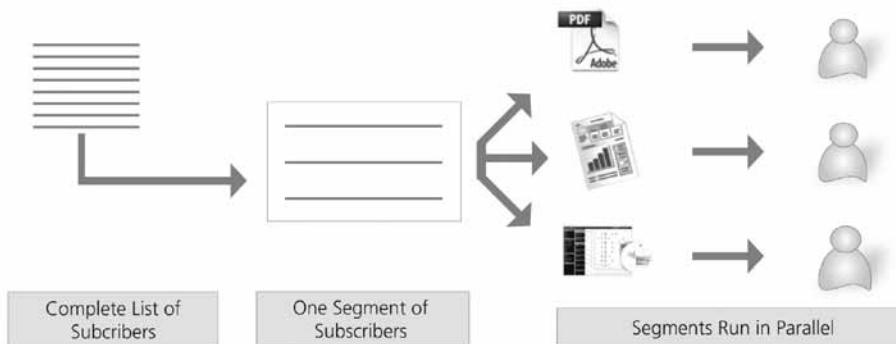


Figure 8-7 Intelligence Server processes subscriptions in segments optimizing resource utilization and controlling job submission rate.

It is common in self-service BI deployments for different business users to subscribe to the identical report or document, using identical prompt answers, filter criteria, and security profiles. Bucketing is a key step in segmentation processing that will identify common subscriptions in order to minimize the number of jobs submitted to the database and being otherwise processed. MicroStrategy analyzes the list of subscriptions and determines if several subscriptions share the same combination of security and personalization requirements for each report or dashboard. The subscriptions with common characteristics are grouped into buckets since they will deliver the same content.

8.5 SUMMARY

MicroStrategy provides a flexible architecture that is designed to support all report scheduling and delivery applications enabling high value, low cost report distribution. Its range of features enables a robust, scalable, and efficient rollout of automatic distributed reports and dashboards to corporate users, and to external partners and customers. The integration into MicroStrategy Web encourages the self-service use of MicroStrategy BI. Robust user interfaces and a highly efficient server architecture allow efficient report scheduling and delivery subscription management and processing. MicroStrategy makes it possible to deliver personalized information to any user community at any time, empowering users to make critical business decisions in a timely manner.

PLATFORM FUNCTIONALITY

9

SECURITY

Most organizations describe their BI projects as strategic or mission critical. As they make their systems available to a wider population of internal and external audiences, information must be safeguarded from potential security threats such as unauthorized access, data tampering, and eavesdropping. A truly secure BI system must have a multi-layered security paradigm; security in the business intelligence platform alone is insufficient since there are multiple points of access to the data warehouse. A wide range of security products and techniques are available for protecting the overall implementation:

- Intrusion detection software
- Antivirus software
- Firewalls and proxy servers
- Authentication schemas
- Data warehouse connections
- Security views in the data warehouse
- Application-specific security
- Control over application functionality
- Cell-level security for reports
- Encrypted data storage and transmission

The MicroStrategy BI platform integrates seamlessly with these security measures, and includes its own security features.

9.1 SECURING BI APPLICATIONS

Securing BI applications and the sensitive data they contain requires resolving three fundamental questions:

- Is the person trying to access the BI system a known user, and can his or her identity be verified?
- For what functionality, reports, metrics, and data is the user authorized?
- How can the system respond to user requests in a manner that prevents the data from being accessed by anyone other than the authenticated and authorized recipient?

More generally, these questions may be abstracted into three critical areas of enterprise software security:

1. User Authentication
2. User Authorization, including three key subcomponents:
 - i. Application functionality security

- ii. Access control lists
 - iii. Data security
3. Internet architecture and transmission security

MicroStrategy provides a BI architecture that enables corporations to confidently address all security requirements in a manner that maximizes flexibility and scalability, and minimizes administrative effort.

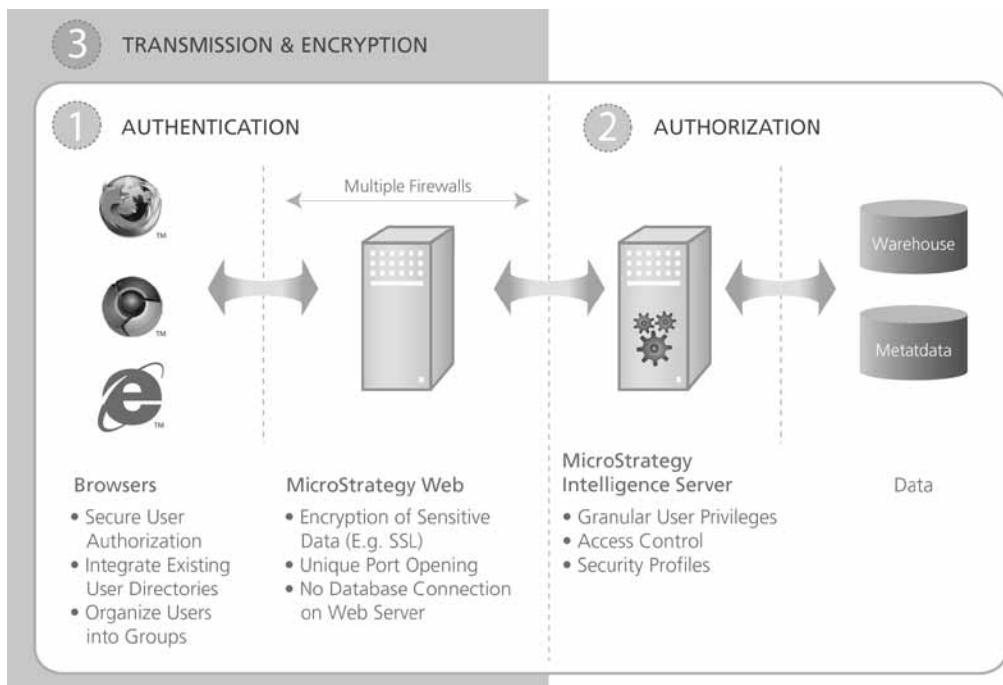


Figure 9-1 Three levels of security ensure that all security requirements are previously embedded into the architecture.

9.2 USER AUTHENTICATION

User authentication is the process of confirming a user's identity. The MicroStrategy BI platform maintains a profile for each user of the BI system. These profiles can be created using a graphical interface, scripts with textual commands, or synchronized directly with 3rd-party security systems. Validation of user credentials during initial access can be performed:

- Within MicroStrategy using “Standard” Authentication
- Outside MicroStrategy using “Linked” Authentication with third-party security systems
- Through MicroStrategy Web integration with single sign-on solutions

Standard Authentication

With standard authentication, all user logins, passwords, and security settings are stored within MicroStrategy metadata in a hashed format. Standard authentication is a manual process where a user is prompted for his or her user login and password each time a new session is started. The login and password are sent to MicroStrategy using a 160-bit one-way encryption algorithm, and is compared to the login and password stored in the MicroStrategy metadata repository; and if a match is found, the user is granted access to the system. Throughout the user's session, all security settings associated with the user login will remain in effect, and be applied by MicroStrategy in a manner that is fully transparent to the user.

Linked Authentication

Many corporations have a single, centralized security directory which governs user access to internal systems and applications across the entire enterprise. MicroStrategy integrates seamlessly with security directories through linked authentication, which is also referred to as external authentication. Benefits of linked authentication include reduced administration and maintenance of users in the BI system and increased ease of use for end users. Linked authentication is often automatic, meaning the MicroStrategy system detects the user credentials, and automatically validates those credentials against one of the following security systems:

- Lightweight Directory Access Protocol (LDAP)
- Microsoft Windows Active Directory
- Windows NT LAN Manager (NTLM)
- Database security
- Integrated Authentication (Kerberos)
- Single Sign-On: IBM Tivoli, CA SiteMinder, or SAP BW

With automatic, linked authentication, users typically use a single login to access both the network and the MicroStrategy BI platform. Business users are able to move freely between all BI applications and other enterprise applications without maintaining multiple logins and passwords. This single sign-on capability further simplifies administration and maintenance of security profiles and groups, and complementary authentication technologies such as digital certificates are applied transparently to the MicroStrategy BI platform.

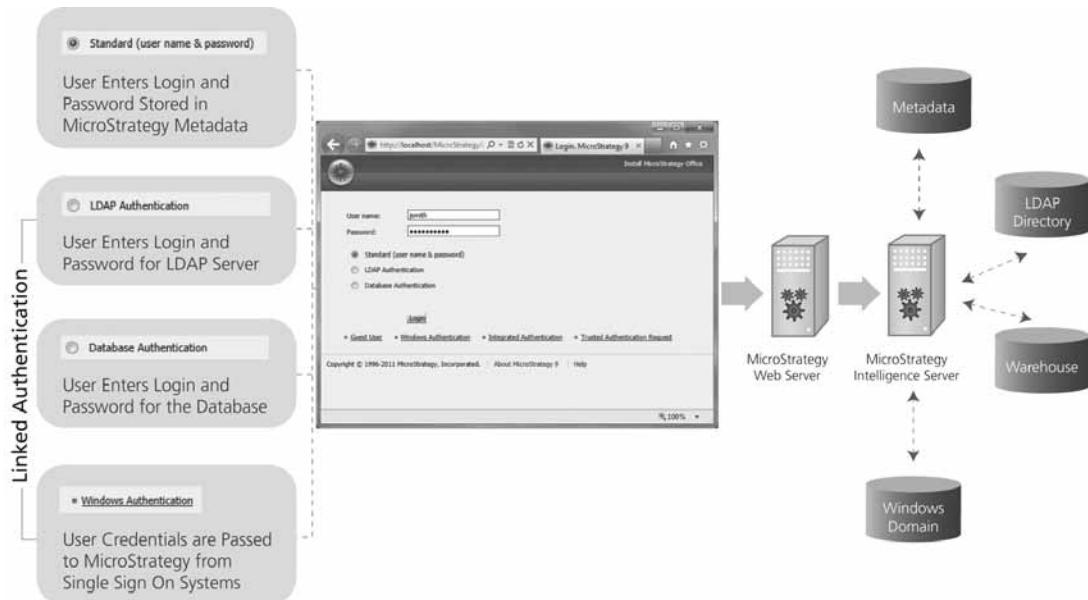


Figure 9-2 User authentication is based upon user security profiles in MicroStrategy's metadata or links to other security systems such as the database, LDAP or Windows.

Integrating with Single Sign-On Solutions

MicroStrategy also provides methods for integrating with security authentication schemes already in use within an organization by customizing MicroStrategy Web and Mobile via an External Security Plug-in. Corporate Portals and identity management products like eTrust Identity and Access Management (formerly Netegrity) or Oracle Identity Management (formerly Oblix) are common systems with which to implement single sign-on integration.

9.3 USER AUTHORIZATION

Within any enterprise application, it is crucial to distinguish between users based on each individual's knowledge, business needs, and security level. Users are not granted the same rights to application functionality, reports, and data. Once a user has been authenticated, the BI system still must enforce security policies governing the functionality, reports, and data for which the user is authorized. Authorization refers to the three-dimensional process by which an application determines:

- Application functionality privileges
- Object access permissions
- Data access security

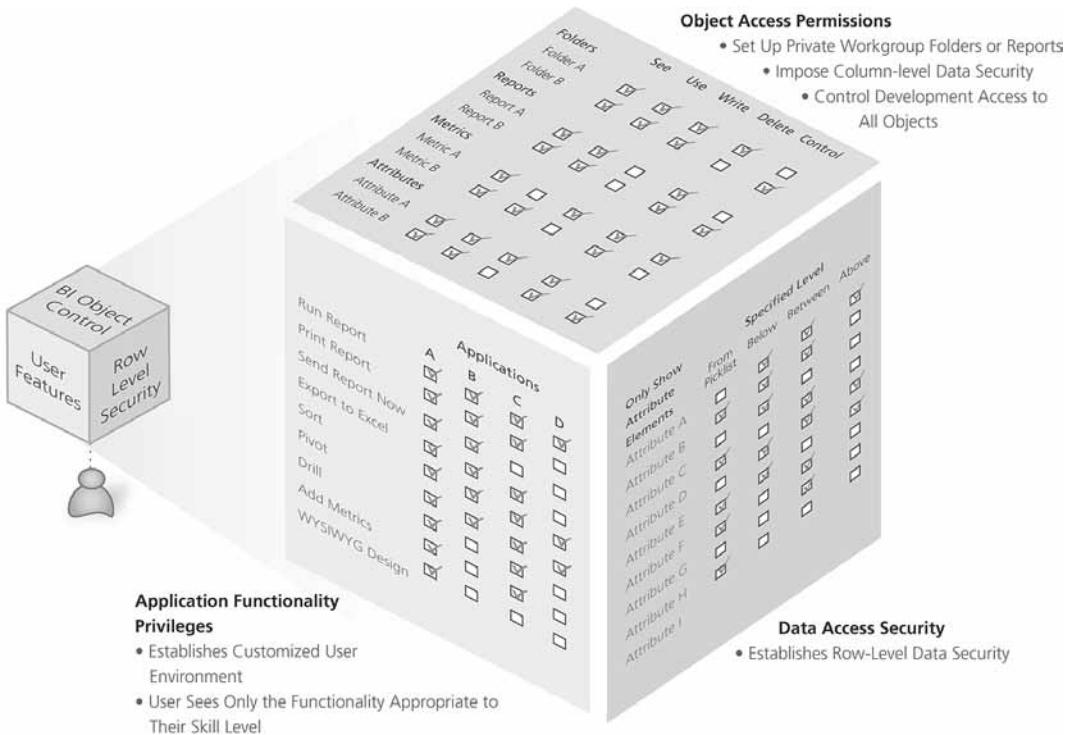


Figure 9-3 MicroStrategy's comprehensive application, object access, and data access security provides granular control for administrators.

Application Functionality Privileges

Users fall into various types, including casual users, power users, application developers, and administrators. Depending on their levels of sophistication and the software licenses purchased, certain users might need basic functionality such as running reports and sorting the results, while power analysts might need to create their own analyses and publish them. Application developers need object creation privileges, and administrators require specific monitoring and management functionality. MicroStrategy employs over 160 privileges to assign application functionality to user groups, user roles, and individual users.

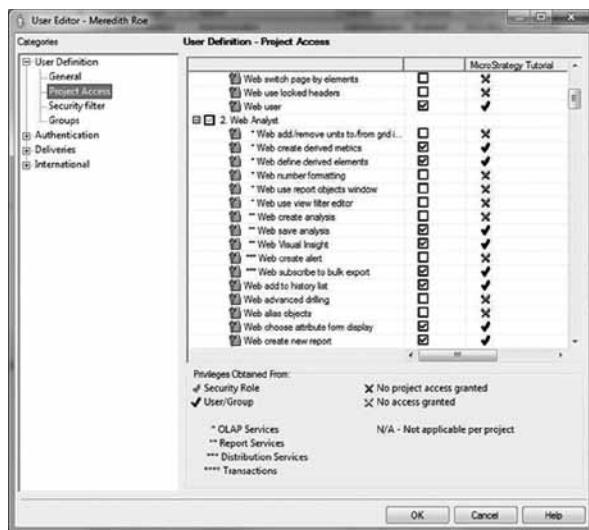


Figure 9-4 MicroStrategy Web and Desktop have graphical interfaces to assign privileges to users, groups, and roles.

These privileges can be assigned to users, user roles, and user groups through a graphical interface, and through text-based commands that can be run in batches with Command Manager.

Such fine-grained control ensures that all users access the MicroStrategy platform according to their level of proficiency. With administrators setting various levels of application functionality security, users can start using the BI applications with minimal training. Over time, administrators may grant users more privileges as they become more experienced.

Object Access Permissions

Individual MicroStrategy metadata objects are governed by their own security permissions, called Access Control Lists (ACL). Each data abstraction object, business abstraction object, report component, and report definition may have its own unique ACL, which grants users or user groups a set of privileges for the object. To simplify application maintenance an ACL can apply to many objects. MicroStrategy's object-oriented metadata allows ACLs to be inherited by child objects and applied recursively. Seven permissions can be combined to grant or deny object behavior to user groups or to individual users.

- Browse – view the object in a folder and viewer
- Read – view the object's properties (definition and, ACL settings)
- Write – modify the object's definition but not the ACL settings
- Delete – delete the object from the metadata repository
- Control – modify the object's ACL settings and take ownership of the object
- Use – reference the object when creating or modifying other objects
- Execute – reference the object when running documents and reports

The permissions have been arranged into predefined groups that reflect the most commonly used sets of permissions:

- View – contains Browse, Read, Use, and Execute
- Modify – contains Browse, Read, Write, Delete, Use, and Execute
- Full Control – contains Browse, Read, Write, Delete, Control, Use, and Execute
- Denied All – explicitly denies all object access

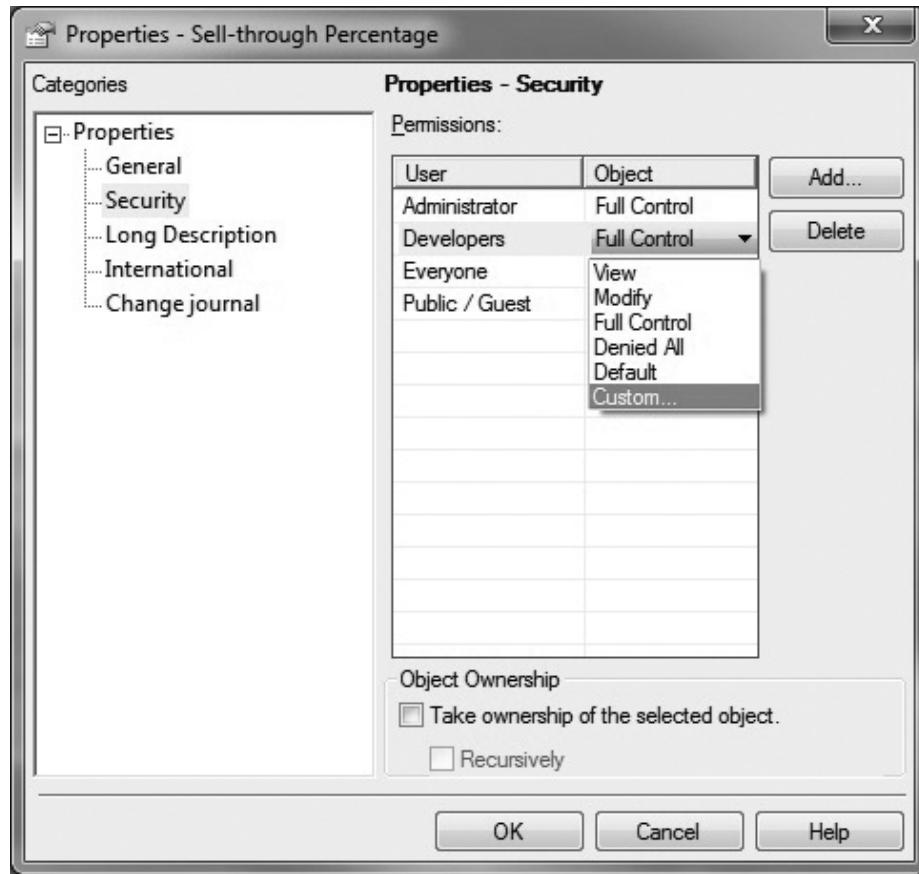


Figure 9-5 Users and user groups are granted or denied access privileges through object properties.

For example, when a report contains many metrics, each metric's "Execute" permission determines whether a user can view that metric in the report. Developers can minimize the number of reports they create and maintain in the metadata repository by making use of this feature. When two users run the same report, they will receive different versions of that report from a single report definition in the metadata, based on their ACL settings.

Different users may even be given access to completely different subsets of the data and business abstraction objects. For example, corporate expense attributes, metrics, and reports can be restricted to managers and finance department users. ACL settings can also be applied when drilling so that some users can drill down to detailed information or up to summarized data, while others cannot.

Securing Data Using Database Security

In a database, security restrictions for database logins can be placed on tables, rows, and columns.

MicroStrategy's BI platform accesses data sources using database connections. Separate database connections can be created to access the same data source with different logins. MicroStrategy users and user groups are linked to database connections using connection maps. All users allocated to a database connection will log in to the database with the same credentials, and will be subject to the security setting in the data source.

Furthermore, database views may include a restriction by database login in their definition. This login, obtained from the database connection information, limits the rows that are selected by the view when processing queries. These security views provide row-level security for every query submitted by the user. Since an administrator defines this security view inside the data source, all query tools accessing the data source with a particular login will use the view. The SQL statement used to create the database view can also be used within MicroStrategy to define a logical table in the metadata.

The main disadvantage of using database views to manage security is that performance may degrade because the view is processed at run-time. Security views must also be recreated every time new data is added to the tables in order to optimize the query that defines the view.

Securing Data with Security Filters

Security filters provide a method for ensuring row-level data security. All the filtering sophistication available in MicroStrategy can be used to limit the data that a user or user group can access. For every data source request, including documents, reports, and prompt lists, additional filtering criteria is automatically added to the query to restrict the result set to information that the user is permitted to access.

Take an example of a new Northwest Regional manager who was transferred from the South at the beginning of 2011. She needs access to all Northwest data to-date, but may also need access to South data for previous years. As a result of this security filter, all data requests will be restricted to the Northeast region, and also the South region for prior years.

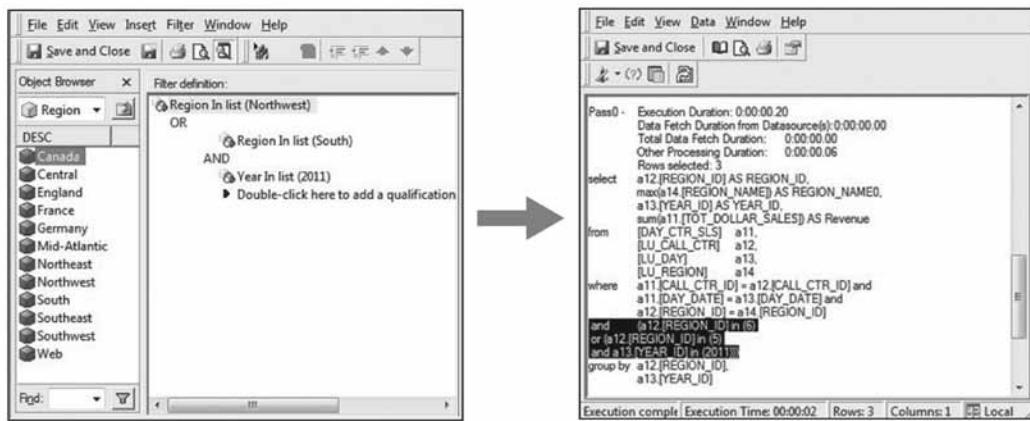


Figure 9-6 A security filter applied to Northwest Manager; the resulting SQL of a report includes the filter condition.

Security filter definitions may also specify Top and Bottom range attributes. A Top range attribute specifies the highest level of detail that the security filter allows the user to view. If a Top level is specified, the security filter expression is NOT raised to any level above the Top level. A Bottom range attribute specifies the lowest level of detail that the security filter allows the user to view. If this is not specified, the security filter can view every level lower than the specified top range attribute that otherwise meets the filter expression criteria.

Many companies use hand-coded SQL queries in their MicroStrategy BI applications to retrieve and distribute select information to groups of business users. The difficulty of ensuring appropriate data security on these freeform SQL queries can be significant. MicroStrategy addresses this difficulty by embedding security filters within freeform SQL queries. Though these SQL queries are hand-coded and static, they can include a “wildcard” that will dynamically insert the appropriate user security condition at run time while leaving the rest of the custom SQL query unchanged.

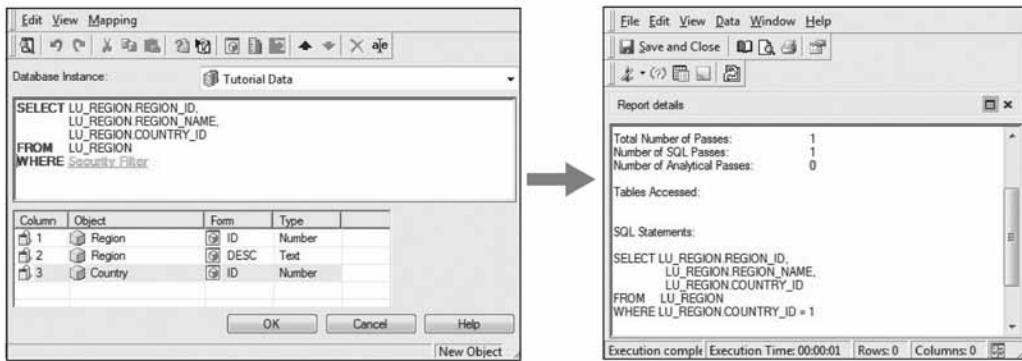


Figure 9-7 Security filters are embedded into freeform SQL queries at run-time.

MicroStrategy also provides a user login condition that incorporates individual user logins as a condition in the SQL query. This ability integrates user data access based upon security tables inside the data warehouse with security set up inside the MicroStrategy metadata. The user login prompt can be set to apply to all users of a BI application or to specific users or user groups. Even more fine-grained control is possible by including the user login condition on a report-by-report basis. For example, all reports run by a user “Brian Kelt”, would include an additional filtering clause in the SQL query with her login, “bkelt”, dynamically inserted at run-time.

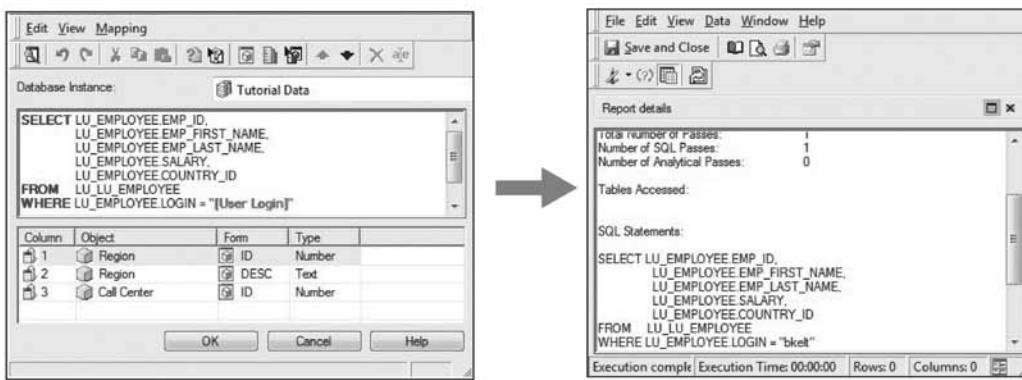


Figure 9-8 The SQL engine dynamically inserts the user id of the authenticated user.

Organizing Users into User Groups and Security Roles

With all of the granular controls for user authorization described above, administrators can easily be overwhelmed by the task of assigning security parameters for every individual. While offering maximum flexibility with full control at the individual level, MicroStrategy’s administration architecture also employs a far more

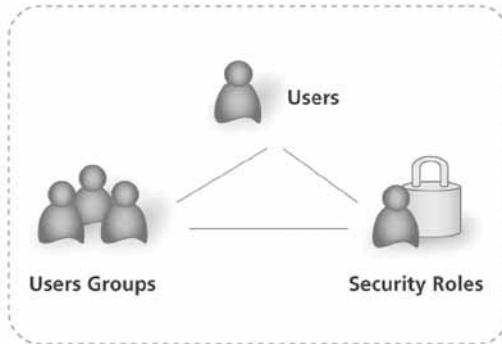
powerful and scalable mechanism to organize security profiles into group- and role-based user assignments. All elemental building blocks of User Authorization – Application Functionality Privileges, Object Access Permissions, and Data Access Security – are first assigned to a series of user groups and security roles. Users are then assigned to any number of groups and roles, and the MicroStrategy architecture dynamically calculates privileges, permissions, and security filters to create an aggregate user profile for each user upon login.

User Groups

- Are a Set of Users
- Privileges and ACLs can be Assigned to User Groups
- User Group Privileges Apply to All Projects
- Example: Users with Common Information Needs:
 - Marketing Users
 - Sales Users
 - Finance Users

Security Roles

- Are a Set of Privileges
- Security Roles can be assigned to Users and/or Specific Groups
- Security Roles Apply to Specified Individual Projects and ACLs can be Assigned to User Groups
- Example: Users with Common Functionality Needs:
 - Executive Users Need to Run, Sort, and Print Reports
 - Business Analysts Need Additional Capabilities to Drill and Change Subtotal Definitions



Users

- Are Identified by a Unique Login and User Name
- Users are Defined in the Metadata Repository
- Users Exist Across Multiple Projects

Figure 9-9 All three dimensions of user authorization (application functionality privileges, object access permissions, and data access security) may be defined at the user, group or security role level.

Security is Managed in MicroStrategy in Two Ways

- Intuitive graphical interfaces make setting up and administering security an easy task. Editors and wizards within MicroStrategy Web and Desktop provide a non-programming environment to change security definitions quickly from a single location. Users can also be imported from external security repositories.
- Consistent with MicroStrategy's emphasis on maximum flexibility and scalability, text-based commands run in Command Manager can easily make mass changes to users, groups, roles, and security profiles. These commands can be generated from third-party tools to automate common tasks and reduce administrative overhead.

The end result is that every user has a personalized user profile that governs all privileges across all BI applications and reports. Each user sees only pertinent data, and interacts with the data in a controlled way, regardless of who created the report, and how the report is defined.

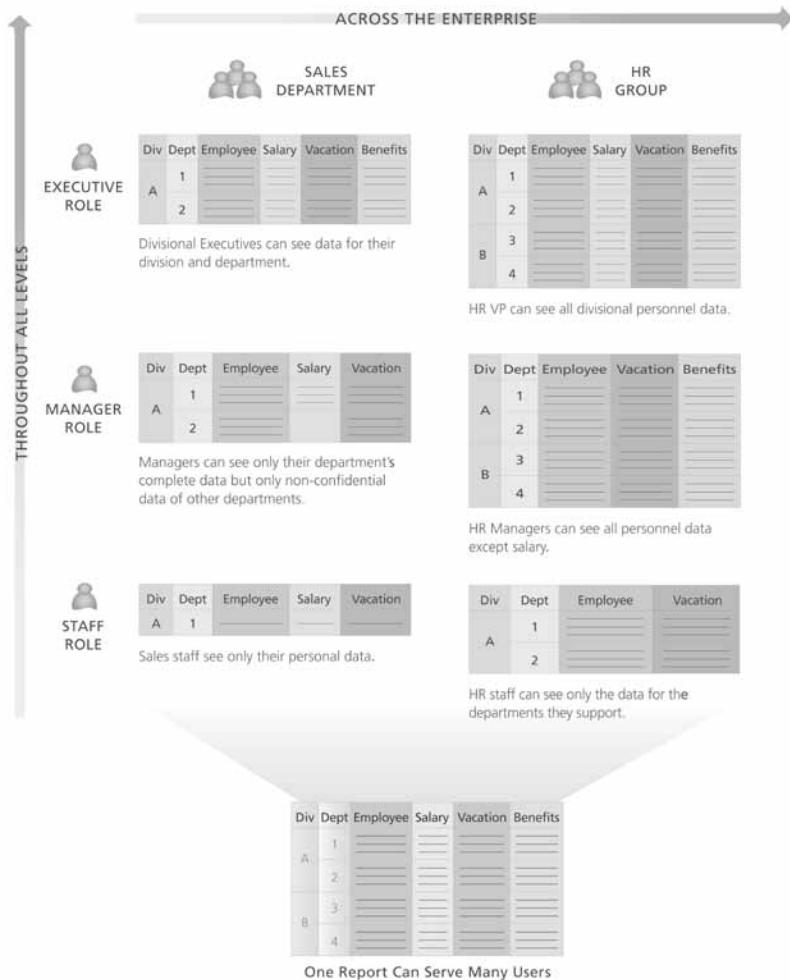


Figure 9-10 Each MicroStrategy user is assigned to any number of user groups and security roles, which combine to define each user's unique security profile. The profile is applied to every report, ensuring that all data is fully secured regardless of who developed the report or who is running it.

Import User Groups

It is possible to import users and groups from a text file, from the Windows user directory or an LDAP directory. The import feature is a convenient way to create MicroStrategy users and groups from sources where the user and group accounts already exist, such as a Windows domain or a text file.

This is useful for a project administrator who must create hundreds or thousands MicroStrategy users, and then organize them into groups based on functional responsibilities. Each user corresponds to a Windows login and belongs to a Windows group, and the Windows group structure should be maintained for the users in MicroStrategy.

9.4 MULTI-TIER WEB ARCHITECTURE AND TRANSMISSION SECURITY

As a Web-based reporting, analysis, and monitoring platform, MicroStrategy supports the most stringent Internet security requirements in two ways:

1. MicroStrategy conforms to the most widely accepted security architecture standards by using a third generation multi-tier architecture as its foundation.
2. MicroStrategy incorporates several data encryption algorithms that ensure the secure transmission of data to all users accessing MicroStrategy BI applications.

Third Generation Multi-tier Architecture

Companies are deploying more and more applications over the Web, making security requirements much more stringent. MicroStrategy addresses these requirements with an elegant architectural design that is optimized for high performance and high scalability while providing secure information to all BI constituents. Five architectural tenets ensure the integrity of the data in the BI system.

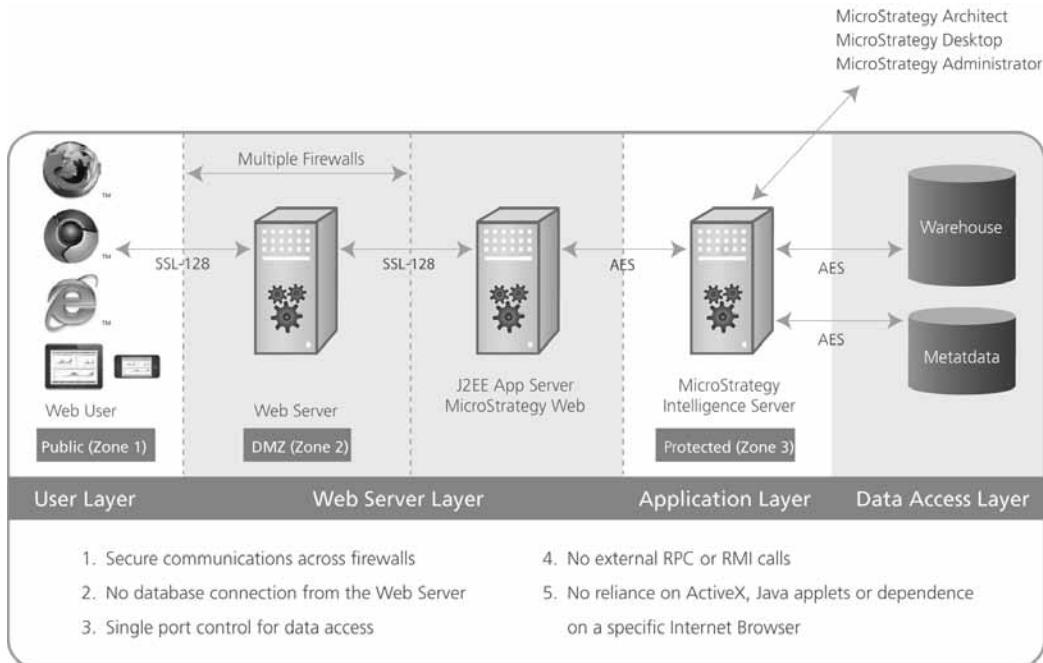


Figure 9-11 Communications between the components of the MicroStrategy platform form multiple protected zones.

1. Secure communications across firewalls

The MicroStrategy platform is normally installed on more than one server to distribute the BI workload. Secure communication across these servers is often governed by layers of firewalls constructed into Demilitarized Zones (DMZ). Using multiple firewalls, two distinct DMZ's are created with one DMZ protecting the Web server and the second DMZ securing the infrastructure of the data source and Intelligence Server.

2. No database connection from the Web Server

An effective DMZ is characterized not only by the mere presence of firewalls. It is equally important that the architectural component that accesses the database should reside behind a firewall. MicroStrategy Intelligence Server is the core of MicroStrategy's BI platform, and is the only component that accesses the database. It resides between two firewalls in the same way that the Web Server resides between two firewalls (see Figure 8-11). Only in this configuration is a hacker who gains access to the Web server prevented from accessing the database.

3. Single port control for data access

Firewalls protect corporate information assets by limiting which application has access rights to certain computer network ports. To take full advantage of this protection, the Web-based application must allow for granular port access control. MicroStrategy's Web and Mobile architecture allows administrators to configure which port is used for data access and for inter-server communication. Connections to other ports can be disallowed by the firewall, thus minimizing exposure.

4. No external Remote Procedure Calls (RPC) or Remote Method Invocation (RMI) calls
RPC and RMI calls are hazardous because they allow hackers to access and control remote and distributed computer processes. These calls often allow anonymous access through separate, open ports in the firewall. MicroStrategy Web and Mobile use only XML to communicate with Intelligence Server, eliminating the need for RPC or RMI calls completely.
5. No reliance on ActiveX or Java applets or dependence on a specific Internet Browser
Business partners and customers who access an extranet business intelligence application usually have to cross the safety of their own corporate firewall, which often prevents all incoming ActiveX or Java applets as a standard security procedure. Unlike other products that mandate the use of applets, downloaded compiled code or a specific Internet browser, MicroStrategy's Web uses HTML and DHTML, and is designed to be browser-independent. This true zero-footprint Web interface provides maximum end user functionality and flexibility without limiting Web deployment and security options.

Data Encryption and Transmission

As corporate business information is transmitted over the Internet, security of the data itself becomes important. MicroStrategy's BI platform can be set to encrypt data as it is being transmitted across the network or while it is being stored. MicroStrategy uses the following encryption protocols:

1. Secure Socket Layers (SSL)
Communication between Web browsers and the Web server can be encrypted using SSL. This uses a public and private key encryption system from Rivest-Shamir-Adleman (RSA), and includes the use of digital certificates. As SSL technology continues to improve, MicroStrategy's BI platform seamlessly takes advantage of better and stronger security.
2. Advanced Encryption Standard (AES)
This is a highly flexible, easy to implement cryptography that takes up minimal memory and produces fast data transmission. MicroStrategy applications use AES-128 cipher where the cipher has a 128-bit block size and the size of the key is 128 bits. AES-128 is also used to encrypt static report results stored in caches and Intelligent Cubes. An alternative to using AES, report transmission and results can be encrypted using TEA (Tiny Encryption Algorithm). This is a 2-way encryption algorithm that uses a 64-bit block cipher with a 128-bit key length.
3. RACE Integrity Primitives Evaluation MD-160 (RIPEMD-160)
Passwords that need to be stored are encrypted using a one-way cryptographic hash function called RIPEMD-160. This encryption occurs before the password is transmitted across the network, ensuring that network sniffers have no way of obtaining clear text passwords.

Enterprise-Class Security at Every Level of the BI Architecture

The MicroStrategy BI platform's industrial strength security architecture is built to meet the highest security requirements of any organization. Seamless integration with Windows, LDAP, databases, and other third-party systems ensures that only appropriately authenticated users are granted access to the BI system. Combining data access security with application functionality privileges and object access permissions provides granular, cell-level data security that is personalized for each individual, user role, and user group. This personalized security of corporate information combines with a multi-tier Web architecture, robust encryption functionality, and a true zero-footprint client in order to offer a layered approach to protecting sensitive data.

9.5 MOBILE SECURITY

Mobile applications are increasingly being leveraged by corporations to distribute relevant corporate data to their workforce. Mobile Business Intelligence applications offer compelling ways for enterprises to share information with employees, customers, and partners wherever they need it. Due to the nature of mobile devices, these applications (“apps”) present new security challenges that must be addressed by both the BI platform and the security capabilities of mobile devices. Data access, data transmission, and data storage must all be considered when deploying a complete and secure solution.

Securing Mobile Business Intelligence Applications

Securing mobile BI applications and the sensitive data they contain may be abstracted into the following critical areas:

- Mobile Device Control and Protection
- Security of the MicroStrategy Mobile App
- Data Protection and Encryption
- Wireless and Public Network Security

MicroStrategy provides a BI architecture that enables corporations to confidently address all mobile security requirements that maximizes flexibility and scalability, and minimizes administrative effort. In mobile business intelligence applications, it is essential to consider the security of the device itself, access to the BI application running on the device, and the security of any cached data persisted on the device.

Mobile Device Control and Protection

Apple's iOS platform enables administrators to establish strong policies for device access. All devices have password formats that can be configured and enforced over-the-air. Additionally, the iPhone provides secure methods to configure the device where specific settings, policies, and restrictions must be in place. These methods provide flexible options for establishing a standard level of protection for all authorized users. Apple mobile devices support password protection that prevents unauthorized users from accessing data stored on the device or otherwise gaining access to the device. An extensive set of password formatting options can be set to meet security requirements:

- Timeout periods
- Password strength and if it's required or not
- Maximum number of failed attempts before all data on the device is erased
- Password history
- Auto-lock device
- How often the password must be changed

The Android Operating System utilizes a secure keystore to encrypt and protect information stored on the device. To protect this keystore and prevent a compromised device from being accessed, a Device Protection Code (DPC) is used, which is established when opening the application for the first time. Once established the user must type in the DPC upon entry to the application. The Android device checks whether or not

the DPC value satisfies the criteria established by the system administrator. These following are the settings available to increase the complexity of the DPC:

- Specify that the DPC may contain a fixed number of characters from a minimum of 4 and a maximum of 8
- Specify that the DPC must contain at least one numeric character
- Specify that the DPC must contain at least one special character in the ASCII range of 33 to 126
- Specify that the DPC must contain at least one upper case alpha character

Security of the MicroStrategy Mobile App

MicroStrategy Mobile uses a secure encrypted keychain for storage of application credentials, extending this functionality with its own authentication options and policy controls. MicroStrategy Mobile enables administrative control of password policies for the MicroStrategy BI platform. These policies are configured by administrators via a web-based Mobile Configuration interface.

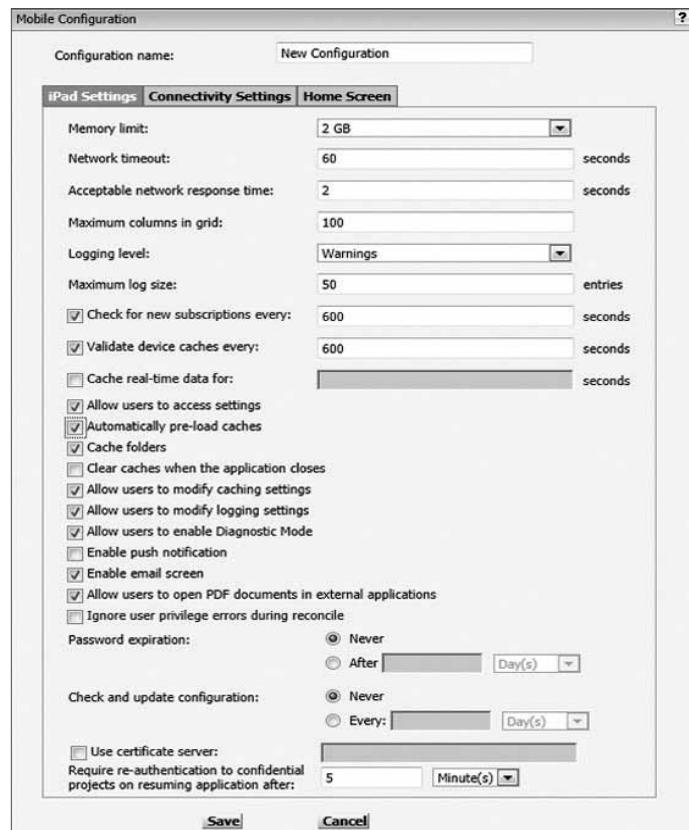


Figure 9-12 MicroStrategy's Mobile Configuration interface configures client app settings for mobile devices.

Administrators set the expiration time for the user' secure session once the user multi-tasks away from the App. After a user visits other apps on the device, and returns before the expiration time is met, they can use MicroStrategy Mobile without having to re-enter their user credentials every time. Users only need to re-enter the credentials if the expiration time has been met, according to the password expiration time defined by the administrator.

Confidential Project Authentication works in online and offline modes, ensuring security standards are put in place whether accessing live data from the Mobile Server or data cached on the device.

Data Encryption and Protection

Any MicroStrategy data stored on the device is further secured by using the data encryption algorithms available. This data encryption capability leverages each user's unique device passcode in concert with hardware encryption to generate a strong encryption key. This encryption prevents data from being accessed when the device is locked.

- iOS: iPhone 3GS, iPhone 4, and iPad offer hardware-based encryption. This encryption uses AES 256-bit encoding to protect all data and configuration profiles stored on the device. Encryption is always enabled, and cannot be disabled by users. MicroStrategy takes full advantage of iOS's encryption to protect application data cached on the device. The MicroStrategy Mobile app can be encrypted when it is backed up in iTunes to a user's computer.
- Android: Once the Device Protection Code value has been accepted per the password format criteria set by the system administrator, the DPC value is converted using the MD5 hash algorithm to a 128-bit value.

Data is further protected if a device is lost or stolen. All the data can be removed from the device by issuing a remote wipe command. This also deactivates the device. If the device is configured with a Microsoft Exchange account, the administrator can initiate a remote wipe command using the Exchange Management Console (Exchange Server 2007) or Exchange ActiveSync Mobile Administration Web Tool (Exchange Server 2010). Users of Exchange Server 2007 can also initiate remote wipe commands directly using Outlook Web Access.

Devices can also be configured to automatically initiate a local wipe after several failed password attempts. This is a key deterrent against brute force attempts to gain access to the device. By default, Apple's mobile devices will automatically wipe the device after 10 failed pass-code attempts. As with other password policies, the maximum number of failed attempts can be established via a configuration profile or enforced over-the-air via Microsoft Exchange ActiveSync policies.

The Mobile Configuration interface of MicroStrategy Mobile allows administrators to set a variety of local data caching options. Administrators can specify a maximum cache size on the device for users, automatically wipe the device of all locally cached every time the MicroStrategy Mobile app is closed, and set options for automatically rebuilding new caches for user subscriptions once the application is re-loaded.

Wireless and Public Network Security

MicroStrategy Mobile for iPhone and iPad leverages mobile device capabilities and incorporates architectural and cryptographic features that maximize the security of data transmission across the private networks where the MicroStrategy platform is managed and through private and public wireless 3G or WiFi transmission. There are three primary network protocols and configurations that should be considered when investigating the security of wireless data transmission to iPhone and iPad mobile devices.

1. Internal Use via Private WiFi. The MicroStrategy App will only be used internally (i.e., only by enterprise employees) and MicroStrategy Mobile BI applications are only accessible via the internal WiFi Network.
2. Internal Use via 3G and Public/Private WiFi. The MicroStrategy App will be used internally and will be accessible by 3G (i.e., GSM) public wireless networks and both internal (private) and external (public) WiFi networks via the internet.
3. External Use via 3G and WiFi. The MicroStrategy App will be used externally (e.g., by consumers or customers) and will communicate via the internet across both 3G and WiFi networks.

Mobile devices support proven encrypted networking technologies for ensuring that users are authorized and that data is protected during transmission. A VPN set up between the mobile device and the MicroStrategy BI platform will provide the strongest security available as it provides secure authentication using standard X.509 digital certificates to ensure that the devices can legitimately access the server, and also encrypts data communications. Implementation and set up are relatively straightforward regardless of the corporate environment and extensions to existing corporate VPNs.

MicroStrategy Mobile supports HTTPS communications between the mobile device and the MicroStrategy Mobile Server. This communication is secure in that the server is authenticated by the client (i.e., the device) and all communications are encrypted. Authentication is based on validation of an X.509 digital certificate. The underlying communications protocol is based on SSL.

MicroStrategy also supports HTTPS mutual authentication also known as two-way authentication. Mutual authentication is facilitated by the addition of a new server component called the “MicroStrategy Certificate Server”. In order to gain access to the MicroStrategy Mobile Server, the user must first enroll the device with the MicroStrategy Certificate Server.

The validation process for authenticating the user credentials can be selected by the customer in accordance with internal security guidelines and procedures. Once the device has been authenticated by the server and the server has been authenticated by the device, communications proceed with the Mobile Server using the AES encryption.

One advantage of using mutual authentication is that certificates can be issued to devices that are not associated with the enterprise (e.g. to customer devices). This is in contrast to devices operating in a Virtual Private Network (VPN) where enterprises would be reluctant to issue access through their corporate VPN to third parties; as such access often entails the ability to gain entry to servers and resources not associated with the MicroStrategy Mobile system.

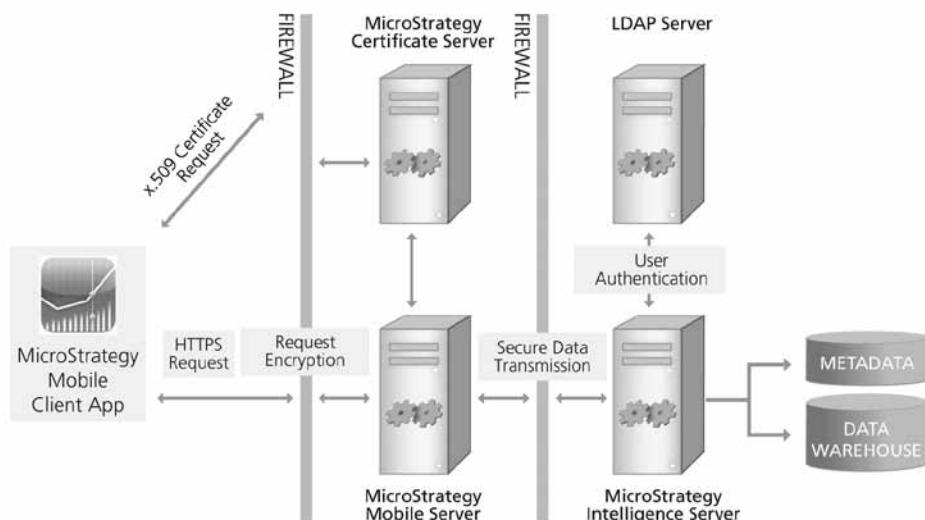


Figure 9-13 MicroStrategy Enhanced Mobile Security – A secure system with authentication usable by third parties.

Apple and Android mobile devices support WPA2 Enterprise to provide authenticated, secure access to enterprise wireless networks. WPA2 Enterprise uses 128-bit AES encryption, protecting data that is transmitted over a Wi-Fi network connection. Apple mobile devices also support 802.1x standard enabling integration with organizations that use the popular RADIUS networking protocol.

Together, these technologies are typically applied to three wireless usage scenarios above as follows:

Network	Network Security Protocol
Internal Use via Private	WiFi WPA / WPA2
Internal Use via 3G and Public/Private WiFi	VPN or HTTPS / SSL
External Use via 3G and WiFi	HTTPS / SSL

Figure 9-14 MicroStrategy supports a variety of secure wireless networking configurations.

9.6 SUMMARY

Keeping an organization's data safe and secure is critical in any Business Intelligence implementation. At the same time, this information should be ready and easily available to the appropriate people but protected from unauthorized access. MicroStrategy provides a robust platform designed to integrate seamlessly with a wide range of security products and techniques. This integration seeks to address three main areas of an enterprise software security implementation:

1. User Authentication
2. User Authorization, including three key subcomponents:
 - i. Application functionality security
 - ii. Access Control Lists/Object Access Permissions
 - iii. Data security
3. Internet architecture and transmission security

MicroStrategy's architecture enables corporations to confidently address all security requirements in a manner that maximizes flexibility and scalability, and minimizes administrative effort.

RELIABILITY AND FAULT TOLERANCE

High availability and reliable high performance are critical requirements for companies implementing enterprise-class business intelligence systems for day-to-day business decision making. Companies must be assured that their implementation is capable of delivering 24x7 availability, that it can scale easily, and that it will perform reliably under heavy loads.

MicroStrategy customers can achieve high availability via clustering, which is the ability to group multiple servers called nodes into one functional unit. If any node fails, users and jobs are transparently routed to available nodes in the cluster in a manner that avoids the dropping or duplication of job processing. This dynamic rerouting is called "failover." MicroStrategy's clustering and failover capabilities make it resilient to failures of particular nodes.

10.1 STATEFUL AND STATELESS ARCHITECTURES

Software applications can be designed to be "stateful" or "stateless." In a stateful system, significant information about the user and the service is kept in server memory (e.g., user session information, cached data, and history of recent user interactions). Stateless systems do not maintain such data, and each interaction with the software has no memory of prior interactions. Such systems use less memory, but each server request needs to include the information that would be kept in a stateful system.

Since critical components require higher performance, MicroStrategy designed Intelligence Server to be stateful. Other MicroStrategy components, including Web and Mobile are designed to be as stateless as possible. This dual approach is illustrated below:

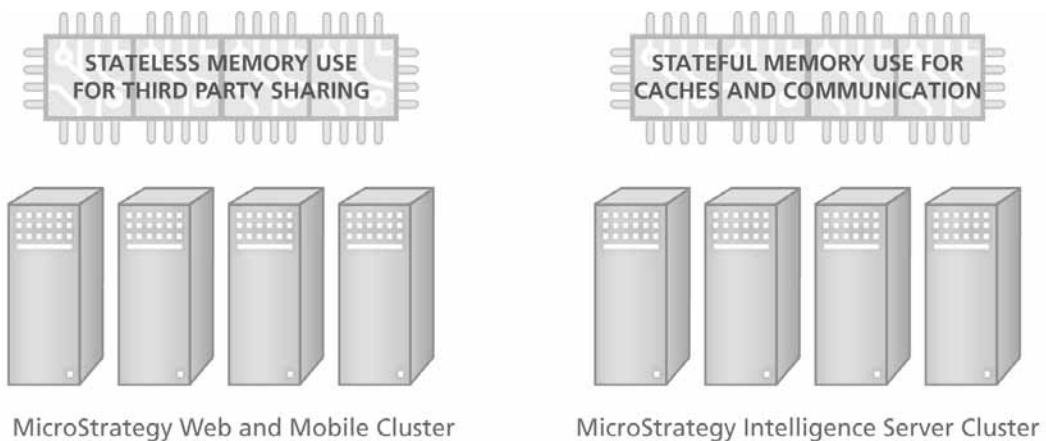


Figure 10-1 The MicroStrategy BI platform is designed to be both stateful and stateless, to work with third-party expertise where appropriate, and to provide MicroStrategy clustering capability only where MicroStrategy has core expertise.

MicroStrategy interacts with third-party components, such as the application server, Web server, file server, and database server, in a standard, stateless manner; these components can work seamlessly with third-party clustering mechanisms. MicroStrategy servers are designed to work with server system management software so that, if a server running MicroStrategy Intelligence Server stops, the load can be taken on by the other nodes in the cluster.

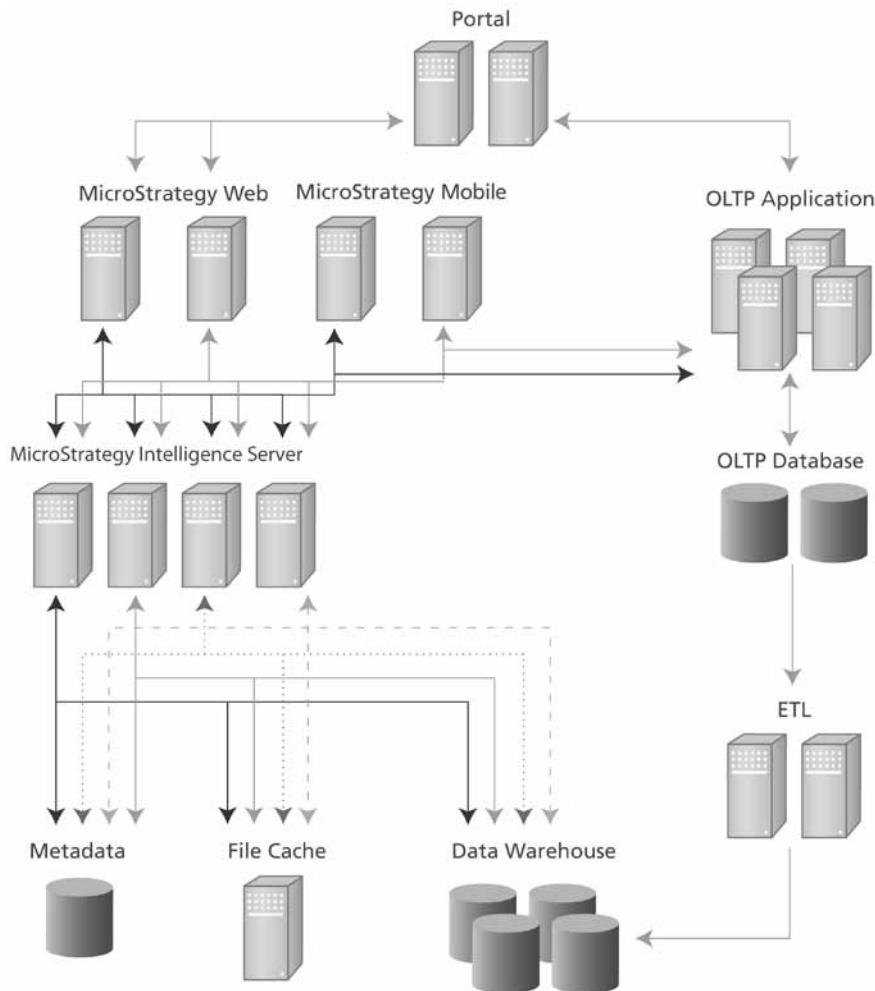


Figure 10-2 A topographical view of an enterprise system using MicroStrategy.

In the stateful portions of this system architecture, implementing effective clustering and failover becomes a complex problem due to cached session data. In order for failovers to be transparent to the user, the data that is cached must be available to a node that starts to service the jobs of a failed node. Solutions from third parties cannot effectively cluster Intelligence Server since they would be unaware of the cached data in memory. Therefore, MicroStrategy offers a clustering solution that makes intelligent use of this data to provide high availability.

Clustering business intelligence servers is a considerable technological achievement. At its core, MicroStrategy offers three key architectural characteristics that make clustering possible:

1. A completely separable and modular set of components
2. Shared, unified metadata
3. Effective intra-cluster communication

Modular Components Can Be Kept Together or Separated

The MicroStrategy architecture encourages enterprises to optimize their BI environments at all stages of the project lifecycle. When the project is small, all components (data warehouse, metadata, file cache, server, and Web server) can reside on a single server. As BI requirements shift towards critical high availability, each component can be separated from the rest, placed on optimized server resources, and clustered to provide complete reliability and fault tolerance.

10.2 SHARED, UNIFIED METADATA

MicroStrategy's customers have always required the best and most efficient use of metadata, especially when supporting the industry's largest user populations and most robust BI applications. MicroStrategy has long maintained a single shared metadata for all servers, projects, users, and reports. This shared metadata can be separated from the rest of the architectural components, and placed on an optimized database server. All nodes of the cluster can point to this central metadata repository, establishing a single version of the truth for all users.

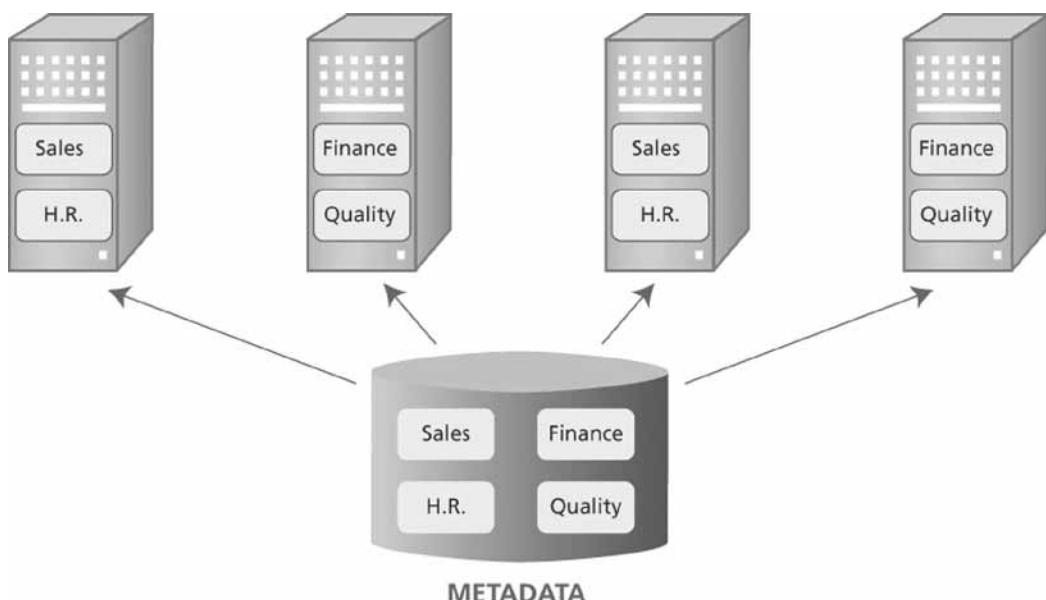


Figure 10-3 A single shared and optimized metadata for all nodes in a cluster.

10.3 COMMUNICATION BETWEEN CLUSTER NODES

A key requirement for efficient clustering is that the nodes of the cluster be able to share and synchronize information on an as-needed and scheduled basis. This capability is achieved with cluster-node communication. The architectural philosophy behind this functionality is that it is far more efficient to share and synchronize information between cluster nodes, consuming a few milliseconds of time and a few bytes of network bandwidth, than to duplicate metadata or report caches.

The Intelligence Server cluster is an all-active peer-to-peer configuration in which:

- Each node is aware of data available to other nodes
- Each node shares and can access data across all nodes

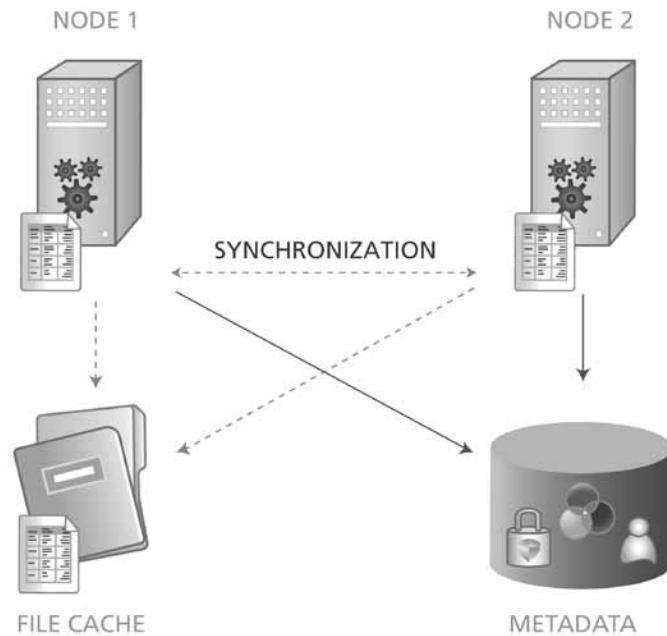


Figure 10-4 All metadata and report file cache data is available to all nodes.

Nodes maintain their own session caches and report caches. The nodes of an Intelligence Server cluster communicate every time the in-memory or the file-based report cache is changed by any node. Moreover, only the exact references to the updates are communicated, which simplifies the message, and reduces the size, complexity, and the time taken to transmit and process the updates. With the updates, each node can access cached results from all another nodes. In order to maximize network efficiency and avoid data replication, no actual report data is transmitted.

10.4 MICROSTRATEGY CLUSTERING CAPABILITIES

MicroStrategy's modular components, unified metadata, and intra-cluster communication are the foundation of a robust clustering architecture that supports asymmetric clustering and automatic failover for Intelligence Server.

Dynamic Provisioning with Asymmetric Clustering

Organizations deploy many different BI applications to monitor and analyze the performance of an increasing number of business processes. Loading many applications on a single node can consume a relatively large amount of memory. Intelligence Server addresses this problem by supporting asymmetric clustering, in which each Intelligence Server node within a cluster can host a different set of projects. This feature enables further opportunities to optimize the overall BI application, better utilizing available resources and increasing performance through:

- Further optimized resource allocation
- Added load balancing options for MicroStrategy Web, MicroStrategy Mobile, and MicroStrategy Office
- Increased agility to react to usage fluctuations
- Enhanced Failover and Fallback Logic

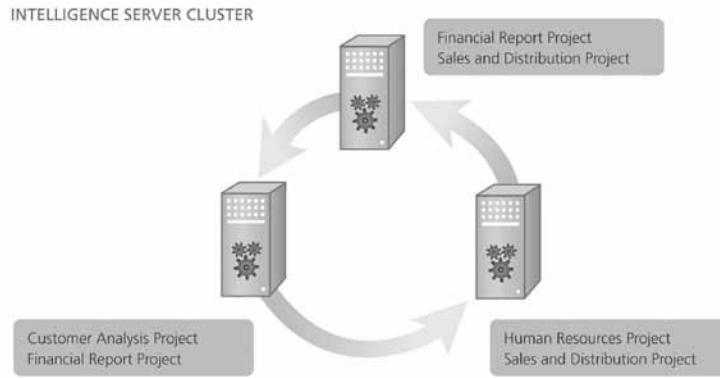


Figure 10-5 Asymmetric clustering in Intelligence Server loads different projects on different server nodes.

With Asymmetric Clustering, an organization can systematically distribute projects across different Intelligence Server nodes within a cluster. Factors influencing the distribution include:

- Usage pattern
 - Average: A collection of projects may be more resource-intensive than others. These can be better serviced by having particular nodes service only these projects, while other projects are more distributed.
 - Periodic: A collection of projects may require more resources at certain hours of the day. These projects can be better serviced at those hours by loading them on more nodes while reducing the number of nodes on which other projects operate.
- Business priorities
 - Mission-critical projects that are used by a large number of users can be loaded onto all nodes in the cluster, while projects used by few users can be loaded on a subset of the nodes. This approach not only provides excellent fault tolerance, but also ensures high scalability and performance. Distributing projects across at least 2 nodes ensures that those projects will be available even if one of those nodes becomes unavailable.

Asymmetric clustering provides flexibility in using BI resources, as the number of nodes running a project can be modified to accommodate the requirements of the users.

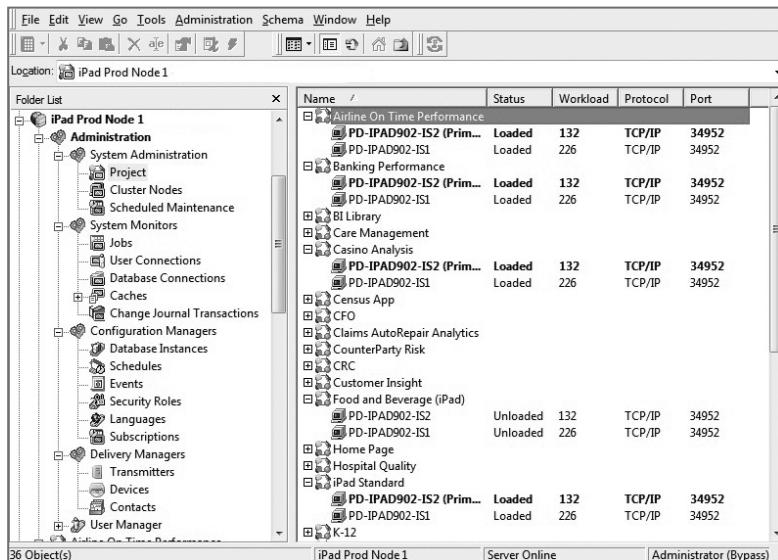


Figure 10-6 Cluster nodes can be administered easily using the Desktop graphical interface.

Clusters are administered through the administrative console in the Desktop interface, and via commands in Command Manager. It is easy to modify the capacity of a growing BI environment because it is simple to add or remove nodes from a cluster, or to load or unload projects from a node. This means that system maintenance and system growth are far more easily accommodated in a cluster than by administering a single machine. Furthermore, the status of all nodes and projects on those nodes in the cluster can be viewed in these interfaces.

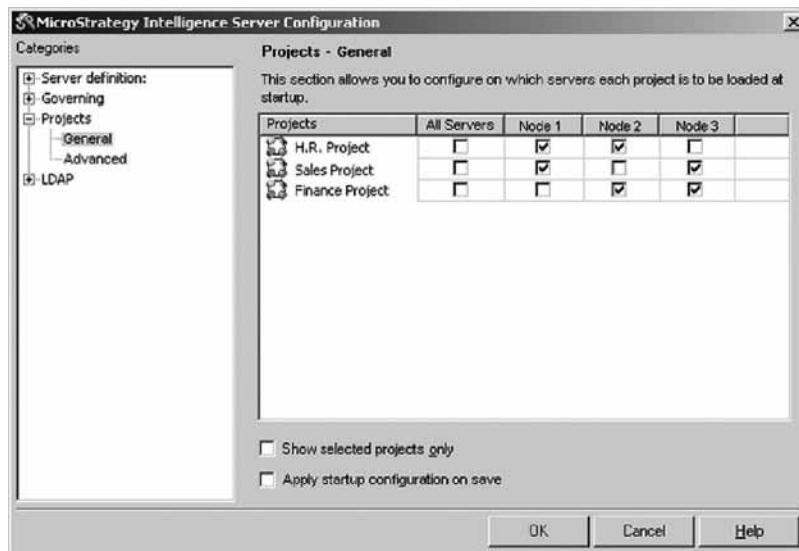


Figure 10-7 A view of the list of projects and their status sorted by the nodes of the cluster.

More Flexible Scheduling Options

Clustering also enables two new options for scheduling:

- Time Based Schedule: Execution of scheduled reports is triggered at a pre-determined time. A Scheduler session is open on the primary node in order to process the scheduled requests.
- Event Based Schedule: Execution of scheduled reports is triggered when a specific event is fired and processed on chosen nodes. This provides more flexibility in order to distribute the scheduling load across all nodes.

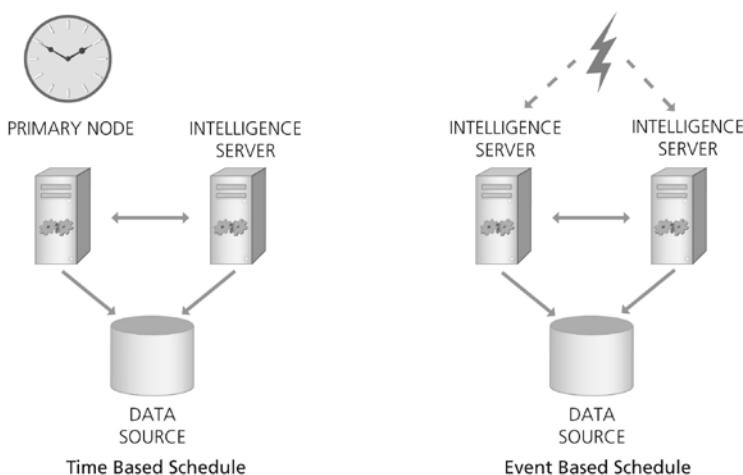


Figure 10-8 Time Based and Event Based Schedules trigger job executions on behalf of users and administrators.

10.5 FAULT TOLERANCE WITH BUILT-IN FAILOVER

In the event of a failure in one node in the cluster, different failover strategies can be implemented that reduce the amount of downtime experienced by the system and minimize the impact on end users. With MicroStrategy, the system administrator has a choice to set up hot, warm, or cold failover strategies. These options require different amounts of hardware resources for Intelligence Server.

Hot Failover Strategy

With the hot failover approach, all nodes in the cluster actively accept and process new jobs from users, and share information about report caching and actively running jobs with other nodes in the cluster. If a node fails, all user sessions in related components (MicroStrategy Web, MicroStrategy Mobile, and MicroStrategy Office) are transferred to the other nodes in the cluster. The failover process has the least impact on users as the job transfer is transparent to end users, who will not even realize that a failure occurred. Each node, however, should have sufficient capacity available to take over jobs from other nodes in case of a failure. When a failed node is restored, it automatically loads the projects assigned to it and starts processing user requests.

Warm Failover Strategy

The warm failover approach has all the nodes running but with one or more nodes acting as backups by not actively processing user requests. In this case, each node in the cluster still has access to all caches and jobs. If a node goes down, projects assigned to that node are loaded onto the backup nodes, and jobs are automatically transferred to the backup node to continue processing the user requests. When the failed node is restored, it automatically becomes a backup node in the cluster with no projects loaded and it does not process new jobs. Warm failover has a higher impact on users as compared to hot failover as some time is needed to load the projects on the backup node. All nodes in the cluster can run at capacity since a failure brings a completely new node online.

Cold Failover Strategy

Cold Failover refers to an environment where a backup machine has MicroStrategy software installed, but not actively running. If an Intelligence Server machine fails, the backup Intelligence Server is manually started and replaces the failed machine. Projects and report caches are loaded from the metadata and from a file server respectively, and the backup machine can start processing user requests. User jobs that were running on the failed machine are lost and should be resubmitted, resulting in a higher user impact. As with warm failover, machines in a cold failover configuration can run at capacity. The advantage of the cold failover approach is that it can be used in both non-clustered single-machine and clustered multi-node environments.

Strategy	Recovery Time	Relative Cost	User Impact	Clustering Required
No Failover	Unpredictable	No cost to low cost	High	No
Cold Failover	Minutes	Moderate	Moderate	No
Warm Failover	Seconds	Moderate	Low	Yes
Hot Failover	Immediate	High	None	Yes

Figure 10-9 Failover Types in terms of cost and user impact.

10.6 ADAPTATION TO CHANGING CONDITIONS

A final aspect of reliability is the ability to perform consistently under varying conditions, such as those resulting from increased or unexpected demands on the business intelligence system. To alleviate potential

processing bottlenecks that can occur as demands on the system increase, the platform configuration can be adjusted manually or automatically using one of the following mechanisms:

- Manual adjustment (by the administrator)
- Self-tuning built into the software
- Integration with system management software

The MicroStrategy platform automatically reallocates computing resources in response to changing demands on the business intelligence system. Either the administrator or system management software can adjust the asymmetric clustering, the number of nodes in a cluster, change the number of open database connections, idle the server or change report execution parameters. There are many other settings that can be used to control the platform's operations and performance at a granular level.

10.7 SUMMARY

MicroStrategy's native clustering capabilities, comprehensive real-time monitoring, and ability to adapt to changing conditions offer the industrial strength reliability and fault tolerance that are necessary to ensure 24x7 availability of the business intelligence system. Clustering at multiple levels in the business intelligence system insulates users from hardware and operating system failures, ensuring that reporting, analysis, information delivery, and transaction execution can be conducted with confidence.

SCALABILITY AND PERFORMANCE

Over the last decade, a series of technological breakthroughs have dramatically changed user perception about performance. Google and the Internet set very high performance expectations that permeate almost every aspect of our lives by what is now known as the Google effect. Today, Mobile applications are once again pushing the performance expectations even further, requiring significantly faster speed for much more users. The same is true in Business Intelligence - while users may have been willing to wait up to 40 seconds for an answer just a few years ago, today's users have far less tolerance for slow performance.

11.1 HIGH-PERFORMANCE INITIATIVE

To match the expectations of an increasingly performance conscious Business Intelligence user population, MicroStrategy embarked on a multi-year, multi-million dollar initiative designed to improve our customers' BI performance through software innovations and best practices for high performance. This High Performance Initiative is given three ambitious goals to meet the needs and requirements of today's and tomorrow's Business Intelligence implementations.

1. Deliver up to 10x faster BI applications
2. Provide faster than 3-second response time for most predictable queries and analyses
3. Provide faster than 5-second response time for the majority of ad hoc queries

Performance = Speed + Scale

The performance of a Business Intelligence system, or any system for that matter, is really about the combination of two things: speed and capacity. In many cases only one or the other is provided. For example, a race car can provide lots of speed but not capacity; a truck may provide capacity without speed. Only a more advanced system, like an aircraft, can provide both: speed and capacity.

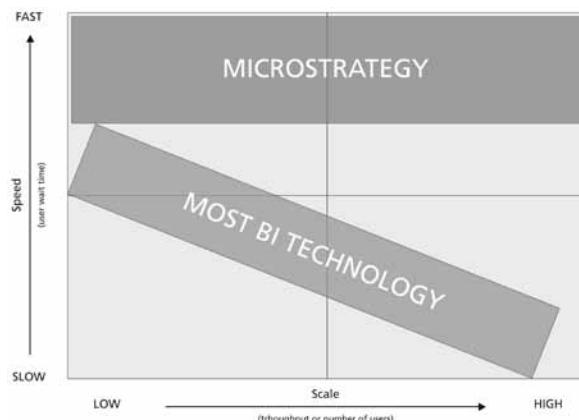


Figure 11-1 The MicroStrategy BI platform provides speed and capacity for enterprise-wide BI implementations

Throughout its history, MicroStrategy has been focused on providing both of these things: Speed with capacity or in other terms, performance at scale. In contrast, most other BI technologies emphasize one over the other, e.g. to provide very high speed but only at a modest scale. When scaling the system to higher capacity, speed starts to suffer.

MicroStrategy has always looked at both dimensions of the problem, providing the most scalable BI platform in the industry. Speed and scale have been and continue to be at the core of MicroStrategy technology. Both the enterprise data warehouse and the business intelligence system must be able to handle accelerated data growth so companies can successfully monitor, report, and analyze their operations. As the amount of information increases to the multi-terabyte size, and the number of users potentially reaches millions, it is vital that system performance does not degrade. At the core of the MicroStrategy BI modular architecture is the MicroStrategy Intelligence Server, which is designed from the ground up to meet both today's and tomorrow's business needs. It achieves maximum scalability and performance through characteristics that include:

- Multi-level Shared Caching
- In-Memory BI
- Query Optimization
- Server Tuning
- Distributable Execution
- 64-bit Technology
- Multi-threaded Processing
- Efficient Communication

Although the initial BI application may not be scoped to reach a large size, the right choice of a scalable BI platform upfront can ensure there are no restrictions to future growth. The MicroStrategy BI platform is the only proven architecture capable of scaling to hundreds of applications, millions of users, and petabytes of data in order to handle all enterprise business intelligence needs.

11.2 MULTI-LEVEL SHARED CACHING

With other BI products, query performance decreases as the number of users on the system increases. When Intelligence Server caching is enabled, and since cache tuning is based on user activity, performance either improves or remains constant. More users on the system provide the Intelligence Server with a more accurate profile of the most popular reports and data. This allows Intelligence Server to more accurately cache the right data closest to end-users for maximum performance.

Some BI products attempt to implement caching by simply storing data at a single level of the middle tier, thus missing most caching benefits. MicroStrategy Intelligence Servers implements a truly efficient self-tuning caching strategy. The foundation of caching within the MicroStrategy BI platform is to implement caches at key points within the overall query flow, taking into consideration various usage scenarios, and user behavior, to provide the most improvement.

A typical user may log into MicroStrategy, search for a report, build a new report, answer a prompt, run the report, conduct further analysis on the result set, and save the report before logging out. Within this query flow, Intelligence Server caches specific data in the following crucial areas:

- Metadata Object Caching
- Attribute Element Caching
- Database Connection Caching

- Result Caching
- Document Caching

The diagram below shows how response time dramatically decreases using strategic cache implementation within the Intelligence Server, even as the number of users increases.

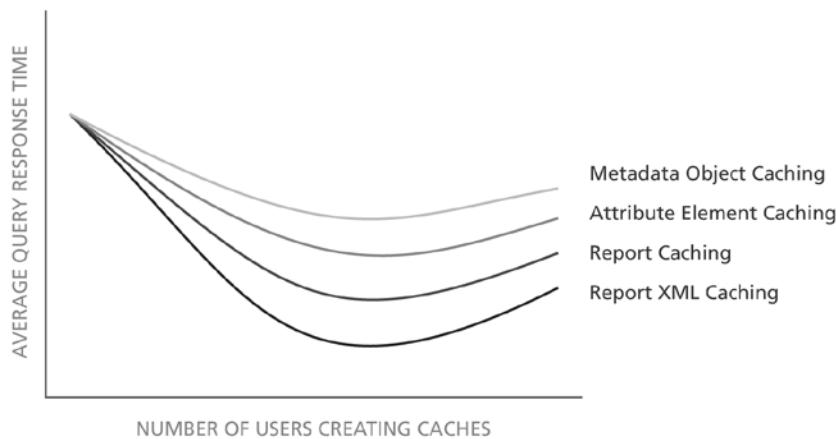


Figure 11-2 Query response time decreases while the number of users increases.

In a clustered Intelligence Server configuration, caches are synchronized across clustered nodes without the massive data transfer between nodes, and users connected to one Intelligence Server node benefit from the caches stored by another Intelligence Server node.

Metadata Object Caching

Metadata objects store information about BI applications that define the data abstraction from the data sources, the business rules and key performance indicators, and the format and layout of documents, reports, and analyses. Users interact directly with these objects and navigate between objects. A responsive system during this interaction is essential for high productivity and unhindered user experience.

Each time a Metadata Object is first accessed, its business definition is cached in memory. When the memory allocation for this cache is reached, the least frequently used object is removed from the cache. Over time, user interaction self-tunes the system to ensure that the most frequently used objects are cached.

Attribute Element Caching

Attribute elements are the row level data for each attribute normally stored in lookup tables in a data warehouse. Lists of attribute elements are displayed when answering prompts, building filters, and browsing the data warehouse. Caching these attribute element lists improves the business user experience and overall system performance.

Intelligence Server optimizes performance by caching the most frequently listed attribute elements and swapping out the least recently used lists. Attribute element lists often span thousands of elements. Intelligence Server employs further techniques for caching this data, and providing rapid response times. As specific sections of attribute element lists are requested, only these snippets of the entire list are initially cached. While the user is perusing the cached section of the list, Intelligence Server pre-fetches upcoming sections and caches them.

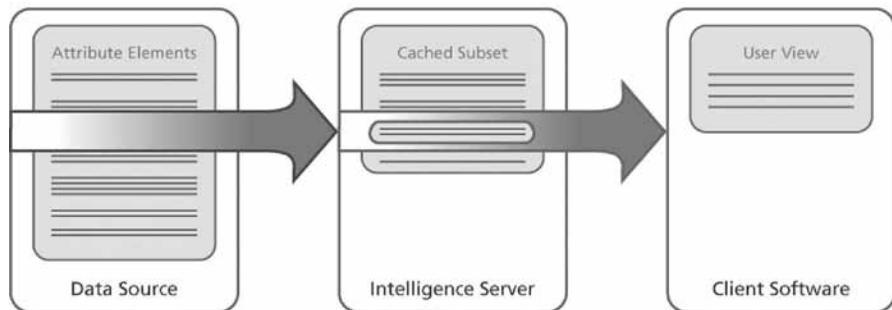


Figure 11-3 Optimized attribute element caching reducing resource requirements while boosting system response times.

Database Connection Caching

When establishing a connection, a database must allocate communication and memory resources, as well as authenticate the user and set up the corresponding security context. To minimize this overhead, the MicroStrategy BI platform implements its own connection pooling strategy via Database Connection Caching. Whenever a connection is established, this connection is kept and re-used when possible. This implementation is database and operating system platform independent, improving the end-user experience and the performance of the BI system.

Result Caching

The MicroStrategy BI platform creates result caches to avoid re-executing requests for exactly the same information against the data warehouse each time, speeding the delivery of information to business users. There are 3 types of report caches in the MicroStrategy BI platform.

1. Result set caches. These store the final results of a report or dataset after Intelligence Server has performed additional analytical processing but before formatting is applied to the results. Result set caches are stored in a highly efficient binary form both in memory and on disk.
2. Formatted PDF and Excel files. Pixel-perfect documents are cached to disk after Intelligence Server has applied all the formatting. Links to these documents appear as messages in individual users' history lists.
3. XML caches. Web receives reports from Intelligence Server in chunks of XML according to the number of rows defined in the "Incremental Fetch" setting, and the contents of the "Page-by" section of a report. These are cached in XML format improving performance for Web users.

Document Caching

Document Caching accelerates dashboard performance by storing it in its final electronic format—Adobe Flash, HTML, PDF, and Microsoft Excel. Along with the data the document cache contains information on formatting, selectors and other components that define the document. Administrators can instruct Intelligence Server to only create cache files in the document formats that are frequently accessed.

Automatic Data Management

Administrators of other BI software spend significant amounts of time maintaining cubes based on business requirements. The cubes reside in multiple locations, such as user desktops, Web servers, and file servers. The data is duplicated across these locations, and time spent managing the data quickly grows exponentially. In contrast, MicroStrategy Intelligence Server automatically builds and refreshes data. Maintenance is orders of magnitude easier and security is always enforced.

Custom Caching for Maximum Efficiency

To further boost performance, administrator can selectively elect to pre-cache groups of frequently used long running reports for the highest impact while allowing Intelligence Server to create caches on-the-fly, as required, for other reports. Alternatively, caching can be disabled for reports querying against Online Transactional Processing (OLTP) data sources. Intelligence Server statistics provide guidance on what should be cached, with information on most frequently used reports, reports with the largest final result sizes, and reports with the longest query execution times. Armed with this information, the administrator can make intelligent custom caching decisions.

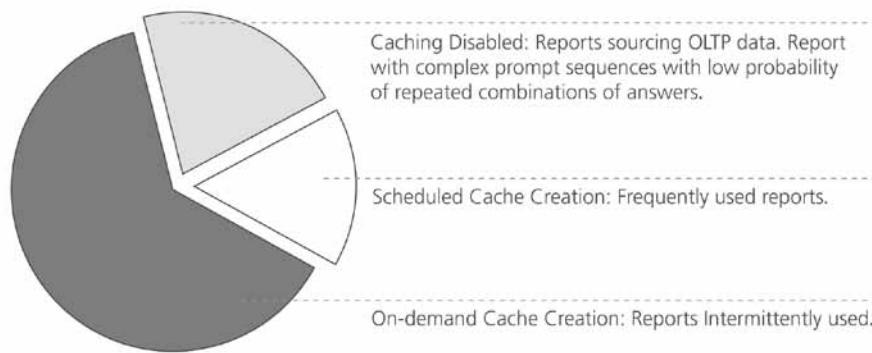


Figure 11-4 Multiple caching strategies applied to all reports ensure the highest performance without overwhelming the system.

11.3 IN-MEMORY CUBES

When assessing the speed of a BI system there's the concept of "computational distance". On any BI system, you take raw data at the very bottom, at the transaction level in the database, and use varying technologies to analyze and transform this data from its very raw state up to the finished answer that the user needs. At every step along the way work is done, either on the database, in the network, in the BI platform, or in the browser. This concept is called computational distance: What is the processing time from the base data all the way to a finished report that gets delivered to the user?

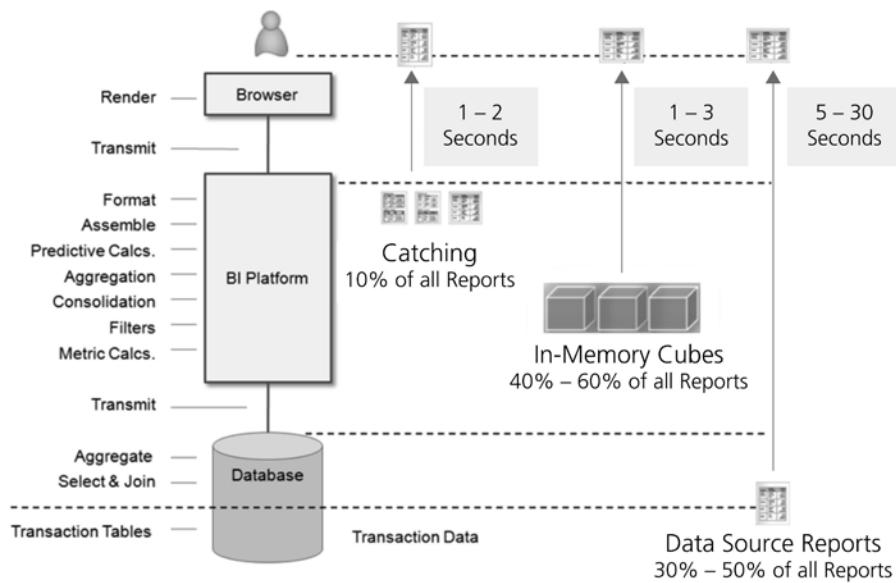


Figure 11-5 In-Memory Cubes build on the performance of caching, but expand its usability to 40%-60% of all reports.

According to our research this whole process takes an average of about 40 seconds for a classic system, not optimized but maybe with a very fast database. In fact, 40 seconds is a very common response time for an average query on many BI systems. To reduce this computational distance, some steps need to be cut out of the computational path making it shorter. And one of the most obvious ways to make it shorter is to cache reports: run all the computations ahead of time so when a user asks for information they get it instantly because all the work has already been done; by pre-calculating the queries you've effectively reduced the computational distance, placing the report results closer to the user.

If possible all reports that every user could possibly need within any given day should be pre-calculated because then everybody would get optimal sub-second response time. Of course this is not feasible because not all reports are predictable and BI systems are also limited by a batch window. In fact, only about 10% of reports benefit from caching in typical BI applications.

Introduced with MicroStrategy 9, In-Memory Cubes are designed to overcome the limitations of caching, opening up the technology to a wider range of reports. In-Memory Cubes have more flexible query characteristics than caches, but because the Cubes reside in main memory, they have the performance characteristics of caches. Because In-Memory Cubes reduce the computational distance effectively they provide a consistent 1-5 second wait time to a range of 40%-60% of reports in a BI application. These superior performance characteristics make In-Memory BI the best option to improve performance significantly.

Performance acceleration for 40%-60% of reports still appears insufficient at first glance until In-Memory Cube technology is looked at from a holistic point of view. When looking at the distribution of reports by frequency of usage and average user wait time it is possible to characterize 4 different categories of reports.

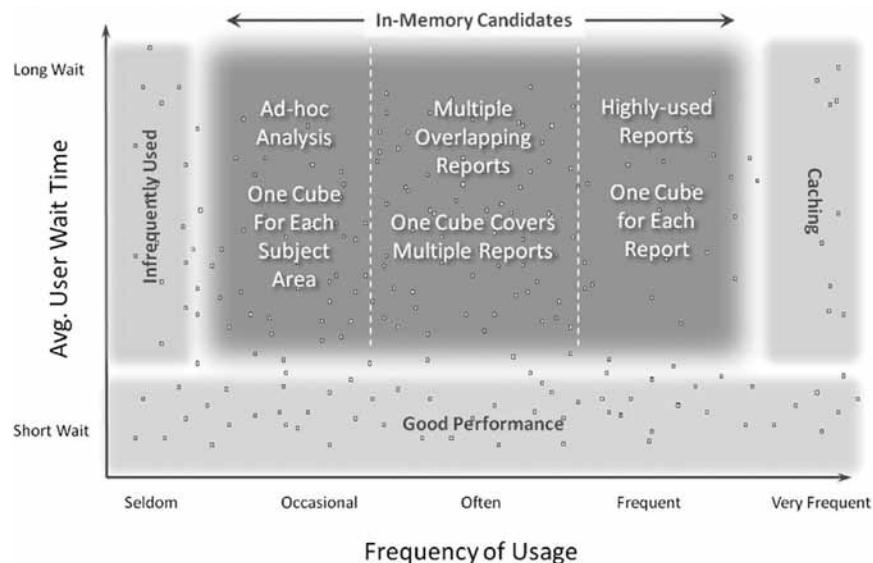


Figure 11-6 In-Memory Cubes are designed to accelerate frequently run reports with increasing levels of user wait time.

First, all the reports that already show good performance are not good candidates for in-memory cubes, because it is unlikely that their performance can be significantly improved. Similarly, reports that are only infrequently used are not good candidates for in-memory cubes, either. Thirdly, there is a category of reports that are well supported by caching. These should make use of caching as it offers better performance due to its shorter computational distance and therefore these reports are not good candidates for in-memory cubes, either. The remaining reports build a group of frequently run reports with increasing levels of user wait time as candidates for in-memory cubes. In most BI use cases this group makes up 40%-60% of all reports.

In-Memory Cubes Provide High Performance Analysis

In-Memory Cubes are not isolated data islands, but exist within the virtual data model that spans a BI application. In-Memory Cubes instantiate part of this data model into the Intelligence Server memory space and give users fast, sophisticated OLAP functionality while maintaining full access to the entire enterprise data warehouse. The data in the cubes is always up-to-date; if a cube is missing data that the user needs, or if the data has been updated in the data warehouse, the Intelligence Server will retrieve it from the warehouse. MicroStrategy In-Memory Cubes present highly interrelated subsets of data to end-users, making it simpler to perform quick analyses. Data proximity accelerates system performance while maintaining user self-service access to all corners of the enterprise data warehouse.

Full Access to Entire Data Warehouse

During regular use, the Intelligence Server automatically accesses In-Memory Cubes whenever relevant without any user intervention. As users analyze the data and expand the scope of their analysis requiring more data, the Intelligence Server automatically switches to access the data source. Accessing data from an Intelligent Cube or reaching through to the data source is seamless to users. This technology, called Dynamic Sourcing, provides a dramatic shift in user analysis. Without bounds to a limited aggregate of data, users on the MicroStrategy BI platform are free to navigate the entire data warehouse for comprehensive analysis. IT departments no longer have to build different cubes to serve different business requirements. Instead, most frequently requested data is bundled once, and users self-serve ad hoc requests.

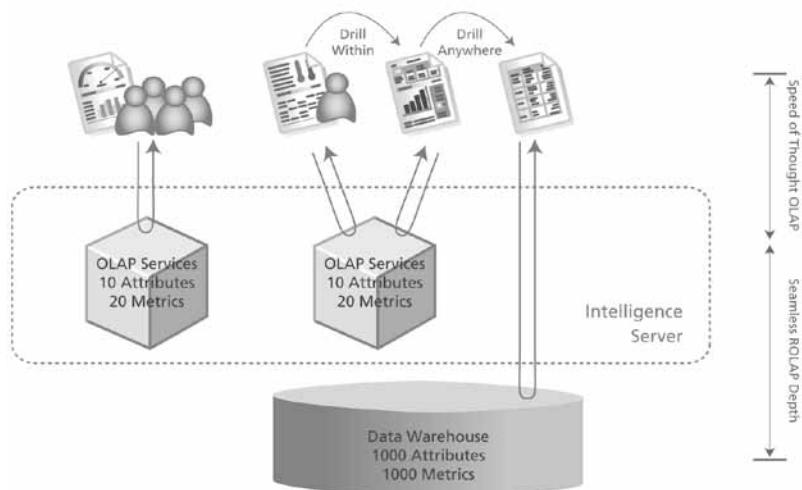


Figure 11-7 Speed-of-thought analysis is powered by In-Memory Cubes.

11.4 QUERY OPTIMIZATION

Database vendors invest heavily into building Relational Database Management Systems (RDBMS) that keep up with the proliferation of data. Each RDBMS provides many strategies to deal with this trend. The MicroStrategy BI platform, with its Relational Online Analytical Processing (ROLAP) architecture, uses the RDBMS for much of its processing, making the most of the RDBMS investment. No time-consuming and costly additional processing is needed before business users can access data. MicroStrategy incorporates the following RDBMS optimizations when generating SQL:

- Dynamic indices, statistics, and hints
- Partitioning, clustering, and parallel queries

- Volatile tables, derived tables, and common table expressions
- Materialized views and parameterized queries

Because of its unique ROLAP approach, MicroStrategy seamlessly works with all of these database-driven strategies for providing quick response time even with massive quantities of data. Any technical advances in the database are seamlessly accessible to the MicroStrategy platform. In addition, the Intelligence Server generates efficient SQL to minimize query response time and work in tandem with strategies employed by databases. These features are automatically applied for all users but can also be customized to meet unique scenarios. Examples of these features are:

- Multi-pass SQL
- Aggregate Awareness
- Very Large Database (VLDB) properties

Any scalable system must capitalize on the strength of each component of the application. VLDB properties, massively parallel syntax, and intelligent aggregate navigation ensure that the use of the data warehouse—the component in which most companies make the biggest investment—is maximized. Organizations can design and build applications with the assurance that MicroStrategy will continue to leverage each database's strengths to provide optimum performance and scalability.

Multi-Pass SQL

One of the key elements to providing analytical sophistication in business intelligence applications is MicroStrategy's ability to generate multi-pass SQL. Multi-pass SQL is required to answer analytical questions that cannot be answered with a single SQL query block. Examples of scenarios that require multi-pass SQL include:

- Set analysis
- Split metrics
- Calculation at different levels of aggregation

Sets allow the definition of complex selections and custom groupings of customers, products, suppliers, or other individual areas of business interest. For example, when data on customers, their recent purchases, and other characteristics related to their transactions is available, sets allow the targeting of customer groups based on specified criteria.



Figure 11-8 Set Analysis using custom groups and relationship filters is performed with multi-pass SQL.

The MicroStrategy BI platform enables much more complex analyses. For example, questions such as "Show me sales by region over the last six months, but only for customers who purchased one of the 5 most popular products in 2012, and for regions where vendors have the most reliable shipment times over all time," can be answered. With multi-pass SQL, when sets cannot be calculated together, MicroStrategy breaks them down into manageable components to achieve results quickly. Similarly, queries requiring calculations against different database tables with complex relationships cannot always be resolved with direct queries and require multiple steps. The Dynamic SQL Engine recognizes the structure of the tables, and generates correct SQL to ensure accurate data with fast performance.

Organizations often need to compare results between different strata. These calculations involve aggregating data into different levels for cross comparison. Individual performances are compared to total group performances to obtain insight into relative value. For example, stores contributing the highest revenues to their regions are identified as benchmarks, and all other stores are compared against them. When this comparison is done at the percent contribution level, no custom SQL coding is required; the analysis is automatically supported using multi-pass SQL.

Support for these scenarios, especially when combined together, provides a framework for answering significant analytic questions, and creating tremendous value to business users. MicroStrategy generates SQL that performs these multi-pass queries as efficiently as possible for each particular database.

Aggregate Awareness

The MicroStrategy BI platform features intelligent aggregate table navigation. Query performance in many data warehouses is enhanced through the use of aggregate tables or summary tables, which store pre-computed aggregate results. Report queries access information from a summarized set of data rather than the detail level data that would be stored in the fact table, often improving query performance by orders of magnitude. Administrators can build as many aggregate tables as are necessary and useful.

MicroStrategy has used an aggregate-aware SQL-generation engine since its products first became available in 1994. MicroStrategy transparently navigates aggregate tables, dynamically directing queries to summary tables when doing so is optimal, without the user having to specify use of the table when creating the query. Once these tables are included in the application, Intelligence Server automatically evaluates all the potential options for generating the results, and intelligently chooses the best aggregate tables for the query each time SQL is generated.

Very Large Database (VLDB) Parameters

Each major relational database has specific syntax to make use of its query optimization features. Small changes to the SQL can guide the database optimizer, and provide a dramatic improvement in query performance.

The MicroStrategy BI platform automatically ensures that each query leverages the high-end capabilities of the database through VLDB properties for the databases. VLDB properties are parameters that optimize the SQL generated for performance. Intelligence Server is automatically configured to use the right set of VLDB properties for each major database when that particular database is selected. Using these out-of-the-box VLDB properties, Intelligence Server adapts to the existing environment, and capitalizes on the inherent strengths and native optimizations of each database platform. Business users are not required to understand any usage rules within the business intelligence system or within the database to benefit from VLDB settings.

For example, databases that support derived tables can provide orders of magnitude increases in performance over generic SQL. Rather than generate generic SQL, Intelligence Server generates SQL that uses derived tables whenever possible. The following SQL sample of a report shows two different SQL statements

generated when the derived table switch is toggled. In this case, the response time was approximately 190 % faster when derived tables were used.

As each BI implementation is unique, the VLDB properties can be manually adjusted to suit a particular database schema, metric calculation, or report. Over 150 properties in 13 categories are customizable to produce highly efficient SQL queries.

Without Derived Tables	With Derived Tables
<pre> create table ZZMDOO nologging as select a12.QUARTER_ID QUARTER_ID, count(a11.SALES_DOC_ITEM_ID) WJXBF51 from F_SALES_DOC_ITEM a11 join L_TIME a12 on (a11.CREATE_ITEM_DATE = a12.DATE_ID) where a11.SALES_DOC_TYPE_ID in (15) group by a12.QUARTER_ID create table ZZMD01 nologging as select a12.QUARTER_ID QUARTER_ID, count(a11.SALES_DOC_ITEM_ID) WJXBF51 from F_SALES_DOC_ITEM a11 join L_TIME a12 on (a11.CREATE_ITEM_DATE = a12.DATE_ID) where (a11.SALES_DOC_TYPE_ID in (15)) and a11.SDOC_ITEM_STS_ID in (2)) group by a12.QUARTER_ID select pa11.QUARTER_ID QUARTER_ID, a13.QUARTER_DESC QUARTER_DESC, pa11.WJXBF51 WJXBF51, pa12.WJXBF51 WJXBF52 from ZZMDOO pa11 join ZZMD01 pa12 on (pa11.QUARTER_ID = pa12.QUARTER_ID) join L_CAL_QTR a13 on (pa11.QUARTER_ID = a13.QUARTER_ID) drop table ZZMDOO drop table ZZMD01 </pre>	<pre> select pa11.QUARTER_ID QUARTER_ID, a13.QUARTER_DESC QUARTER_DESC, pa11.WJXBF51 WJXBF51, pa12.WJXBF51 WJXBF52 from (select a12.QUARTER_ID QUARTER_ID, count(a11.SALES_DOC_ITEM_ID) WJXBF51 from F_SALES_DOC_ITEM a11 join L_TIME a12 on (a11.CREATE_ITEM_DATE = a12.DATE_ID) where a11.SALES_DOC_TYPE_ID in (15) group by a12.QUARTER_ID) pa11 join (select a12.QUARTER_ID QUARTER_ID, count(a11.SALES_DOC_ITEM_ID) WJXBF51 from F_SALES_DOC_ITEM a11 join L_TIME a12 on (a11.CREATE_ITEM_DATE = a12.DATE_ID) where (a11.SALES_DOC_TYPE_ID in (15)) and a11.SDOC_ITEM_STS_ID in (2)) group by a12.QUARTER_ID) pa12 on (pa11.QUARTER_ID = pa12.QUARTER_ID) join L_CAL_QTR a13 on (pa11.QUARTER_ID = a13.QUARTER_ID) </pre>

Figure 11-9 The same SQL query is generated differently when the “derived table” switch is turned on.

VLDB Property Levels

VLDB properties can be configured at different levels in the MicroStrategy BI environment. A system administrator can customize the global and application settings, while a report designer can tweak individual metrics, templates, and reports. In the following diagram, the VLDB properties set at the report level override the same properties set at higher levels.

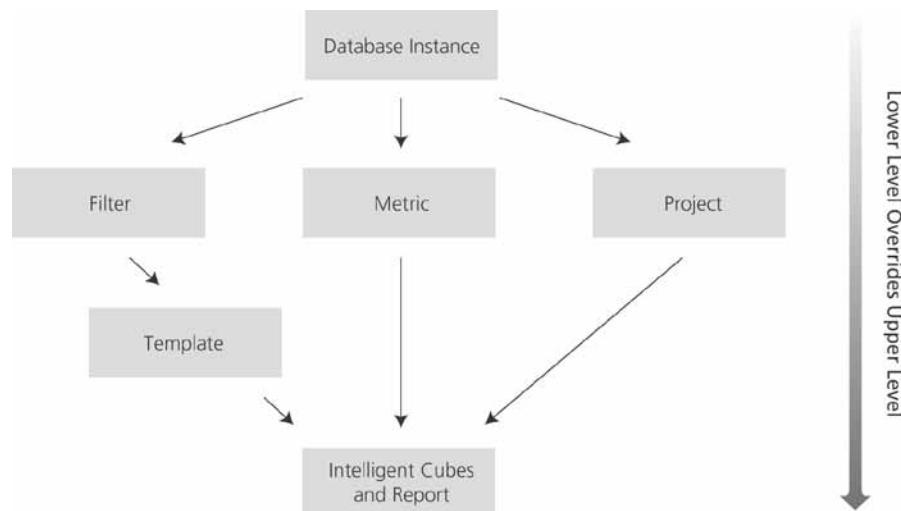


Figure 11-10 VLDB Properties Hierarchy determines which VLDB properties to apply to a query.

⁷See Appendix B for a list of VLDB properties available

Types of VLDB Properties

VLDB properties at each level are defined in a simple graphical interface in the Windows development environment. A SQL preview shows the effects of changing a setting. The VLDB properties for SQL generation fall mainly into the following types:

- Query optimization
- Fallback optimization
- Join guidance
- Parallel execution guidance
- Database tuning optimizations

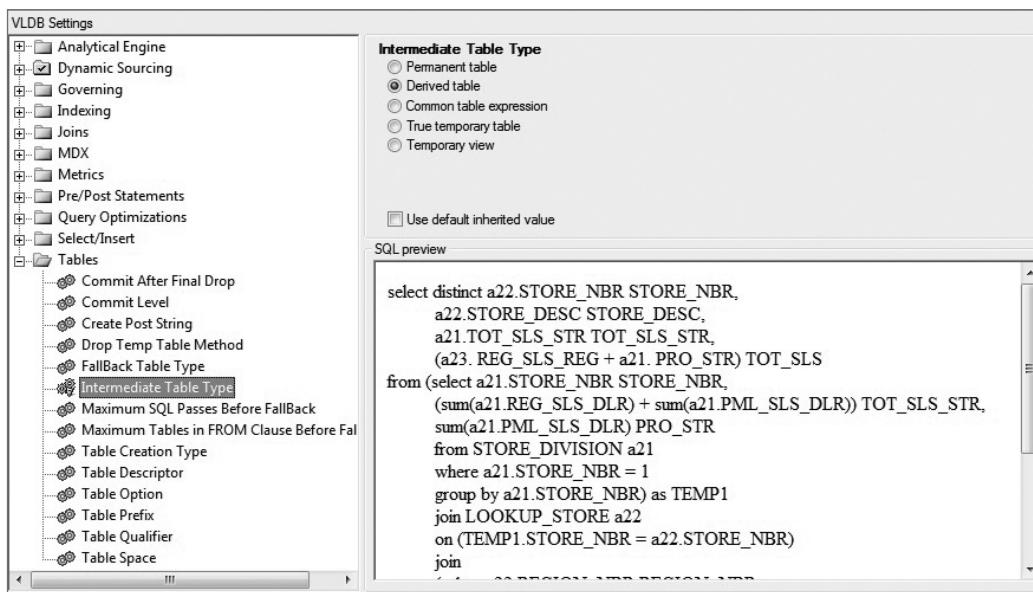


Figure 11-11 VLDB properties that can be set for a specific report.

SQL Query Optimization

Query optimization properties generate SQL for the best possible query performance by adjusting the Data Definition Language (DDL) and Data Manipulation Language (DML) automatically for each major database. Even then, VLDB properties can be customized to match unique reporting scenarios. For example, if using derived tables is not feasible, MicroStrategy can utilize Common Table Expressions instead. Similarly, if joins to attribute descriptors impede performance, MicroStrategy can delay these joins until the final SQL pass. The latter can reduce query execution times by 90% or more.

In addition, an experienced Report Designer has the opportunity to further improve query performance by orders of magnitude with custom Optimizer hints. For example, the Oracle optimizer uses hints that can be customized per report.

SQL Query Optimization Property	Description
SQL Hints	Provides guidance to the optimizer on how best to execute a query.
Intermediate Table Type	Provides guidance on how intermediate data is stored. Storage types include derived tables, common table expressions, volatile tables, and global temporary tables.
WHERE clause driving table	Provides guidance on which tables to use in WHERE clause qualifications. This setting is used as an optimization for databases that perform better when columns from smaller tables are used in the WHERE clause.
Additional Final Pass	Provides guidance on when to join back to lookup tables for attribute descriptions. This setting is used in situations where SQL performance can be enhanced by isolating the retrieval of metric calculations from attribute text columns.
Separate COUNT DISTINCT	Provides guidance on issuing COUNT DISTINCT. In certain environments, separating COUNT DISTINCT into isolated SQL statements improves performance by reducing concurrent resource requirements.
UNION Multiple Inserts	Provides guidance on use of UNION clauses rather than multiple separate intermediate tables with INSERT clauses. In certain environments, queries against partitioned tables using UNION clauses can improve performance.

Figure 11-12 Examples of SQL query optimization VLDB properties.

Fallback Optimization

In addition, MicroStrategy VLDB properties provide fallback mechanisms for query optimizations that are used as an alternative to the default setting. For example, if a particular query cannot be executed using derived tables, VLDB properties will reconfigure the generated SQL to alternate intermediate table syntax to ensure that the query returns correct results.

Fallback Optimization Property	Description
Sub-Query Type	Provides guidance on sub-queries syntax. Controls whether EXISTS IN statements or intermediate tables are used when performing sub-queries.
Fallback Table Type	Provides guidance on the intermediate table SQL syntax used. If the primary intermediate table type is unavailable in a database, either permanent or true temporary table syntax will be used.

Figure 11-13 Examples of fallback optimization VLDB properties.

Join Guidance

Correct joins are essential for accurate results and can also impact query performance. MicroStrategy automatically generates optimal SQL with the appropriate joins for data accuracy and query performance. Custom configuration of joins may be required for particular user requests. For example, the sequence in which tables are joined can impact query performance depending on the database. VLDB properties can adjust this sequence for maximum performance.

Join Guidance Property	Description
Lookup Table Join Order	Provides guidance on the sequence in which database lookup tables are joined. This setting is used as an optimization in scenarios in which a database optimizer may not be up-to-date.
From Clause Order	Provides guidance to the sequence in which database tables are used in the FROM clause of SQL. This setting is used as an optimization in scenarios in which a database optimizer may not be up-to-date.

Figure 11-14 Examples of join guidance VLDB properties.

Parallel Execution Guidance

As the volume of data in an enterprise data warehouse increases, partitioning the data provides significant improvements in query performance by distributing the workload between multiple resources in parallel. By dividing an SQL statement among multiple processes, a database can run the statement more quickly than if a single process ran it.

As part of this functionality, high end databases allow tweaks to be added into SQL for additional guidance. For example, when creating tables in a partitioned database, the partitioning key is used to determine the partition on which each row of data is stored. When performing join operations, MicroStrategy uses the partitioning key to improve performance.

Queries with intermediate tables may also benefit from indexes on these tables. In MicroStrategy, most intermediate tables will either be joined on the attribute columns back to fact tables or serve as temporary storage for metrics that will be joined with other intermediate tables to produce a final result. Creating an index on the attribute columns may improve performance, since those columns will usually be involved in a join later in the report SQL.

Parallel Execution Guidance Property	Description
Intermediate Table Index	Partitioning keys can be turned off altogether using the Intermediate Table Index VLDB property. When using this setting with a partitioned database, the database will distribute data according to the first column in the table.
Allow Index on Metric	By default, the partitioning key is created on all attribute ID columns that are used in the intermediate table. Fact columns can also be included in the partitioning key, using the Allow index on metric VLDB property. Depending on joins required, some queries may benefit from distributing data by the fact columns as well as the attribute ID columns.
Max Columns in Index	The application designer can control the size of the partitioning key through the Max columns in Index VLDB property.

Figure 11-15 Examples of parallel execution guidance VLDB Properties.

Database Tuning Optimizations

In order for a database to calculate the most efficient SQL execution plan, optimization engines require up-to-date information especially when new objects, such as intermediate tables, are created. Using VLDB settings, MicroStrategy Intelligence Server can insert complete or partial SQL statements throughout the dynamically generated SQL. For example, the Table Post Statement VLDB setting can be used to update optimizer statistics on new intermediate tables and indexes.

Tuning DB Optimization Engines VLDB Property	Description
Pre/Post Statements	Triggers updates to optimizer statistics when used with database-specific SQL syntax. For example: "Collect Statistics," and "Runstats."

Figure 11-16 Example of VLDB Properties to tune database optimizers.

String Insertion

Besides VLDB properties that automatically alter the overall structure of the SQL statements, MicroStrategy Intelligence Server also provides a multitude of manual entry VLDB settings. These VLDB properties control string syntax used in SQL queries generated by the Intelligence Server. A report designer can further optimize SQL for their specific environment using these string insertion settings. The following illustrates example locations for string insertion for Oracle. Each potential location is highlighted in bold.

```

[Report Pre Statement]
[Table Pre Statement]

CREATE [Table Qualifier] TABLE [Table Descriptor] [Table Prefix]<table_name>
[Table Option] [Table Space] AS
SELECT [SQL Hint] <column_expressions>
FROM <table_list>
WHERE <joins_and_filter_expressions>
GROUP BY <column_expressions>
[Insert Post Statement]

CREATE [Index Qualifier] INDEX [Index Prefix]<index_name> ON
<column_expressions> [Index Post String]

[Table PostStatement]

SELECT [SQL Hint] <column_expressions>
FROM <table_list>
WHERE <joins_and_filter_expressions>
GROUP BY <column_expressions>

[Report PostStatement]

```

Figure 11-17 Example of string insertion VLDB locations.

Global Optimization

The last step in the SQL generation process is called Global Optimization. Once the SQL Engine has processed the entire report definition, it reviews whether any additional optimizations can be made. This step is “global” because it optimizes across the full scope of the report. This is different to all the other 100+ VLBD optimizations that only act locally – independent of other parts of the report definition. The implementation of Global Optimization addresses a spectrum of SQL optimizations:

- Eliminate unused SQL passes
- Reuse intermediate passes of SQL that are identical
- Combine multiple passes of SQL where the SELECT list is different
- Combine multiple passes of SQL where the WHERE and/or HAVING clauses are different

VLDB Properties Updates

When database vendors add new functionality to their databases, administrators can easily update the VLDB properties. MicroStrategy provides new VLDB property files, along with new releases when necessary, that reflect the added functionality. Administrators simply update their properties using the new file and they are ready to go.

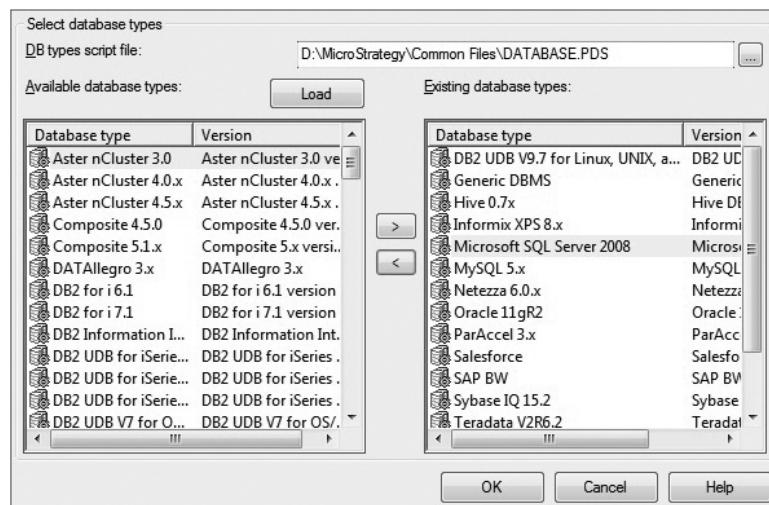


Figure 11-18 Upgrading VLDB properties via a simple dialog box.

11.5 DISTRIBUTABLE EXECUTION

As organizations move toward BI consolidation together with increased data warehouse sizes and more enterprise users, an enterprise BI system needs to be able to handle the increased load. In addition, organization with Service Level Agreements (SLA) requiring predictable levels of service need confidence in their BI system as it grows. These needs are met using distributable execution. MicroStrategy Web, MicroStrategy Mobile, and Intelligence Server are all cluster capable.

Multiple MicroStrategy Intelligence Servers configured in a cluster work cohesively as a single logical unit, and each node services end-users seamlessly. Workload can be distributed across nodes in a cluster, and adding new nodes is straightforward. Resources available to different projects can be dynamically adjusted based on real-time workload. MicroStrategy Intelligence Server clustering enables this via:

- Load Balancing
- Asymmetric Clustering

In addition, clustering allows enterprises to maximize their hardware utilization by starting with a small server implementation, and adding hardware incrementally as the number of end-users increases. As a result, the initial investment in hardware is always preserved, and performance is always excellent.

Load Balancing

Load Balancing refers to a process or algorithm that shares the load on a system among the available servers. Within the MicroStrategy BI platform, load balancing is performed using algorithms on the MicroStrategy Web server or MicroStrategy Mobile server that route new requests for user sessions to the Intelligence Server with the most available open sessions. Once a session is established, all requests from a connected user are directed to the same Intelligence Server. Load balancing ensures that response time is maintained as the number of end-users increases. The additional users are distributed across multiple servers which process jobs simultaneously.

Unlike other BI products where the load balancing algorithm cannot be customized, load balancing within an Intelligence Server can be customized by the load balance factor. With MicroStrategy Web and MicroStrategy Mobile, each node within an Intelligence Server cluster is assigned a work load weight; user connections are distributed to reflect the ratio of these weights. As a result, business needs or hardware considerations such as the following can be incorporated into the MicroStrategy BI platform.

- Upgrade Transition
- Mixed Clients Deployment
- Geographic Proximity

During an upgrade cycle, Intelligence Server nodes are upgraded in a rolling window to minimize risk and avoid downtime. After a server is upgraded, the load balancing factor can be used to give the upgraded machine a heavier load.

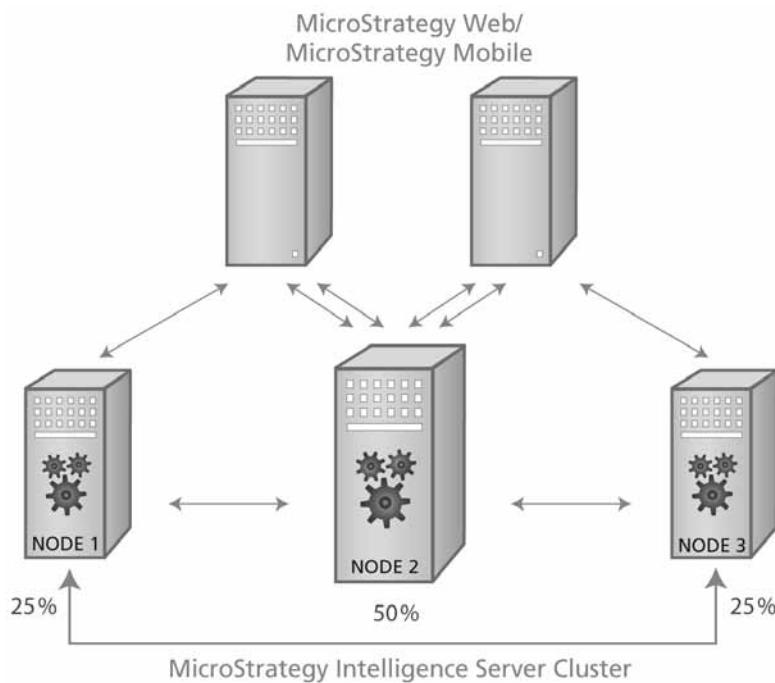


Figure 11-19 Load balancing across Intelligence Server nodes is tunable depending on each node's hardware resources.

An organization with a mix of Desktop Power users and Web end users can decide to dedicate an Intelligence Server to Desktop users due to their usage profiles. If there is a small number of desktop users, resources available on the dedicated node may not be used heavily, but can peak sporadically. The organization can choose to divert a small amount of Web user load onto this box to capitalize on the idle resource, and speed up end-user experience for the Web users.

An organization with a multi-location environment can choose to point each location to a particular Web server, and configure the load balancing factor of that MicroStrategy Web to favor the closest Intelligence Server node.

Asymmetric Clustering

MicroStrategy Intelligence Server provides Asymmetric Clustering, in which each Intelligence Server node within a cluster can host a different set of projects. This feature enables further opportunities to optimize the overall BI application, better utilizing available resources, and increasing performance. Asymmetric clustering in MicroStrategy Intelligence server allows:

- Further optimized resource allocation
- Increased agility to react to usage fluctuations
- Enhanced Failover and Fallback Logic

Additional load balancing configurations are available for MicroStrategy Mobile, Web, and Office.

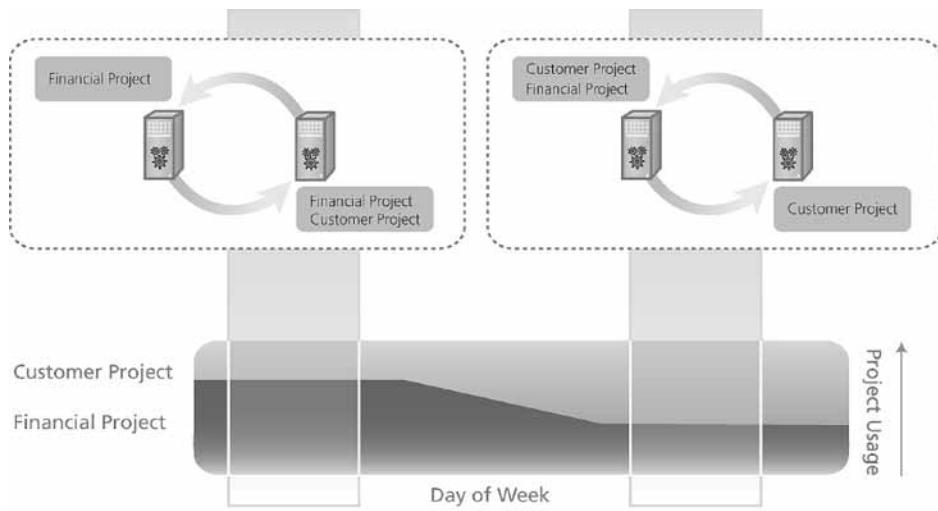


Figure 11-20 Asymmetric clustering in Intelligence Server spreads the load of the BI applications across nodes.

11.6 64-BIT BUSINESS INTELLIGENCE

Continuing to spearhead business intelligence system scalability and performance, MicroStrategy began development work towards true 64-bit technology in 2001. Part of this challenge was to incorporate a new technology framework compatible with all operating systems on any architecture.

Envisioning the boost in scalability and performance, MicroStrategy started incorporating parts of this technology framework in MicroStrategy offerings in 2002. Development in this direction continued until its completion and the release of MicroStrategy Universal in 2004. With a single code base, there is functional equivalence across all platforms. All platforms can be released simultaneously, and businesses choosing to leap to 64-bit MicroStrategy Business Intelligence technology can simply switch out the software supporting their existing implementations.

Leaping from 32-bit technology to 64-bit technology provides a gargantuan capacity increase for a business intelligence system. The more memory the business intelligence system can track, the larger the data sets it can manage and hold, and the faster it can service users.

Many business intelligence vendors only offer 32-bit solutions where the addressable memory is limited to 4 gigabytes. Others may execute on a 64-bit environment but may not be optimized for a 64-bit environment, and still be limited to 4 gigabytes of addressable memory. A true 64-bit solution can address up to 16 exabytes of memory. As a result of being a true 64-bit technology, MicroStrategy BI systems can support:

- Increased number of users
- Increased performance through increased cache capacity
- Richer BI applications through increased metadata capacity
- Reduced numbers of servers

True 64-bit MicroStrategy BI technology seamlessly scales with the growth in BI applications, users, and data sizes while delivering ultra-high performance.

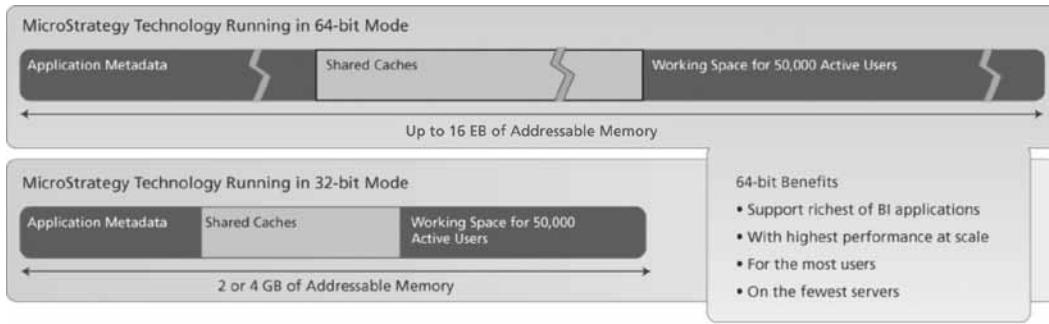


Figure 11-21 64-bit MicroStrategy technology supercharges BI allowing more users, applications, and caches.

More Users for Enterprise Wide Reporting and Analysis

With MicroStrategy 64-bit technology, far more users are able to tap into the largest databases, and obtain answers to their reporting and analysis inquiries. Information collected throughout business operations becomes available to the entire enterprise instead of select groups of individuals. Critical and active analysis is now available throughout the organization.

More Sophisticated and Comprehensive Analysis with Complete Data Warehouse Coverage

With MicroStrategy 64-bit technology, more capacity provides the opportunity to perform multi-layered, multi-level complex analysis on the entire data warehouse beyond the bounds of the database by business users. Analysis is no longer bound by 32-bit limits. BI systems have more capacity to match the growth of enterprise data warehouses, and to cover all aspects of information gathered and stored inside. More sophisticated questions can be answered to shed more comprehensive insight into the entire business operation.

Reduced Number of Servers Drives Enterprise Standardization

With increased capacity comes the opportunity to consolidate. A single BI system can support more users with better performance and richer applications. The management of BI systems is simplified with a reduced number of servers. MicroStrategy 64-bit technology provides a new level in support of an organization's BI standardization efforts.

11.7 MULTI-THREADED PROCESSING

Compared to other BI products, MicroStrategy Intelligence Server is a fully multi-threaded application that consists of a group of well-defined components. Each component performs specific tasks, and each task is performed by multiple processing threads. This approach, called multi-threaded processing, leverages all available processors in a system, giving it vertical scalability. According to Gartner Research:

"This technique enables an application to support many users simultaneously with a single image of the software. Instead of executing a separate copy of the application for each user requesting services, a single instance of the application spawns "threads" or sub-processes, as needed, to support concurrent users. This imposes less of a load on the server, and does not require the initial startup time for each new instance of the application. Additionally, threads can take advantage of multiple processors, if they are available, further enhancing scalability. Developing multi-threaded code is difficult, especially in the debugging stage. Consequently, many BI software providers do not support multithreading, or they offer only limited support."

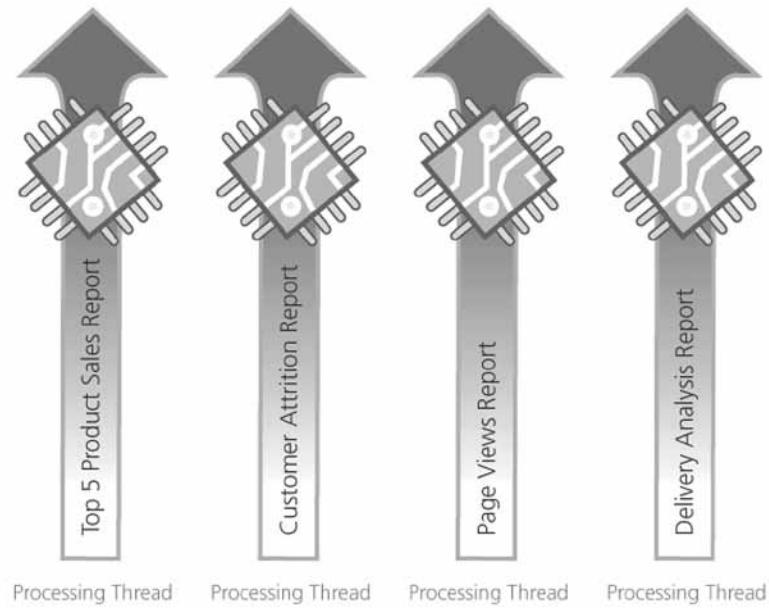


Figure 11-22 Multi-threaded processing concurrently executing reports on multiple processors.

The following are examples of multi-threaded components within MicroStrategy Intelligence Server. Each component is assigned a certain number of threads. The higher the number of threads allocated to a component, the more capacity it has.

- Object Browsing Engine
- Resolution Engine
- Command Engine
- Dynamic SQL Engine
- Dynamic MDX Engine
- Query Engine
- Formatting Engine
- Analytical Engine

The demand for individual components will vary depending on the numbers and types of job requests for report creation, report execution, and data manipulation being processed. For example, when many users request different queries simultaneously, the Dynamic SQL Engine, being one common component, experiences a load increase. When the report results have been retrieved from the data source, processing shifts to the analytical and formatting engines.

To prevent a bottleneck in any one of these components, Intelligence Server employs an advanced automatic resource-balancing scheme. This consists of dynamically allocating resources for each component based on its immediate needs. In the above scenario, the Intelligence Server will initially allocate more threads to the Dynamic SQL Engine. When the Analytical and Formatting Engines take over, Intelligence Server dynamically increases the number of threads for these components while reducing the number of threads for the Dynamic SQL Engine. This adaptive balancing of internal capacity ensures effective use of available resources, and allows the MicroStrategy BI platform to scale while maintaining excellent performance.

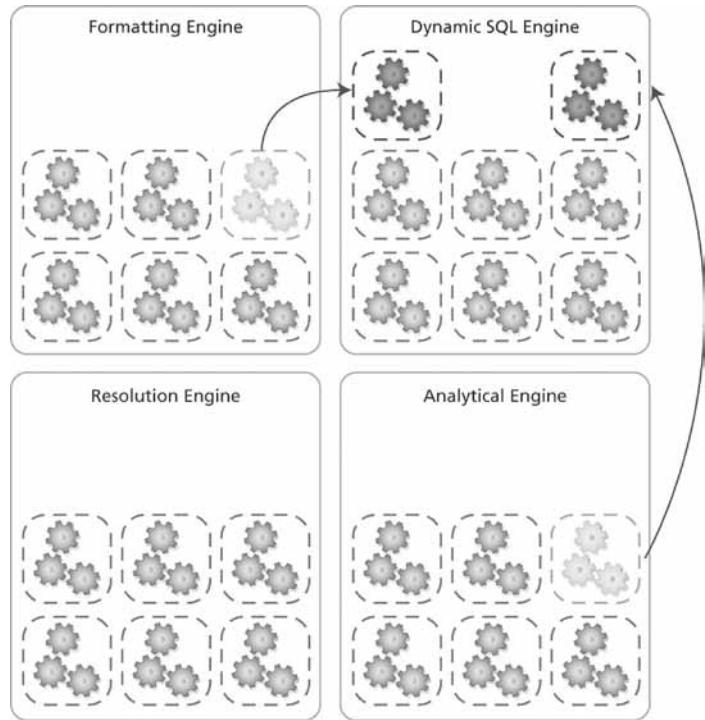


Figure 11-23 Distributing processing threads between processing engines automatically removes performance bottleneck.

11.8 EFFICIENT COMMUNICATIONS

The MicroStrategy BI architecture is designed to minimize network traffic between the different components within the architecture. While each component handles different parts of the user query, coordination, and communication between these components are required.

Network traffic from the data sources and metadata repository to the Intelligence Server, coordination between nodes of an Intelligence Server cluster, and communications between client interfaces and the Intelligence Server are network efficient.

Interacting with Data Sources

When Intelligence Server interacts with data sources, only the requested data is brought back, and as much aggregation as possible is performed at the data source. When data needs to be written back to an RDBMS, parameterized queries are used to reduce the net amount of data transferred. When querying a large attribute element lookup table, only the rows that are needed are retrieved and cached. When accessing business definitions within the Metadata Repository, Intelligence Server caches the most frequently requested definitions to reduce traffic.

Report Snapshot Delivery

With other BI products, when reports containing thousands of rows are returned to the client, the transmission process can slow down or be limited by the volume of network traffic. Even as bandwidth becomes cheaper, optimizing data transmissions remains important as the data still requires processing for client rendering. Just as search engines results typically display 20 hits per page, despite finding 30,000 hits

or more, the MicroStrategy BI platform returns results to the end user in 20-row snapshots. This number is configurable by the end user by adjusting the incremental fetch. Users have the ability to click “Next,” or select a page number, to view the next batch of results.

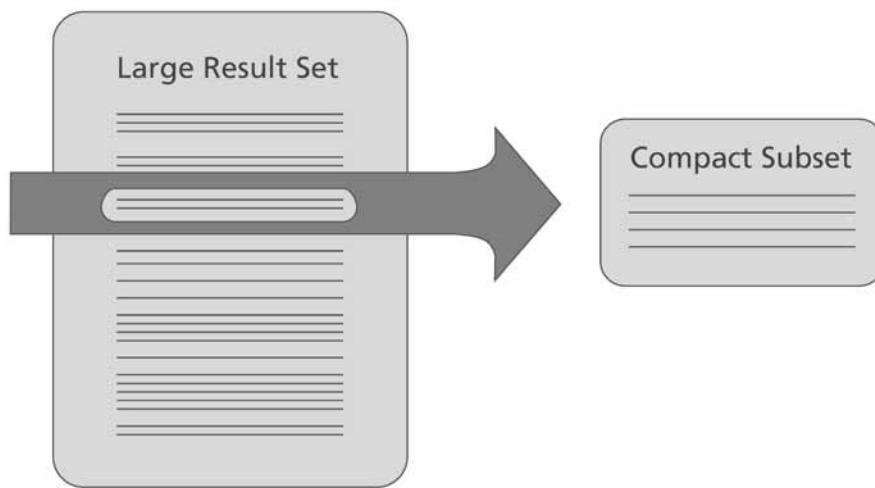


Figure 11-24 Report snapshot delivery incrementally fetches needed subsets of results to users.

Report snapshot delivery minimizes network traffic while maximizing performance as the amount of data processing on client machines is minimal.

Efficient Report Manipulation with Drilling

Drilling anywhere allows users to simultaneously navigate through the entire data warehouse and perform train-of-thought analysis. Other BI products require users to retrieve all items, all regions, and all months for drilling. Instead of this approach, which increases network traffic and poses data volume limitations, the MicroStrategy BI platform drill anywhere feature allows users to specify the specific slice of data for drilling. For example, when a user wants to drill from a specific item to geographic information, only the geographic information for that particular item is retrieved. In addition, drill paths can be customized per report, giving guidance to end-users and reducing the number of errant jobs.

Result Set Transmitted in Binary Format

To reduce the total data volume transmitted to client devices MicroStrategy Intelligence Server uses a compressed binary format when sending data to a client computer. By doing so the data volume is reduced by a factor of 1:5 to 1:10 compared to a standard XML representation of this data. Mobile client devices, such as Apple iPads, Apple iPhones, or Android phones, are primary beneficiaries from this technology because there are especially sensitive to data volume due to their limited on-board memory (RAM). Also, the report data is sent in binary format when data display is requested in Flash format.

Asynchronous JavaScript and XML (AJAX)

MicroStrategy’s web interface employs Asynchronous JavaScript and XML (AJAX) technology. This web based technology allows a web application to send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. MicroStrategy Web uses this technology to speed up page display by only sending the data to the web browser that needs to be

displayed and is requested by a user. For example, additional data is loaded asynchronously from the server when requested by the user to:

- Display Menu structures
- Answer Prompts for report and document execution
- Load and display interactive dialogs (e.g. Subscription or Send Now dialog)

11.9 PERFORMANCE TUNING

The MicroStrategy Platform provides you with a large number of settings to tune your BI system. It is important to note that each environment is different and therefore, there is not a single standard combination of settings that will work on all systems.

Every BI System is different and is with different user loads, databases, operating systems and query loads. MicroStrategy provides 200+ configuration settings to tune the BI system for these input parameters producing a highly tuned BI system with high speed and throughput as an output.

These settings can be divided into three distinct categories. The first category is called User Management settings. This category contains the settings that are available to manage user behavior. Users use resources from the Intelligence Server and sometimes may request more than necessary resources. Since resources inherently could be limited by the hardware, it is necessary to monitor and control user behavior so that enough resources are available for other users.

The second category is called Resource Management. The main resource to manage is memory. Even though memory in a 64-bit system can be plentiful these days, it is important to manage because unmanaged memory tends to be used so inefficiently that it can have a performance impact.

The last category is Workload management. This category consists of settings used to govern resources consumed by user requests, also called jobs, to prevent overloading of the Intelligence Server and the database with too many requests.

User Management Settings		Resource (Memory) Management Setting		Workload Management Setting	
2500	Max Number User Sessions	128GB	Max RAM for In-Memory Cubes	3-6-1	Job Prioritization
10MM	Max Number of Jobs	64GB	Max RAM for Working Set	20	Database Connection Threads
10K	Max Number of Interactive Jobs	1GB	Max RAM for SQL Generation	OFF	Enable Complete Logging
1000	Max Number of Scheduled Jobs	16GB	Max RAM for Report Caches	ON	Collect Detail Statistics
100	Max Number Concurrent User Sessions	16GB	Max RAM for Document Caches	12	Expire caches (hours)
40	Max Number Jobs per User	4GB	Max RAM for Object Caches	OFF	Load Caches on Startup
20	Max Number Scheduled Jobs per User	4GB	Max RAM for Element Caches	SM	Max Rows – Reports
40	Max Number interactive Jobs per User	70MB	Max RAM for Exporting to Excel	100K	Max Elements
1000	Max Number History List Subscriptions per User	50MB	Max RAM for Exporting to PDF	10M	Max Rows – In-Memory Cube
200	Max Number Mobile Subscriptions per User			10	Max Warehouse Execution Time (min)
100	Max Number Email Subscriptions per User			2	Intelligence Server Execution Time (min)
900	Max User Idle Time (sec)			OFF	Change Journaling
				ON	Job-Based Load Balancing
				ON	Session-Based Load Balancing
				ON	System Preference Monitoring

Figure 11-25 A list of the most important settings used to tune the MicroStrategy platform for performance.

11.10 PROVEN SCALABILITY AND PERFORMANCE

With all of the above technologies, the MicroStrategy BI platform leads the pack in both scalability and performance. By embedding the goals of building an architecture with world class scalability and

performance into the design process, MicroStrategy sets the standard for these characteristics. With each user query, multiple complementary technologies work hand-in-hand to boost performance, and reduce total response time. By servicing users in the most expeditious manner, resources remain available for other users.⁸

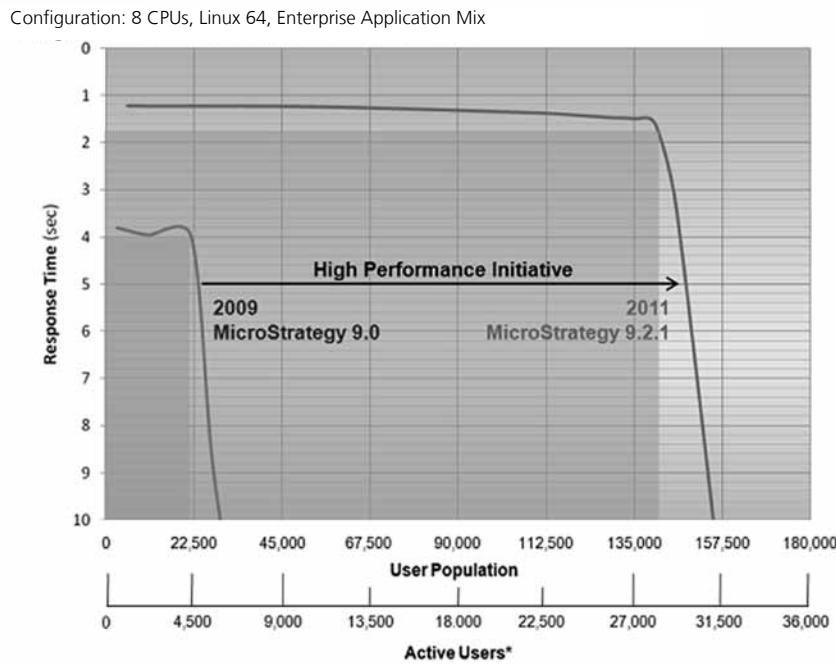


Figure 11-26 Performance improvements in the MicroStrategy software improve both the speed and the capacity of existing BI applications.

Large numbers of customers using the MicroStrategy BI platform are able to scale their BI deployments using the MicroStrategy BI platform both in terms of the number of users, and the amount of data accessible to them. Proven scalability and performance are hallmarks of the MicroStrategy BI platform.

11.11 SUMMARY

Through its design for both speed and scale, the MicroStrategy BI platform has emerged as the clear performance leader in the BI space. Achieving this performance is not a simple feat, but requires a long set of technologies to work together. Optimal query speed requires complex caching algorithms to work together with state-of-the-art In-Memory technology, supported by a best-in-class SQL generation engine containing a long list of query optimizations. Delivering this performance at scale is possible with careful server tuning optimizing the use of multi-threaded processing and distributed execution. Continuing improvements and innovations allow pushing the boundaries of what once was thought possible and maintain optimal performance in changing environment with challenging requirements.

Successful BI applications accelerate user adoption and enhance productivity, resulting in demand for more users, data, and reports. MicroStrategy provides the ability to adapt quickly to constant changes and evolve along with business requirements. MicroStrategy Intelligence Server has been proven in real-world scenarios to deliver the highest performance at scale with the fewest servers and minimum IT overhead.

⁸Active users are defined as people who perform their BI interaction during the peak hour. A good rule-of-thumb for converting active users to total user population is to assume that 10-20% of the total population is active during the peak hour.

CLOSED-LOOP BUSINESS INTELLIGENCE

Business intelligence reports, dashboards, and analyses provide the foundation for effective, data-driven decision making. One requirement in many business intelligence applications is for managers, analysts, and executives to take immediate action on the data presented. For example, a report showing a low inventory of popular products may contain a mechanism to automatically create purchase orders for these items directly from the BI report. This eliminates the need to manually access other IT applications, or process telephonic or paper requests. Closed-loop business intelligence is this interaction with BI reports and other systems, specifically to write data back to any target IT system, and to execute any process necessary to complete a desired action. This closes the loop between the BI system and the operational systems, turning insight into action.

Closed-loop BI applications vary widely from industry to industry, and within horizontal applications, such as sales, finance, and human resources. Closed-loop BI applications have a set of common requirements that satisfy a broad set of business demands:

- Reach every business user regardless of location using open, Web and Mobile technologies
- Connect to any number or type of database-driven applications
- Perform common database functions to modify/update, insert, or delete data
- Act on business information anytime, anywhere by approving requests, submitting orders, changing plans, and capturing instant feedback

12.1 CLOSED-LOOP APPLICATIONS REDUCE THE TIME TO ACT

Closed-loop BI applications implemented with the MicroStrategy BI platform reduce the time required to act on business insight, and allow companies to standardize internal processes and procedures like the stock reordering process described above. Furthermore, because business users have this closed-loop BI capability directly within MicroStrategy Web reports, there is no additional training required. The MicroStrategy BI platform implements closed-loop BI using:

- MicroStrategy Transaction Services
- Dynamic Datamarts

With MicroStrategy Transaction Services, users can embed transactional capabilities in Mobile devices – tablets and smartphones – and dashboards accessible through a Web browser. Users interact with data through a variety of data input controls such as sliders, calendars, switches or typing information directly into text fields. In all cases, MicroStrategy's Report Services documents are the interface that initiates the closed-loop interactions between the BI application, and any other IT system such as an enterprise resource planning (ERP), customer relationship management (CRM), supply chain management (SCM), data mining, or business intelligence application.

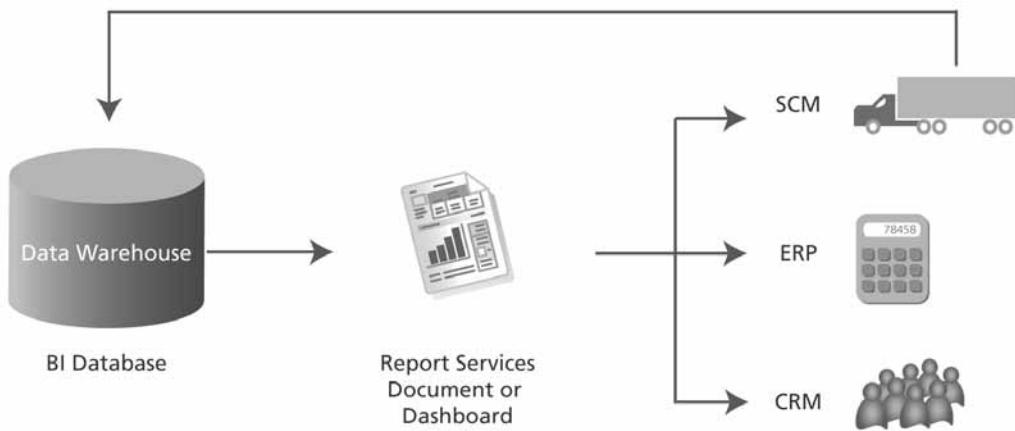


Figure 12-1 A closed-loop BI application turns insight into action by integrating with operational systems, such as SCM, ERP, or CRM applications.

12.2 CLOSED-LOOP BI USING MICROSTRATEGY TRANSACTION SERVICES

MicroStrategy Transaction Services provides write-back capabilities from Report Services documents. The predominant use case is implementing Mobile BI Apps; however it also works within Web-based DHTML documents. MicroStrategy Transaction Services connects to backend transaction systems, initiates a secure connection and writes-back to ERP and other operational systems via web services using XQuery, and to relational databases via SQL queries.

MicroStrategy Transaction Services provides users the flexibility of designing freeform transaction reports and hand-picking the data to perform each transaction on. MicroStrategy Transaction Services allows users to perform any of the following three actions: modify/update data, insert new information, or delete existing records.

From Insight to Action

Transactional mobile apps empower individuals and organizations to take immediate action. Executives, the sales force, and field employees can enter data, approve requests, adjust forecasts, add comments, upload photos, scan product SKUs, and much more, without ever needing to return to a desk. In an instant, these actions are sent to transaction systems, triggering business processes, resource reallocation, and accelerating business activity.



Figure 12-2 MicroStrategy Transaction Services allows users to take action anytime, anywhere.

MicroStrategy Transaction Services Architecture

MicroStrategy Transaction Services is an add-on component to the Intelligence Server, just like OLAP Services or Report Services. During installation, it displays as an option under the MicroStrategy Intelligence Server product. The Transaction Services developer interfaces are available in the FreeForm Query Editor in MicroStrategy Desktop and in the Document Editor in MicroStrategy Web. End users access transactional functionality in a document in the Express Mode in MicroStrategy Web and in apps in MicroStrategy Mobile.

Impact of Transaction Services on the BI System

In a traditional business setting, users capture data using the transactional OLTP (Online Transaction Processing). The OLAP (Online Analytical Processing) system is used to report on the captured data after the data is loaded from the operational systems into a data warehouse or datamart. The OLTP and OLAP systems are typically implemented independent of each other, use different end-user interfaces, and target different audiences. The common factor between the two systems is the actual data that is being written and read. Data is transferred from the OLTP system to the OLAP system using the ETL (Extract, Transform, and Load) process.

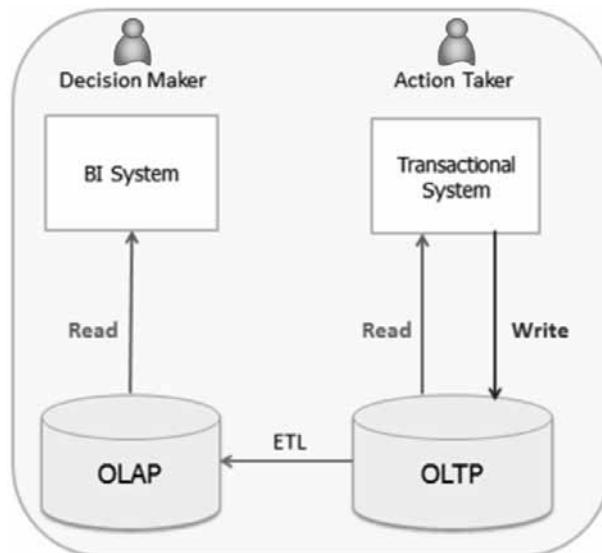


Figure 12-3 Traditional disconnected BI and Transactional System Architecture.

MicroStrategy BI Platform enables users to gain insight by querying your OLAP system while empowering them to take immediate action against the OLTP system. Users can accomplish both goals by creating a single application with a unified user interface—a MicroStrategy Mobile app or Web-based Report Services document powered by MicroStrategy Transaction Services.

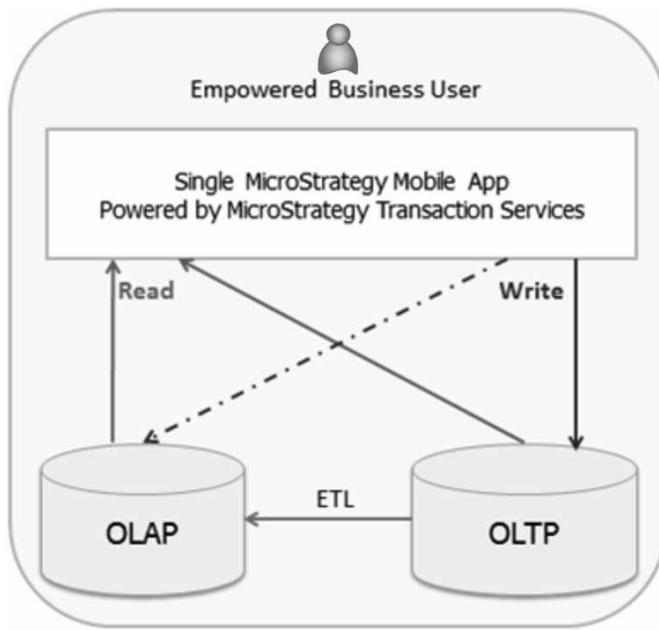


Figure 12-4 A fully integrated BI and Transactional System Architecture provided by the MicroStrategy Platform.

12.3 ENABLING TRANSACTIONAL FUNCTIONALITY IN MICROSTRATEGY

Adding transactional functionality involves creating query reports that retrieve data, transaction-enabled documents and dashboards that expose transactional functionality to the end users, and transaction reports that write back to the back-end systems.

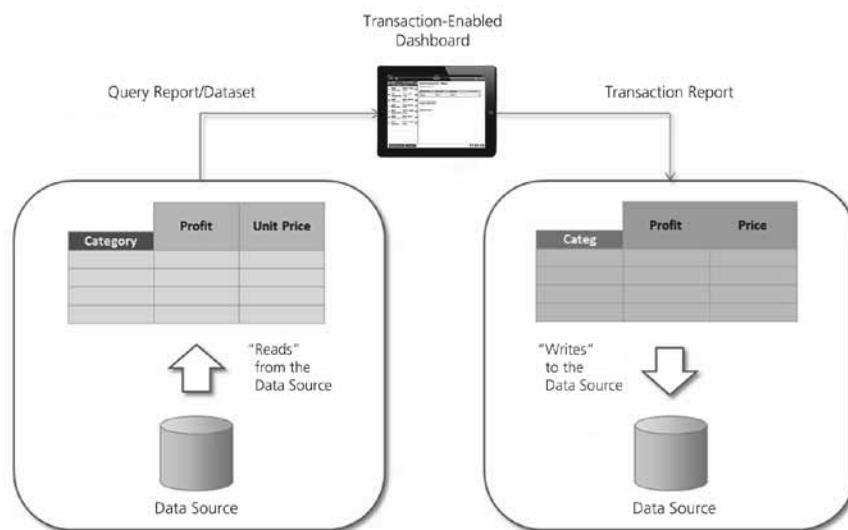


Figure 12-5 MicroStrategy Transaction Services uses Query reports, Transaction documents, and Transaction reports to implement write-back apps.

Transaction Reports Interact with the Data Sources

A transaction report is a new metadata object used to manage writing data. It's similar to a FreeForm SQL or XQuery report and can only be defined in the Freeform Query Editor in MicroStrategy Desktop. After a

transaction report is saved, the transaction report is associated with a grid\graph or a set of fields in a panel stack in a Report Services document. This Report Services document is then executed by clicking or tapping an action selector button/link, triggering the transaction. Transaction reports are created in a similar manner to FreeForm SQL or XQuery reports, however there are several differences:

- In addition to a standard SQL or XQuery statement, the query must be surrounded by ‘Begin Transaction’ and ‘End Transaction’ tags.
- Instead of mapping SQL or XQuery columns to output parameters, users map them to input objects. Input objects provide a link between the data that users enter and the query that updates operational system.
- Transaction reports do not return or display any data. A transaction report, however, requires at least one output object to be included on the report template:

Input Objects Map Input Data with the Transaction Report

Instead of hard-coding the input values in a SQL or XQuery statement, users define placeholders or variables—either attribute forms or metrics. When a transaction document is executed, the end user provides values for the input objects by manipulating sliders, steppers, calendars, or typing in the text boxes. When the end user submits the transaction after making changes, the input object placeholders are automatically replaced with the user’s entries.

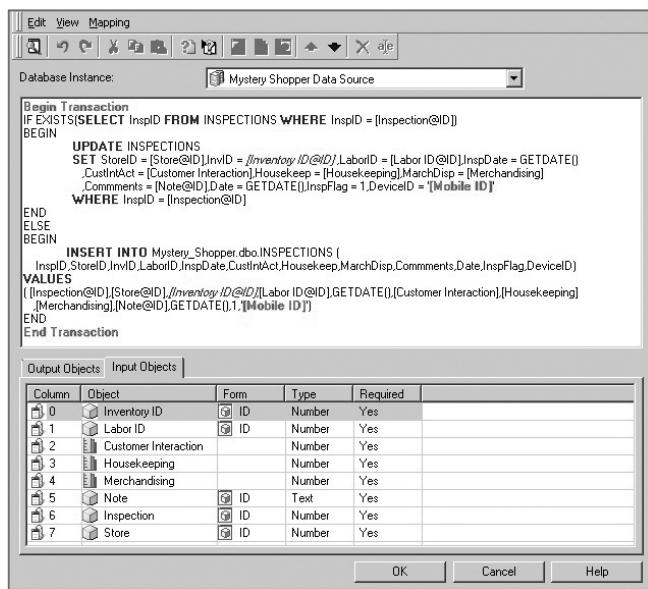


Figure 12-6 Input Objects are defined within the FreeForm SQL/XQuery report editor.

Query Reports Populate the Transaction Dashboards and Documents

A query report is a regular report used to retrieve data from the data source so that it can be modified through the execution of the transaction. This report can be a Grid report, Intelligent Cube, View Report or a Freeform Query report. An important note to make about the query report is that report caches must be disabled in order to extract the latest data from the data sources.

Transaction Dashboards and Documents Enable Data Input

Query and transaction reports are the building blocks of transactional applications in MicroStrategy. Depending on whether users want to manipulate (update or delete) existing data or insert new data to the transactional data source, different approaches are employed in building the document:

- Updating Existing Data

When updating existing data, the data must first be displayed in the document using a query report: either as fields or a grid in Web or Mobile or as an interactive grid widget in a Mobile app. Data cells that can be changed are marked as editable, which can be displayed as drop-down lists, text areas, toggle buttons, or any of the data input control styles available. Additional data that act as key identifiers and filter criteria must be included in query report and transaction report as needed. This data is marked as non-editable and can be hidden from the end user.

For example, users might want to approve or deny time off requests; this could be achieved by displaying queued requests in an Interactive Grid widget in a Mobile device where they would toggle the approve/deny buttons, type comments in a pop-up text area, and submit the transaction back to the data warehouse.

- Inserting New Data

When inserting new data, there is no need to display any existing data in the dashboard. Instead, a set of input fields (text boxes) is defined, placed on a panel stack or in a grid, and marked as editable. It is possible to display some data from a query report to orient the user, but it is not necessary.

For instance, users could enter inspection details for a store they are visiting. In this case, the name and address of the store will be presented to users (retrieved from a data source with a query report) along with a set of sliders, drop-down lists, and a text area to enter their observations.

Data Input Controls

Data entry needs to support the input of different types of data. MicroStrategy Transaction Services provides many data input controls to make data entry easy and accurate. Depending on the data type of the data, users can modify or enter data through a number of different data input control styles available in both Web browsers and mobile devices.



Figure 12-7 MicroStrategy Transaction Services offers a diverse and comprehensive set of Data Input Control styles.

In addition to interacting with sliders, steppers, calendars, drop-down lists, toggle buttons, text areas, and action buttons, Mobile users can also submit input with two transaction-enabled widgets:

- **Interactive Grid Widget** – Transactional functionality is enabled in an Interactive Grid widget on a mobile device by associating the widget's dataset with a transaction report and defining select grid objects as editable.
- **Photo Uploader Widget** – The Photo Uploader widget enables users to transfer images from a mobile device to the Mobile server. Existing photos from your photo library or new photos can be uploaded. This is achieved via the transaction report which records the name and location of each photo as a new record in a database table.

Some data input controls are unique to smartphones and tablets:

	Mobile Devices	Web Browsers
Text Field	X	X
Text Area	X	X
Calendar	X	X
Time Picker	X	X
List	X	X
Slider	X	X
Toggle/Switches	X	X
Steppers	X	X
Photo Uploader Widget	X	
GPS Co-ordinates	X	

Figure 12-8 Input controls available on mobile devices and in Web browsers.

Executing a Transaction

After the Report Services document or dashboard has been configured for Transaction Services, users need to decide whether to submit their changes back to the back-end system, or cancel the data input. This is achieved with an Action Selector button or link. When an action selector is defined, the selector's target is automatically assigned to the grid or a panel stack that holds the data input controls associated with a transaction. The text displayed in all the selectors is customizable. There are three types of action selectors:

- Submit – Submits data entered: insert, update, or delete data from the data source. A number of options are configurable when data is submitted:
 - Request user confirmation before submitting the transaction
 - Display a message after the transaction is submitted
 - Determine the subsequent action once input has been submitted:
 - Do nothing
 - Refresh current document
 - Run a new report or document
- Recalculate— Recalculates derived metrics, subtotals, and sort order, and updates other Analytical Engine calculations. Only the display is changed; no data is submitted to the Transaction Services report.
- Discard Changes— Discards any changes made by the user that are not yet submitted.

12.4 CLOSED-LOOP BI USING DYNAMIC DATAMARTS

Dynamic datamarts create datamarts, or tables of information, in a database using a report definition stored in the metadata. This eliminates the need to move large amounts of data to client machines and export the information. Datamarts are ideally suited for automatically storing BI data in relational databases for several reasons.

Creating and executing datamarts in MicroStrategy is straightforward and efficient. MicroStrategy Desktop defines the datamart table structure with a datamart report to distinguish it from other MicroStrategy report types. Once created, MicroStrategy Web or Desktop executes the datamart, populating it with the specified data.

- Datamarting uses MicroStrategy's full breadth of analytical richness including sophisticated filtering, time transformations, and predictive analysis to return the exact set of data required.
- Datamarting benefits from MicroStrategy's optimized SQL generation that ensures datamarts run efficiently and rapidly. Datamarting makes interfacing with the operational systems easy by incorporating pre-SQL and post-SQL capabilities.
- Dynamic datamarting stores the resultant datamart to any available database, including the data warehouse, SCM, ERP, or CRM databases. Once created, these applications use the datamart directly to automate business processes.
- Datamarts are efficient and straightforward to maintain and update. They have the flexibility to refresh entire datamarts or append new data to existing datamarts.

Entire Datasets are Integrated With Other Applications

Consider the scenario of a closed-loop application that combines BI and a CRM system. The CRM system contains a wealth of customer-specific data detailing demographic information, purchase history, personal preferences, and much more. MicroStrategy analyzes this data using sophisticated data mining algorithms to understand which customers will have the highest lifetime value, and the highest propensity to respond to a targeted marketing campaign. With the analysis complete, MicroStrategy creates a datamart directly in the CRM system. Using post-SQL scripting capabilities, MicroStrategy informs the CRM system that a new datamart is ready for the campaign management tools to initiate the targeted marketing campaign.

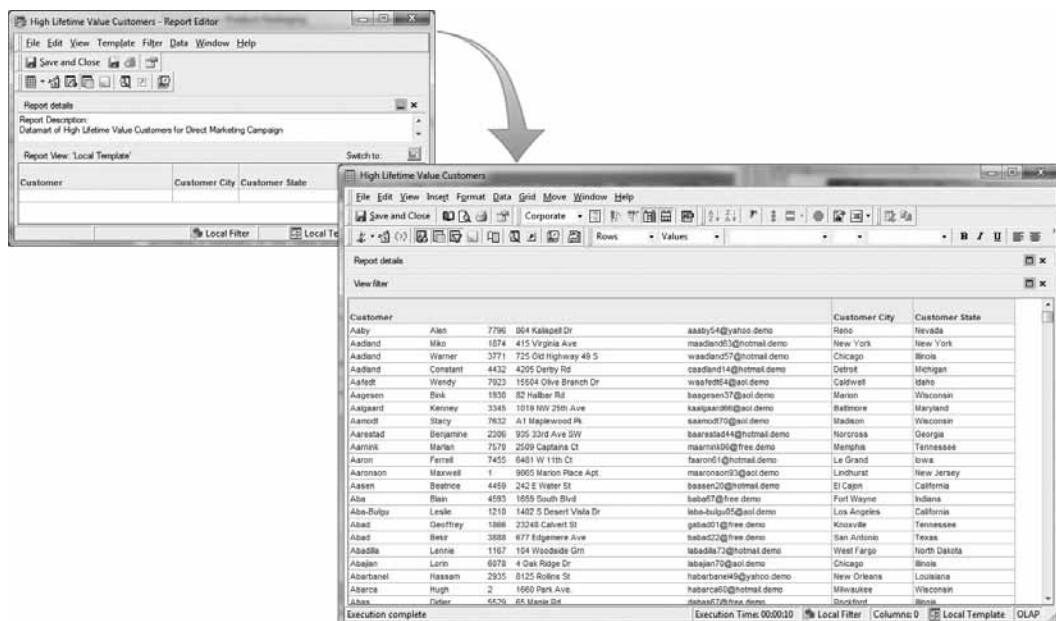


Figure 12-9 The datamart definition (top) lays out fields necessary to create and store the resulting datamart (bottom).

12.5 SUMMARY

Closed-loop business intelligence describes the ability to interact with data, specifically to write data to any target IT system, and to execute any process necessary to complete a desired action. This closes the loop between the business intelligence system and the operational systems, turning insight into action. Closed-loop BI applications reduce the time required to act on business insight. The MicroStrategy BI platform delivers two powerful options to cover the range of closed-loop BI applications using MicroStrategy Transaction Services and Dynamic Datamarts. Each one has specific strengths depending on the application's requirements.

MicroStrategy Transaction Services delivers the flexibility to implement complex closed-loop BI applications. MicroStrategy Transaction Services is outstanding for integration with operational systems. MicroStrategy Transaction Services connects to backend transaction systems, initiates a secure connection and writes-back to ERP and other operational systems via web services using XQuery, and to relational databases via SQL. Transactional mobile apps empower individuals and organizations to take immediate action. Executives, the sales force, and field employees can enter data, approve requests, adjust forecasts, add comments, upload photos, scan product SKUs, and much more, without ever needing to return to a desk.

Datamarting is particularly effective in generating large lists of data that get stored in a database as datamart tables, eliminating the need to move large amounts of data to client machines and export the information. This provides a carefully chosen set of data for use by other applications.

DATA MINING AND PREDICTIVE ANALYSIS

While providing sophisticated historic analysis of an organization's business has been the mainstay of BI applications, using the historical information to generate and distribute predictive insight is often a challenge for organizations. Organizations have invested in homegrown utilities and standalone data mining products in search of predictive analysis capabilities. These data mining tools incorporate statistical analysis, modeling techniques, machine learning, and database technologies to search for hidden patterns in historical data. Organizations develop predictive models from these hidden patterns, and use them to assess the likelihood of future events.

Despite the practical value of developing predictive models with data mining tools, adoption of predictive analysis by mainstream business users has been slow. Some commonly cited reasons for poor adoption of data mining by the business user community include:

- Data mining practitioners must be highly skilled in statistical analysis, modeling techniques, numerical analysis, database technologies, and process knowledge. These highly specialized skills are not the core competency of typical business users who need to use the predictive insights to make everyday decisions.
- Data mining tools often require the extraction of sample data from the enterprise data store for use with the data mining tools. Data samples are subject to pre- and post-processing so that the data mining tool can use them. The data extraction and subsequent pre- and post-processing often lead to islands of information, differences in business definitions, and the challenge of reconciling information between the BI system and data mining tools.
- Implementing data mining requires practitioners to learn and understand these specialized data mining products. Since the data mining products are separate products, distinct from the BI applications, they represent yet another set of tools that business users need to master.
- Perhaps the most common reason for poor adoption of data mining is the inability to distribute the predictive models in an easy-to-consume, highly formatted, and timely manner that is integrated with other BI reports and analyses.

Integration of data mining into the business intelligence platform has to satisfy these requirements:

- Offer various options for generating predictive analysis for everyday reporting.
- Make it unnecessary for business users who must consume the predictive insight to be highly skilled in the data mining technology and predictive algorithms.
- Provide a consistent, unified interface for the distribution and consumption of the predictive analysis that can be incorporated into scorecards, dashboards, enterprise reports, advanced analysis, and alerting.
- Re-use a corporation's existing investment in data mining products.

13.1 INTEGRATING DATA MINING INTO BUSINESS INTELLIGENCE

In general, data mining software assists and automates the process of building and training highly sophisticated data mining models, and applying these models to larger datasets. The data mining process involves the following steps:

1. Creating a predictive model from a data sample.

A sample of data with a known outcome is extracted from the enterprise data store and pre-processed for the development of the predictive model. Advanced statistical and mathematical models are used to identify the significant characteristics and trends using the pre-processed fields as inputs, resulting in a predictive model. Generally, only a small subset of the all characteristics and trends in the sample data is used in the model.

2. Training the model against the dataset and its known results.

The new predictive model is applied to additional data samples with known outcomes to validate whether the model is reasonably successful at predicting the known results. This gives a good indication of the accuracy of the model.

3. Applying the predictive model to a new dataset with an unknown outcome.

Once the predictive model is validated against the known data, it is used for scoring, which is defined as the application of a data mining model to forecast an outcome. For example, a data mining model that predicts the likelihood of a customer responding to a marketing campaign will generate a score for each customer that indicates his or her likelihood to respond. This score can be a simple binary result, such as a "Yes" or "No," or it could be a number indicating the propensity or confidence in that customer responding, say "97%."

As mentioned earlier, the "Create-Train-Apply" process is typically the domain of the statistician or the data mining analyst. A solid understanding of data mining concepts, statistical concepts, techniques, and data mining tools is necessary in the "Create" and "Train" steps. Applying the predictive model requires less expertise, and is available for all business users.

Scoring Data for a BI Application

There are three main approaches to integrating predictive insight into a BI application:

1. Data mining tool scores the database.

The data mining tool scores the records in a batch process, and saves them as columns in database tables. The BI application references the scored columns as required.

2. Database does the scoring.

Database uses embedded scoring algorithms to score records in response to SQL queries from a BI application.

3. BI application does the scoring.

The BI Platform scores records using scoring metrics and reports.

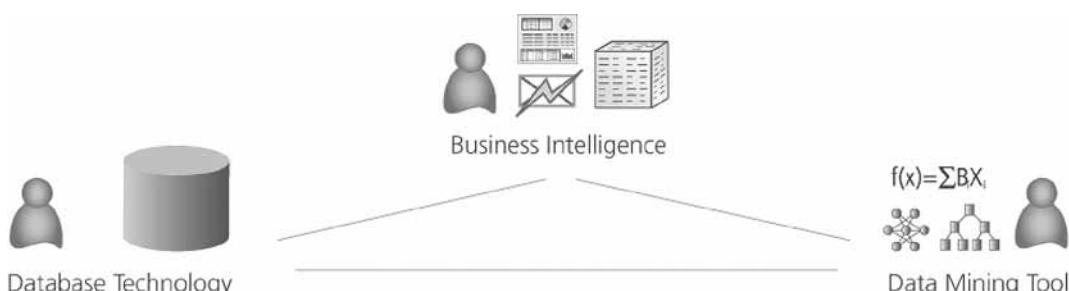


Figure 13-1 Scoring can be performed by any one of the three components of enterprise BI applications, i.e., data mining tool, database, or the BI application.

All three approaches are viable methods for deploying data mining results throughout the enterprise. Determining which approach to use depends greatly on the business need for predictive analysis, and the IT infrastructure and philosophy.

The starting point for most data mining implementations is to use the data mining tool for scoring. Although it is very common for the data mining analyst to provide scores in standalone flat files or spreadsheets, integrating scored results into databases has long been a common practice.

When scoring is required on a real time basis, or when predictive models are created and changed faster than scores can be calculated and stored in the database, one of the other approaches must be adopted. If the database supports data mining, deploying models in the database is a possible next step. If the BI Platform contains data mining capabilities, deploying models directly in BI applications can speed the adoption of predictive analysis by business users.

Irrespective of which approach is suited for an organization's predictive analysis needs, the MicroStrategy BI platform can integrate predictive insight into all BI applications used by business users. The following sections discuss the benefits and drawbacks of each approach.

Data Mining Tool Does the Scoring

In this approach, a data mining scoring application calculates, and inserts scores into the database as new tables, new columns in existing tables or updates to existing (old) scores. Once the scores are part of the database, the BI application reads these scores just like any other data, directly from the database. Historically, this approach has been the most common, and has the following benefits and drawbacks:

Benefits:

- Since a data mining tool does the scoring, model complexity, and performance is hidden within the scoring engine. The scoring process does not require any BI resources, and should not impact other concurrent BI processes
- At runtime, BI applications simply read the scores from the database without having to calculate scores on the fly

Drawbacks:

- Requires database space and database administrator (DBA) support
- Large datasets can take a very long time to score
- New records inserted after batch scoring are not scored
- Updating the data mining model or scores requires more database and DBA overhead
- Adding new or changing existing models requires rescore the data

Database Does the Scoring

In this approach, data mining features inside the database management system perform the scoring. Several major databases have the ability to score data mining models. The most common approach is to import the predictive model into the database, and then generate scores by using extensions to SQL queries. A key feature of this approach is that the model can be imported and stored in the database. Several standards, such as the Predictive Model Markup Language (PMML), OLE DB for Data Mining, and the JSR-73 Java standards, enable the database to import of predictive models. The sophisticated techniques needed to create the model are not required to score the data. Scoring simply involves mathematical calculations on a set of inputs to generate a result. This approach has the following benefits and drawbacks:

Benefits:

- Scores can be done “on the fly” even if new records are added
- Updating the model is easier than having to re-score the entire database
- Requires less database space than scoring the database since scores do not have to be persisted in the database
- BI applications can take advantage of this approach by using the database’s data mining capabilities directly

Drawbacks:

- Requires database space and database administrator support
- Requires application knowledge of the database’s data mining capabilities. Typically, this is different from the database administration skills
- BI applications must be customized for each database’s data mining implementation

Business Intelligence Tool Does the Scoring

The third approach for integrating data mining uses enterprise data resources without significantly increasing the database overhead. This is accomplished by importing predictive models into the BI platform as standard metrics. Deploying predictive models in the BI platform allows sophisticated data mining techniques to be applied directly within the business intelligence environment on only the data that has been requested. Like the other approaches, it also has benefits and drawbacks:

Benefits:

- Scores can be done “on the fly” even if new records are added
- Adding a new model or updating an existing model is simply a matter of importing the new definition
- Does not require database space or database administrator support

Drawbacks:

- Input characteristics need to be passed to the BI application even if they are not displayed on the report
- Very large datasets may use a large amount of BI resources

13.2 DATA MINING SERVICES IN MICROSTRATEGY

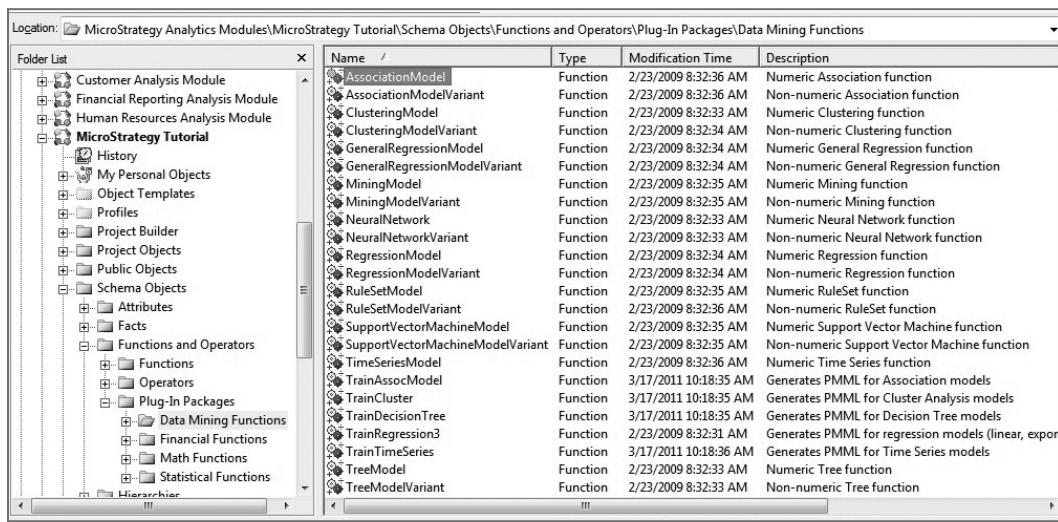
MicroStrategy introduced Data Mining Services to provide data mining scoring capabilities in Intelligence Server, making MicroStrategy the only BI platform that supports all three approaches to integrating data mining with mainstream business intelligence. The goals behind the introduction of Data Mining Services were as follows:

- Let business users apply the predictive models in their everyday reports, against all relevant data in data warehouse
- Minimize the time between data scoring, and the availability of the score to end users
- Improve the distribution and reach of the predictive insight through the various user interfaces of the MicroStrategy platform, including Mobile smartphones and tablets, the zero-footprint Web client, the integration with Microsoft Office, the Desktop development environment, and the proactive report delivery and user self-service capabilities

- Provide a unified security and industrial-strength BI architecture for data mining that is integrated with the rest of the BI platform
- Accomplish all of the above with minimal assistance from the database administrator

13.3 DATA MINING ALGORITHMS

Data Mining Services added numerous scoring algorithms to the extensive library of over 270 analytic functions available in the MicroStrategy BI platform.



The screenshot shows the MicroStrategy Analytics Modules interface. The left sidebar displays a folder tree under 'MicroStrategy Tutorial'. The 'Schema Objects' section is expanded, showing 'Attributes', 'Facts', 'Functions and Operators', and 'Plug-In Packages'. Under 'Plug-In Packages', 'Data Mining Functions' is selected, revealing a list of 30 functions. The right pane lists these functions with their names, types, modification times, and descriptions. Some descriptions mention PMML generation for specific models like Association, Clustering, Neural Network, Regression, and Tree models.

Name	Type	Modification Time	Description
AssociationModel	Function	2/23/2009 8:32:36 AM	Numeric Association function
AssociationModelVariant	Function	2/23/2009 8:32:36 AM	Non-numeric Association function
ClusteringModel	Function	2/23/2009 8:32:33 AM	Numeric Clustering function
ClusteringModelVariant	Function	2/23/2009 8:32:34 AM	Non-numeric Clustering function
GeneralRegressionModel	Function	2/23/2009 8:32:34 AM	Numeric General Regression function
GeneralRegressionModelVariant	Function	2/23/2009 8:32:34 AM	Non-numeric General Regression function
MiningModel	Function	2/23/2009 8:32:35 AM	Numeric Mining function
MiningModelVariant	Function	2/23/2009 8:32:35 AM	Non-numeric Mining function
NeuralNetwork	Function	2/23/2009 8:32:33 AM	Numeric Neural Network function
NeuralNetworkVariant	Function	2/23/2009 8:32:33 AM	Non-numeric Neural Network function
RegressionModel	Function	2/23/2009 8:32:34 AM	Numeric Regression function
RegressionModelVariant	Function	2/23/2009 8:32:34 AM	Non-numeric Regression function
RuleSetModel	Function	2/23/2009 8:32:35 AM	Numeric RuleSet function
RuleSetModelVariant	Function	2/23/2009 8:32:36 AM	Non-numeric RuleSet function
SupportVectorMachineModel	Function	2/23/2009 8:32:35 AM	Numeric Support Vector Machine function
SupportVectorMachineModelVariant	Function	2/23/2009 8:32:35 AM	Non-numeric Support Vector Machine function
TimeSeriesModel	Function	2/23/2009 8:32:36 AM	Numeric Time Series function
TrainAssocModel	Function	3/17/2011 10:18:35 AM	Generates PMML for Association models
TrainCluster	Function	3/17/2011 10:18:35 AM	Generates PMML for Cluster Analysis models
TrainDecisionTree	Function	3/17/2011 10:18:35 AM	Generates PMML for Decision Tree models
TrainRegression3	Function	2/23/2009 8:32:31 AM	Generates PMML for regression models (linear, expon)
TrainTimeSeries	Function	3/17/2011 10:18:36 AM	Generates PMML for Time Series models
TreeModel	Function	2/23/2009 8:32:33 AM	Numeric Tree function
TreeModelVariant	Function	2/23/2009 8:32:33 AM	Non-numeric Tree function

Figure 13-2 MicroStrategy supports numerous scoring algorithms that execute the predictive analysis models such as regression, neural network, decision tree, clustering, and segmentation.

- Regression

Regression is a powerful, commonly used algorithm that evaluates the relationship of one variable, the dependent variable, with one or more other variables, called independent variables. By measuring exactly how large and significant each independent variable has historically been in its relation to the dependent variable, the value of an unknown dependent variable can be estimated using known values of the independent variables. Regression models are widely used in applications such as seasonal forecasting, quality assurance, and credit risk analysis.

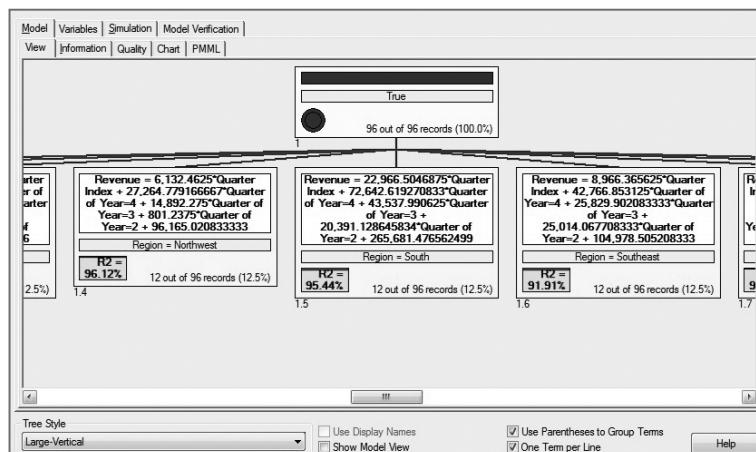


Figure 13-3 An example of a regression model that applies a multi-variate regression predictor for sales by region and quarter. The tree-regression model predicts quarterly on-line sales revenues for each region.

- Neural Network

A Neural Network is a sophisticated pattern detection algorithm that uses machine learning techniques to generate predictions. This technique models itself after the process of cognitive learning and the neurological functions of the brain capable of predicting new observations from other known observations. Neural networks are very powerful, complex, and accurate predictive models that are used in detecting fraudulent behavior, predicting the movement of stocks and currencies, and improving the response rates of direct marketing campaigns.

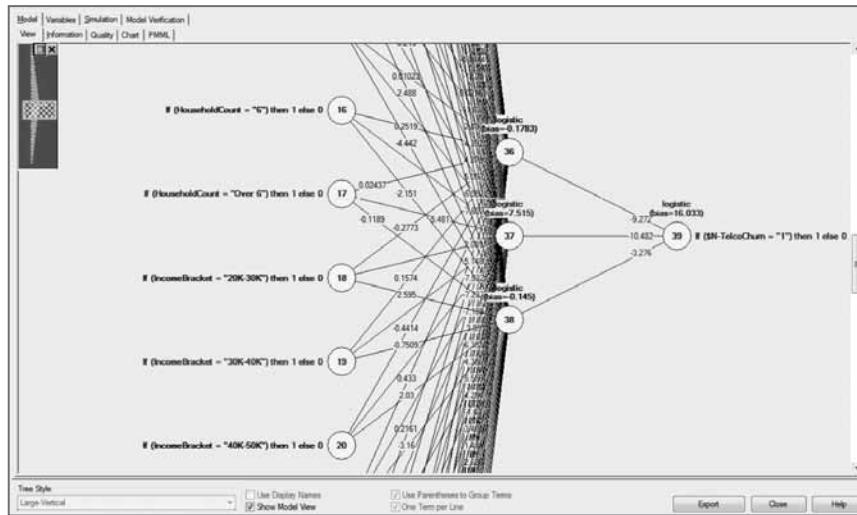


Figure 13-4 Example of a simple neural network model used to predict a customer's likelihood to respond to a marketing campaign using the customer's psychographic profile.

- Decision Tree

A Decision Tree is a tree-shaped graphical predictive algorithm that represents alternative sequential decisions, and the possible outcomes for each decision. This algorithm provides alternative actions that are available to the decision maker, the probabilistic events that follow from and affect these actions, and the outcomes that are associated with each possible scenario of actions and consequences. Their applications range from credit card scoring to time series predictions of exchange rates.

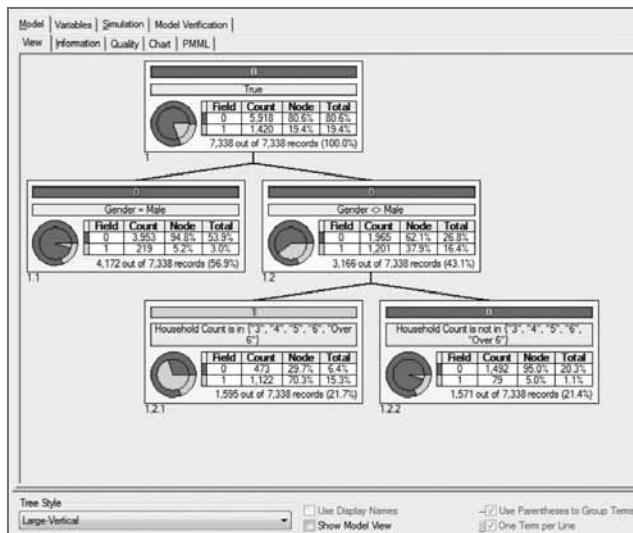


Figure 13-5 A Decision Tree model used to predict the lifetime value of a customer based upon demographic and psychographic factors. It displays the number of customers by income range with a high lifetime value. The tree algorithm then breaks down high lifetime value customers by the number of people in a household.

- Clustering/Segmentation

Clustering/Segmentation is the process of grouping items together to form categories. Once the characteristics of a group of items are determined, they can be used to find customers with similar characteristics. This algorithm is used to create logical groups, such as customers for marketing campaigns, rate groups for insurance products, and crime statistics groups for law enforcement.

Field Name	Cluster 1	Cluster 2	Cluster 3
Age Range=24 and under	0.07	0.09	0.07
Age Range=25 to 34	0.15	0.14	0.20
Age Range=35 to 44	0.37	0.34	0.37
Age Range=45 to 54	0.17	0.19	0.14
Age Range=55 and over	0.24	0.25	0.23
Education=Graduate	0.20	0.19	0.22
Education=High School	0.40	0.41	0.36
Education=Other	0.09	0.11	0.11
Education=Undergraduate	0.31	0.29	0.31
Gender=Female	0.43	0.45	0.43
Housing Type=Dependent	0.17	0.19	0.19
Housing Type=Owner	0.33	0.34	0.35
Housing Type=Renter	0.50	0.47	0.46
Marital Status=Married	0.00	0.00	1.00
Marital Status=Previously Married	0.00	1.00	0.00
Marital Status=Single	1.00	0.00	0.00

Tree Style: Large-Vertical

Use Display Names Use Parentheses to Group Terms
 Show Model View One Term per Line

Figure 13-6 Example of a classification model used to segment customers based upon products in their shopping cart.

- Time Series Analysis

Time series represents a broad and diverse set of analysis techniques which use a sequence of measurements to make forecasts based on the intrinsic nature of that data. The past behavior of the target variable is used to predict its future behavior. This past behavior is measured in a time-based sequence of data, and most often that sequence is a set of measurements taken at equal intervals in time. By analyzing how values change over time, time series analysis attempts to find a model that best fits the data. Using averages has the effect of smoothing the data by diminishing the peaks and valleys. A common solution is a technique called exponential smoothing. There are several models that can be used:

- Simple average of all data points
- Moving Average which is the average of the previous time period target data. It does a better job of following the data than the simple average. But, tends to lag behind the target data
- Centered version of the Moving Average. This is an improvement over the non-centered version since the lag has been eliminated, but at the cost of delaying the forecast

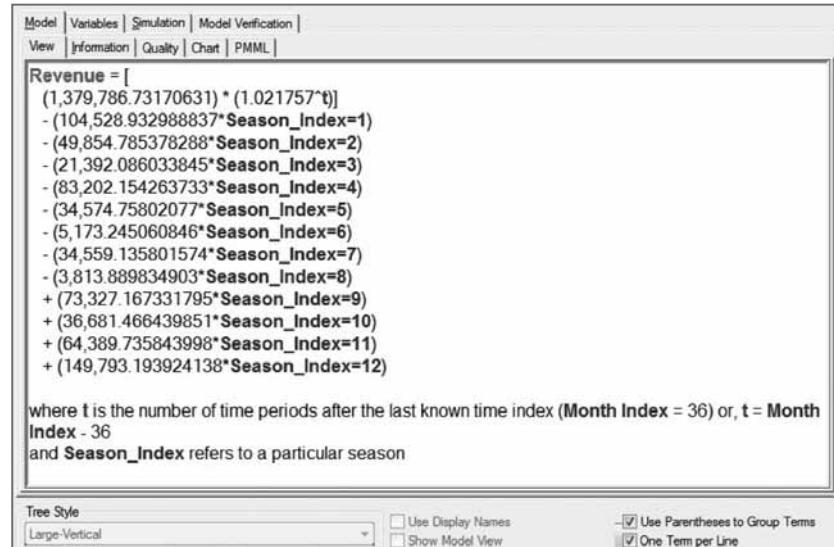


Figure 13-7 Example of a Time Series Analysis used to forecast the Quarterly Revenue over time.

- Association Rules

Association rules look for relationships between items. The most common example of this is market basket analysis. Market basket analysis studies retail purchases to determine which items tend to appear together in individual transactions. Online retailers use market basket analysis on their websites to suggest additional items to purchase before a customer completes their order. The key to this type of analysis is the ability to find associations amongst the items in each analysis. This can include associations such as which items appear together the most frequently, and which items tend to increase the likelihood that other items will also appear in the same transaction.

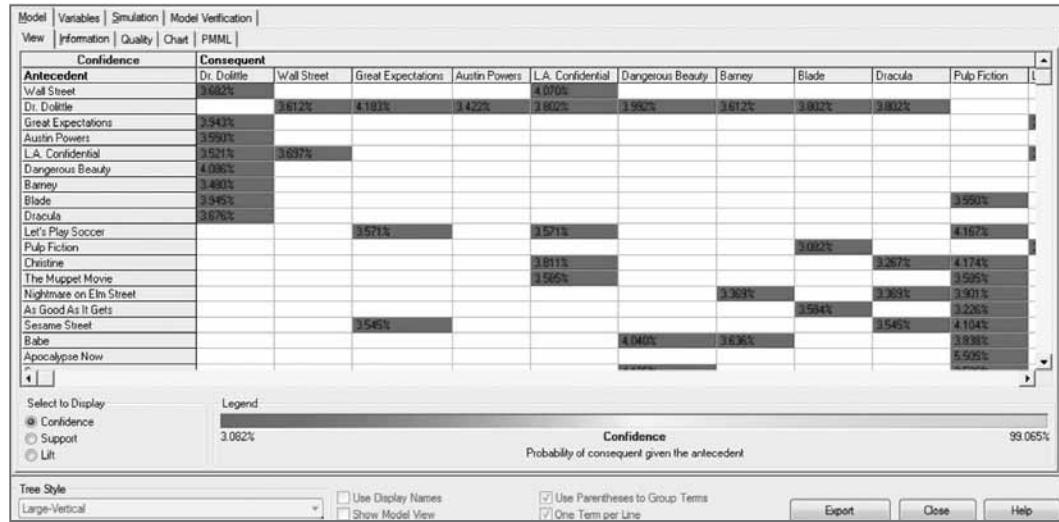


Figure 13-8 Example of an Association Rules Analysis matrix.

- Monte Carlo Simulation

Monte Carlo Simulation is a class of computational algorithms that rely on repeated random sampling to compute their results. Monte Carlo methods are often used when simulating physical and mathematical

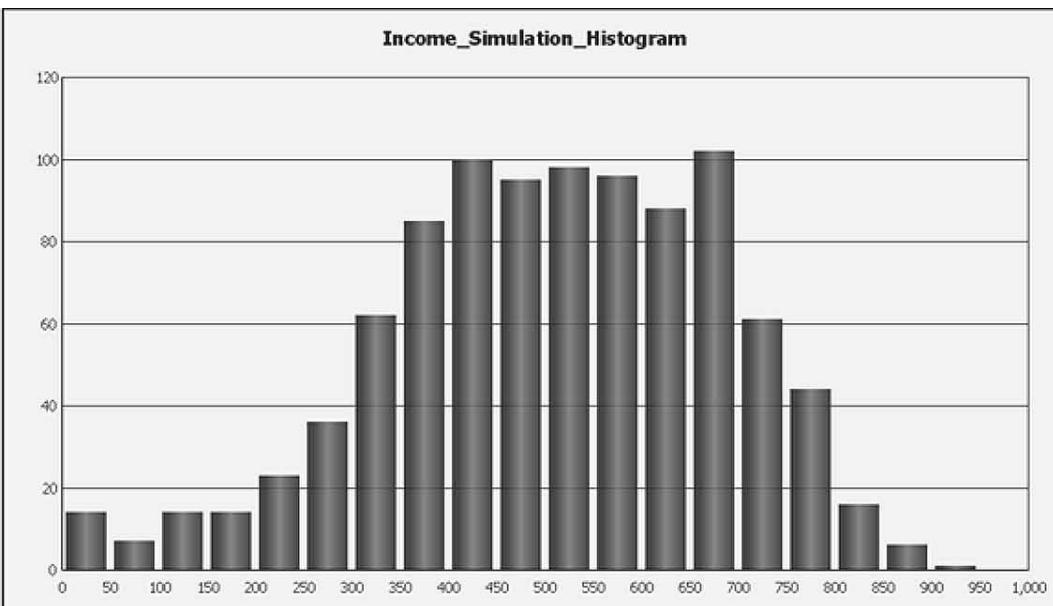


Figure 13-9 A histogram of a Monte Carlo Simulation that predicts the Income based on existing parameters such as Profit and Cost.

Predictive Analysis Using MicroStrategy Analytic Library

The MicroStrategy BI platform contains an analytic function library of over 270 basic, OLAP, mathematical, financial, and statistical functions that can be used to create business metrics and key performance indicators. In addition, it offers numerous advanced analysis capabilities such as set analysis, transformations, dimensional, and conditional metrics that can be used in conjunction with the function library to provide predictive analysis without requiring a third-party data mining tool.

For example, market basket analysis identifies customers who are likely to buy certain products based upon the existence of other items in their shopping carts. This technique uses historical purchase information from other customers to identify products that are commonly purchased together. This information can be used to target customers who did not buy a complementary product with a marketing campaign.



Figure 13-10 Market basket analysis predicts customers likely to purchase TVs.

Another example of the native predictive analysis capabilities in the MicroStrategy platform is the ability to define multi-variate regression models that relate several independent variables to predict an outcome of the dependent variable. The regression model supports both linear and exponential regression. These models can be constructed as tree models where each node of a tree contains a separate regression.

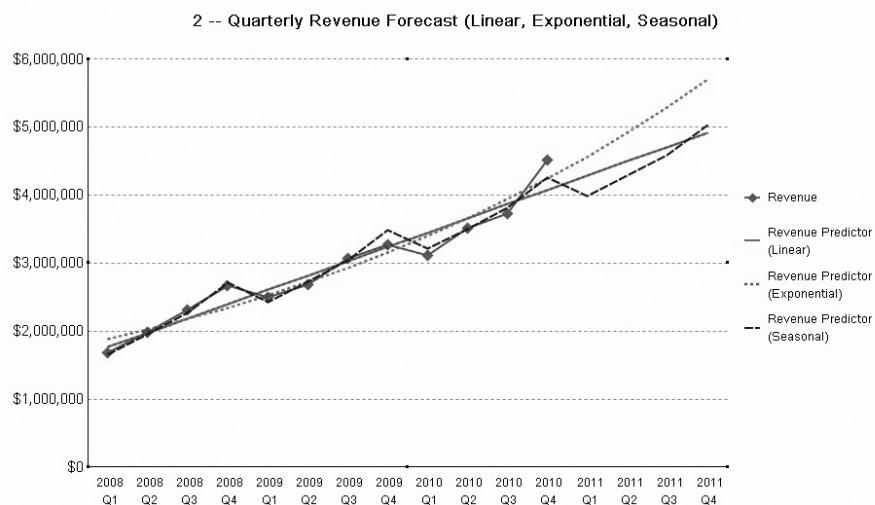


Figure 13-11 Business users can create and view predictive reports just as they would with any other MicroStrategy report. This report shows how linear regression, the straight line, and multi-variate regression, the dotted line, is used to forecast sales using quarterly periods.

Regression models are created with a Training Metric Wizard that prompts for the dependent variable, independent variables, and tree levels. The Wizard also specifies the desired outputs of the model, including the score and the confidence level of the score. Upon completion, the wizard generates a standard MicroStrategy metric, and a PMML rendering of the model.

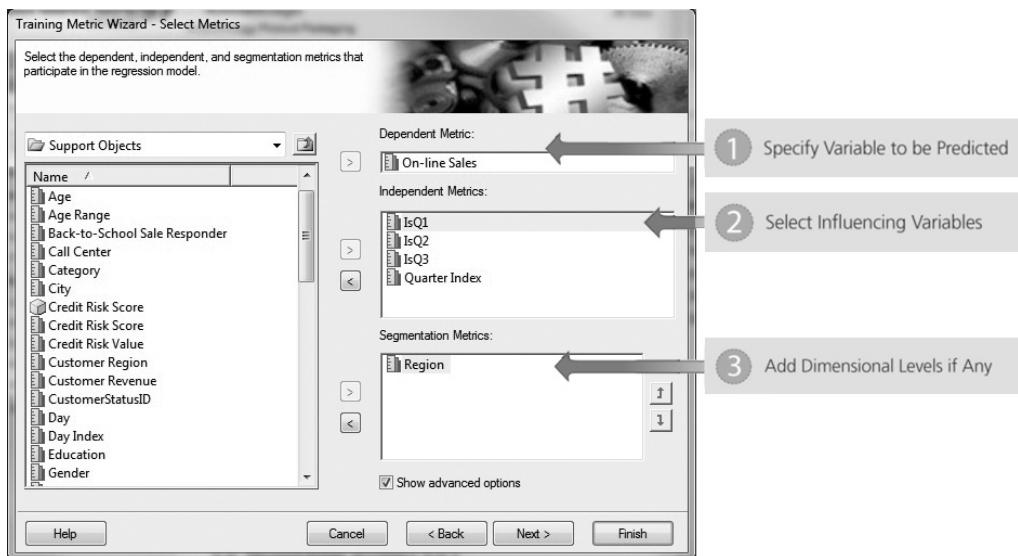


Figure 13-12 A Training Metric Wizard allows users to select the dependent and independent variables, and automatically generates the regression model and associated predictive metric.

Predictive Analysis Using Third-Party Data Mining Products

With Data Mining Services, organizations can maximize their investment in data mining products by importing a predictive modeling mark-up language (PMML) representation of the predictive models generated by the data mining product, and seamlessly encapsulating the model as a predictive metric for use with any report or document. Business users continue to inherit all the object-reuse, security, administrative, and manageability benefits of the MicroStrategy BI platform.

The workflow that integrates the MicroStrategy platform with third-party data mining tools to generate and use predictive metrics is described below.

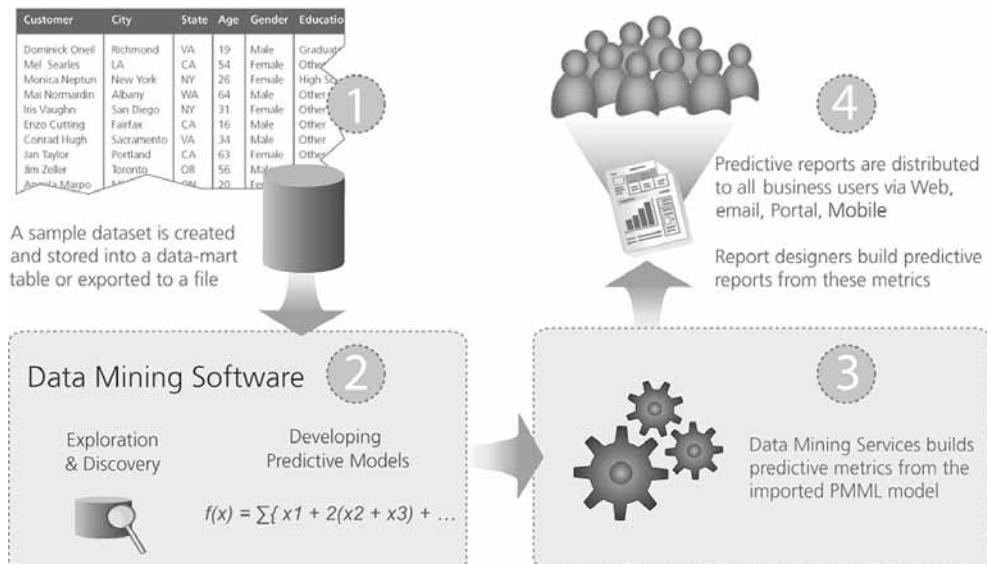


Figure 13-13 Data mining workflow 1) create a data sample in MicroStrategy, 2) import the sample into a data mining tool and develop the predictive model, 3) import the model into MicroStrategy, and 4) use the predictive metrics in reports, filters, and other metrics.

1. Create a sample dataset in MicroStrategy and export it to a third-party data mining tool

A user creates a standard report in MicroStrategy that contains a known historical outcome to a business issue, and a list of characteristics that may or may not influence the outcome. The user runs this report, and can add more information or drill down or across to other hierarchies to get a desired data sample. Next, the data sample is either exported to various formats, for example text files or Microsoft Excel, or saved to a database table using a dynamic datamart in MicroStrategy. Dynamic datamarts are a preferred approach when dealing with large datasets, whereas exporting to a text file or Excel may be more convenient for smaller datasets.

2. Use third-party data mining tools to develop predictive models

A domain expert, data mining analyst or statistician imports the data sample into the third-party data mining tool through an ODBC connection to the datamart table or by simply uploading the exported file. The data mining tool analyzes the data sample in search of predictive information, and builds a predictive model from this dataset. When complete, the predictive model is exported to an XML file in PMML.

3. Upload the PMML predictive model, and generate predictive metrics

The PMML file is uploaded into MicroStrategy, which interprets the predictive model, and automatically creates scoring and confidence predictive metrics. These new predictive metrics are stored in the MicroStrategy metadata, and can be used like any other metric. By complying with the PMML standard, MicroStrategy works with all leading data mining tools.

4. Build and distribute reports with predictive metrics

Predictive metrics can be added to reports just like any other MicroStrategy metric. All report manipulations, such as sorting, filtering, adding new calculations, and defining thresholds are possible. Predictive metrics also inherit all the object-reuse, security, administrative, and manageability benefits of the MicroStrategy BI platform.

13.4 BENEFITS OF INTEGRATING BUSINESS INTELLIGENCE AND DATA MINING

While adding data mining scores and predictive models directly in the database is beneficial, there is additional value to be gained by integrating data mining scoring inside the BI platform.

- Business users can view predictive reports in a wide variety of user interfaces
- Highly formatted predictive reports provide the easiest possible user consumption and professional presentation
- Personalized messages and predictive reports can be delivered to very large user populations based on alerts or schedules
- Ad hoc query and analysis that includes predictive metrics is possible without requiring knowledge of SQL, table structures, or predictive models
- Business analysts can perform further analysis, such as slice-and-dicing data, ad hoc report creation, drilling, pivoting, and sorting, on predictive reports
- Strict security is applied to users within, and outside the organization

The industry leading MicroStrategy BI platform is the first to deliver data mining and predictive analytics to all users through a fully integrated enterprise-caliber BI system. Business users, report designers, and analysts can view and build predictive reports using MicroStrategy, and distribute these reports to all relevant decision makers and stakeholders.

13.5 SUMMARY

Data mining involves use of existing information to gain new insights into business activities by applying predictive models and analysis techniques. MicroStrategy's Data Mining Services facilitates the development and deployment of these techniques such as multi-variate linear regressions, multi-variate exponential regressions, logistic regressions, decision trees, clustering predictive models, association rules, and time series models. This helps organizations use their data to forecast future outcomes.

14

HETEROGENEOUS DATA ACCESS

In most organizations, data is not stored in a single enterprise data-warehouse but instead is stored in different, distributed data sources across the organization. While the aim of any organization is to maintain all data in a single enterprise-wide data warehouse, there are always compelling reasons to keep certain types of data separated:

- Prototyping – Data is in a Proof-of-Concept application that has not been fully tested or passed data-quality checks
- Pre-aggregated data – Financial or Planning and analysis data stored in OLAP cubes
- Hosted data storage – Sales force automation information hosted in Cloud applications
- Varying departmental requirements – Organizational structure may necessitate that different departments maintain their own data sources
- Unstructured data – Large sets of unstructured data stored in a Hadoop distributed file system

14.1 DATA IS STORED IN MANY PLACES

It is critical, however, for any organization to deploy a BI platform that easily connects to heterogeneous data sources and reports on them. MicroStrategy's BI platform provides a single object-oriented business metadata layer that acts as an information gateway for business information and masks the technical structures related to the data sources. Business users simply view the information as a set of attributes, metrics, and other metadata objects using terminology with which they are familiar. MicroStrategy's BI platform supports a wide range of data sources:

- Relational databases
- Multi-dimensional sources
- Web Services and XML Sources
- Salesforce.com
- Hadoop distributed file systems
- Excel files

MicroStrategy uses various techniques to ensure that the correct data is retrieved, with the minimal amount of network traffic and the highest performance. The platform also provides the capability to merge data from different sources either at a report level or at a document level.

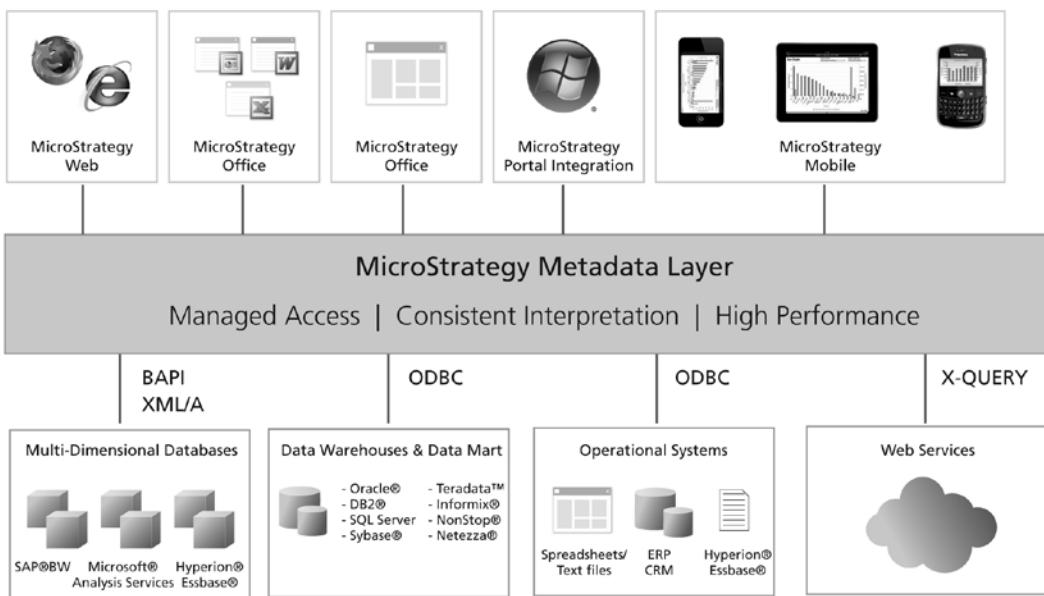


Figure 14-1 The MicroStrategy metadata layer abstracts data sources into a single view for business users and all applications.

14.2 RELATIONAL DATA SOURCES

The MicroStrategy BI platform provides business users three types of reports to retrieve data from relational data sources. Each report type accesses the relational data source using wire protocol ODBC (Open Database Connectivity) drivers that pass SQL queries to the data source for execution and return query results to the Intelligence Server. MicroStrategy can access data stored in any ANSI 92 relational data source accessible by ODBC, including data warehouses, data warehouse appliances, Microsoft Excel files, CSV text files, and other operational databases of systems such as ERP systems, RFID applications, web site tracking, call center tracking, and CRM systems.

Business Model-Based Reports

Over the past 20 years, the MicroStrategy's SQL Engine has evolved to support complex business intelligence requirements. A physical data warehouse schema is abstracted into a logical business model in the MicroStrategy metadata repository using business terms and rules used by the business. Users build queries by dragging and dropping metadata objects in the business model onto rows, columns, and pages of a report grid, and into the filter criteria for the query. The SQL Engine, which is part of MicroStrategy Intelligence Server, translates the metadata objects that define reports and dashboards into one or more SQL queries optimized for the specific database platform being queried.⁹

⁹See Chapter 11 Scalability and Performance for details on optimized SQL generation

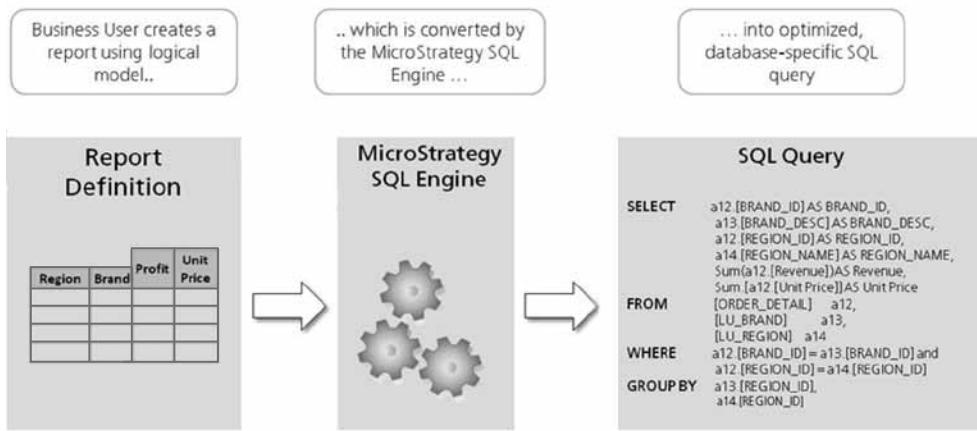


Figure 14-2 The MicroStrategy SQL Engine translates a business user report into an optimized SQL query.

Freeform SQL Reports

Hand-crafted, or freeform, SQL statements can be embedded directly into the MicroStrategy BI platform to define a report or a logical table similar to a database view. Freeform SQL statements are subject to the full suite of security features of the MicroStrategy BI platform. Prompts can also be included, increasing the flexibility of the SQL statements. The data retrieved by a freeform SQL report can be mapped to the business model or to other freeform SQL queries. This mapping lets dashboards that display data from different sources be controlled by a single set of selectors.

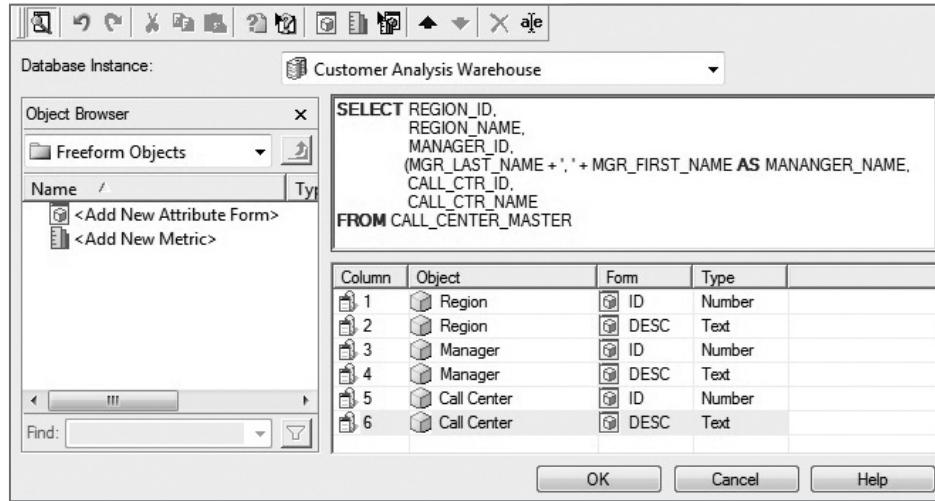


Figure 14-3 Freeform SQL lets users write their own SQL queries and execute them against any data source that is accessible via ODBC.

Query Builder Reports

The Query Builder interface is a report design interface that generates a SQL query defined by a graphical view of tables, columns, and joins. The interface displays the available tables and columns in a database,

which are dragged onto a palette and joined together. The SQL Engine converts these table, column, and join diagrams into a SQL query and identifies any missing parameters. Query Builder reports inherit the security settings of the MicroStrategy platform. The query design can include prompts, filters, and advanced calculations.

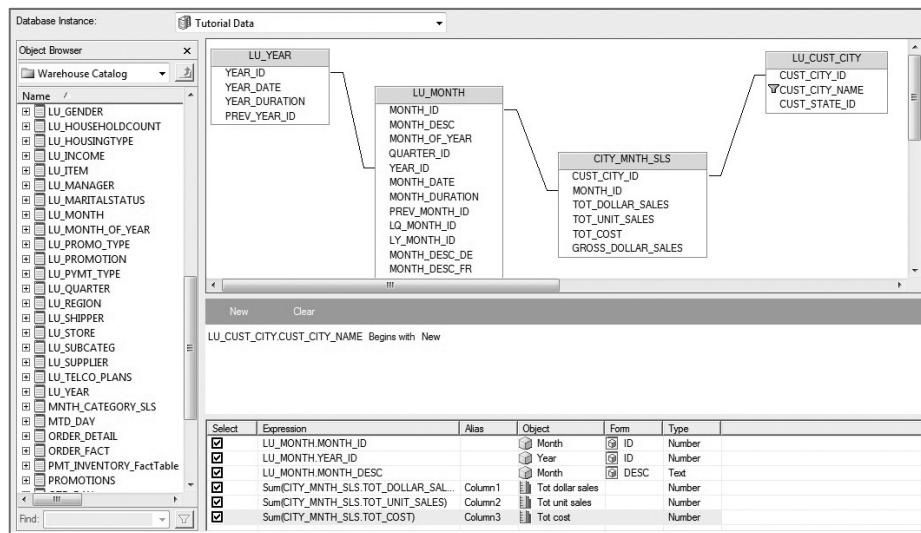


Figure 14-4 The Query Builder interface enables users to visually build their SQL query and execute it against any relational data source.

Import Data Directly into Intelligent Cubes

To facilitate user self-service, data from relational databases can be imported directly into MicroStrategy Intelligent Cubes from a web browser without any IT assistance, and used to build report grids, graphs, dashboards and visual analyses. A Web user can type or copy freeform SQL queries and import data into MicroStrategy from existing database connections defined in the metadata and from new connections defined at the time of import. Optional mapping of the SQL columns to the MicroStrategy business model lets dashboards display and manipulate imported data and enterprise data together through a single set of selector controls.

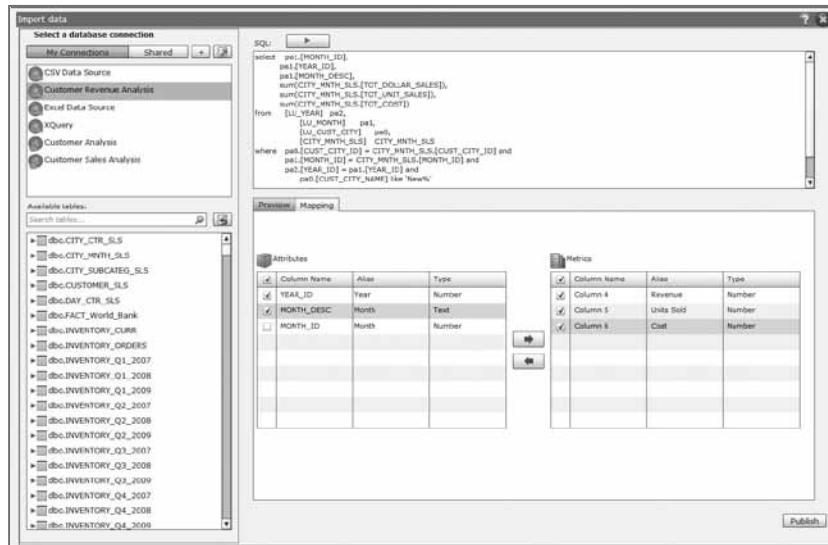


Figure 14-5 Business users can import the results of their hand-written queries into MicroStrategy from a Web browser.

14.3 MULTI-DIMENSIONAL SOURCES

Many organizations use multi-dimensional cube sources to store business data for reporting and analysis. These cubes contain the multi-dimensional business model as part of the cube and are queried using the MDX (MultiDimensional Expressions) query language. The MicroStrategy BI platform can extract business information from the following multi-dimensional sources:

- SAP BW
- Hyperion Essbase
- Microsoft Analysis Services
- IBM Cognos TM1

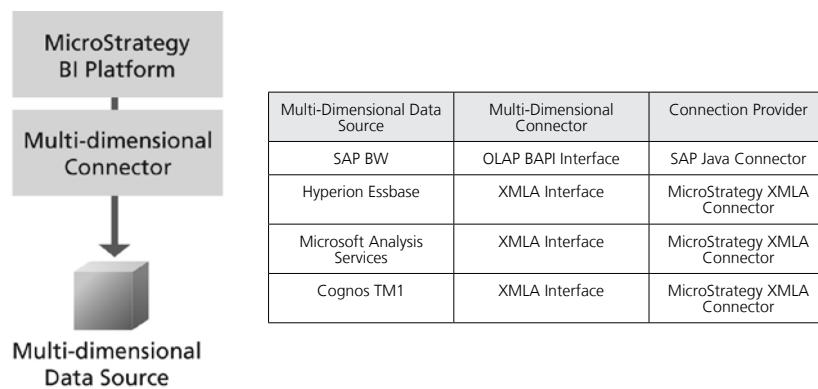


Figure 14-6 MicroStrategy connects to multi-dimensional sources with connectors that are specific to each source.

Multidimensional Cube Reports

A cube definition from multi-dimensional sources is imported into the MicroStrategy metadata repository and its reporting structures are automatically defined as MicroStrategy metadata objects. Depending on the cube platform, characteristics, structures, and levels are defined as MicroStrategy attributes, key figures and measures as MicroStrategy metrics, hierarchies and dimensions as MicroStrategy hierarchies, and variables as MicroStrategy prompts. Business users and developers build their reports in the same way as with relational data sources – by dragging and dropping these metadata objects into pages, rows and columns of a report grid and filter. The MicroStrategy MDX Engine then generates an MDX query from the report definition. All BI functionality of the platform, including security, prompting, filtering, thresholds, and drilling is available on the report.

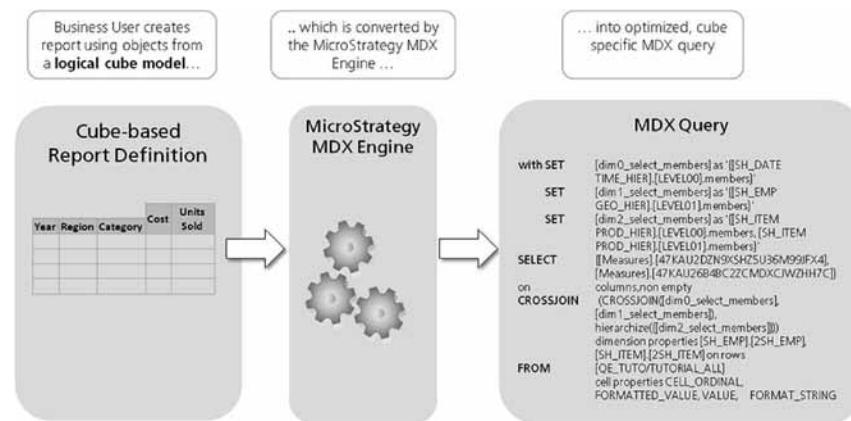


Figure 14-7 The MicroStrategy MDX engine converts a report definition into a multi-dimensional expression (MDX Query).

14.4 DATA ACCESS TO OTHER DATA SOURCES

Traditionally, business transactions are written to operational databases, from which they are extracted, cleansed, and loaded into a relational data warehouse or pre-aggregated multi-dimensional cubes. As more businesses are using data to gain a competitive advantage, access to additional data sources, both on-premises and external to the organization, has become prevalent to cater for the variety and volume of data needed. The MicroStrategy BI platform seamlessly accesses and queries these new data sources, including Web Services, Hadoop distributions, and Salesforce.com enabling organizations to report on data stored in these non-traditional systems.

Accessing Data from Web Services and XML sources

MicroStrategy retrieves data from SOAP and REST Web Services and XML data sources using XQuery scripts. XQuery is a scripting language specifically designed to query XML data. Some of the benefits of XQuery are:

- Apply filters to retrieve the information of interest
- Join data from different XML sources
- Sort, group and aggregate XML data
- Transform the XML data into a different structure

MicroStrategy's freeform XQuery reports can query a large number of XML data providers, such as ERP systems, web sites, Cloud applications, and social networking sites that use web services to let 3rd-party applications interact with their data. The MicroStrategy XQuery Engine acts as a gateway: extracting the XQuery from the freeform report, establishing communication with the web service or XML source, executing the XQuery, fetching the data in XML format, and transforming the results into a MicroStrategy report. MicroStrategy supports SOAP and REST Web Services, as well as local and remote XML files.

The screenshot shows the MicroStrategy Freeform XQuery interface. The top window is titled "XQuery" and contains the XQuery script:

```

let $result := doc("file:///Program Files %28%86%29/MicroStrategy/Tutorial Reporting/CSVFiles/XML_FreeForm_XQuery.xml")
return <Table>
<ColumnHeaders>
<ColumnHeader name="RegionID" type="xsd:integer" />
<ColumnHeader name="RegionDesc" type="xsd:string" />
<ColumnHeader name="CategoryID" type="xsd:integer" />
<ColumnHeader name="CategoryDesc" type="xsd:string" />
<ColumnHeader name="SubCategoryID" type="xsd:integer" />
<ColumnHeader name="SubCategoryDesc" type="xsd:string" />
<ColumnHeader name="YearID" type="xsd:integer" />
<ColumnHeader name="Revenue" type="xsd:number" />
<ColumnHeader name="Cost" type="xsd:number" />
<ColumnHeader name="UnitsSold" type="xsd:number" />
</ColumnHeaders>
<Data>
{
for $records in $result//XML_Dataset
return
<Row>
<RegionID>{data($records/Region_ID)}</RegionID>
<RegionDesc>{data($records/Region_DESC)}</RegionDesc>
<CategoryID>{data($records/Category_ID)}</CategoryID>
<CategoryDesc>{data($records/Category_DESC)}</CategoryDesc>
<SubCategoryID>{data($records/Subcategory_ID)}</SubCategoryID>
<SubCategoryDesc>{data($records/Subcategory_DESC)}</SubCategoryDesc>
<YearID>{data($records/Year_ID)}</YearID>
<Revenue>{data($records/Revenue)}</Revenue>
<Cost>{data($records/Cost)}</Cost>
<UnitsSold>{data($records/Units_Sold)}</UnitsSold>
</Row>
</Data>
</Table>

```

The left side of the interface features an "Object Browser" pane titled "Freeform Objects". It lists various objects such as "Add New Attribute Form", "Add New Metric", "# Months Support", "# of Engineers", "% Cases under 1 week", "% Cases under 1 week target", "% Closed under 1 week", "% Critical Cases", "% Critical Cases target", "% Customers High Satisfaction", "% Customers High Satisfaction", "Accounts receivable days", "Accounts receivable days gro", "Air_Month", "Air_Region", "Air_Airport", "Air_Airport_ID", "Air_Traffic_Type", "Airport", "Airport Comments", "Airport Country", "Airport Expansion", "Airport Long Name", "Amount", "Average Call Time", "Avg. Resolution Time", "Avg. Resolution Time LM", "Avg. Resolution Time LY", "Avg. Resolution Time target", "CAL County", and "CAL County ?". A "Find:" search bar is at the bottom of the Object Browser.

The bottom right corner of the interface shows a table with 10 columns and 10 rows, representing the schema of the XML dataset:

Column	Object	Form	Type
1	XQ_Region	ID	Number
2	XQ_Region	DESC	Text
3	XQ_Category	ID	Number
4	XQ_Category	DESC	Text
5	XQ_Subcategory	ID	Number
6	XQ_Subcategory	DESC	Text
7	XQ_Year	ID	Number
8	XQ_Revenue		Number
9	XQ_Cost		Number
10	XQ_Units_Sold		Number

Figure 14-8 Freeform XQuery provides the users with ability to retrieve data from Web Service and XML data sources.

MicroStrategy includes a graphical utility, the XQuery Editor and Generator (XEG), that generates XQuery scripts using the WSDL for SOAP web services, URI endpoints for REST web services, and DTDs for XML files. The XEG uses drag-and-drop features, wizards, and forms to generate, run, and display results from XQuery statements which can then be copied into MicroStrategy freeform query reports.

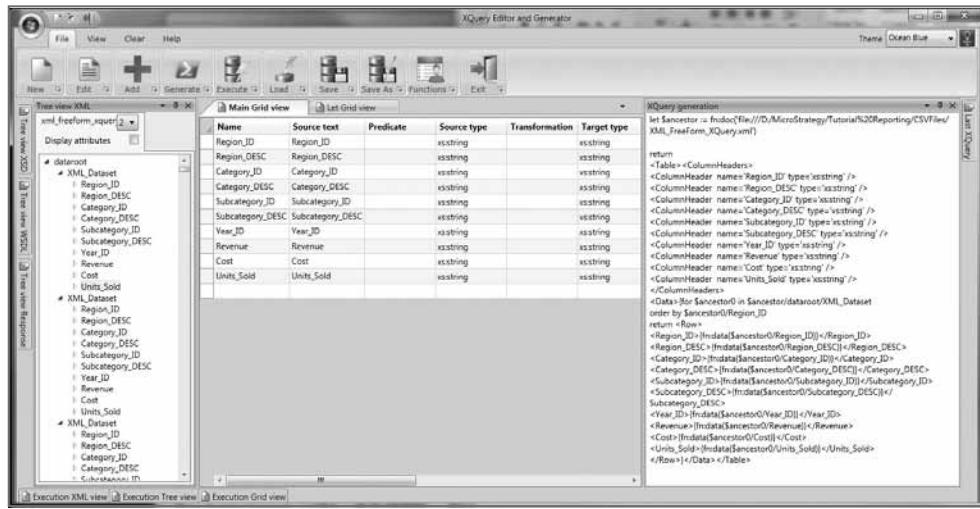


Figure 14-9 The XQuery Editor and Generator builds an XQuery visually from WSDL, URI endpoint, or DTD definitions.

Accessing Data from Salesforce.com

Many organizations use Salesforce.com for customer relationship management and sales force automation. Salesforce.com operates as a Software as a Service (SaaS) platform providing access through standard Internet connections. MicroStrategy provides four methods to access data from Salesforce.com:

1. SQL Metadata Model – MicroStrategy includes a Salesforce.com ODBC driver that accesses Salesforce.com as any other relational source. The Salesforce.com schema can be modeled as attributes, metrics and hierarchies. Business users and developers build their reports in the same way, as they would typically do – by dragging and dropping these metadata objects into pages, rows and columns of a report grid and filter. The MicroStrategy generates dynamic SQL queries as with any modeled relational database or MDX cube.
2. Data Import – Data from Salesforce.com can be imported directly into MicroStrategy from a Web browser using hand-written SQL statements. MicroStrategy's data import interface connects to Salesforce.com, retrieves the data and maps it into metadata objects. The imported data is stored in an Intelligent Cube, which is available for building report grids, graphs, dashboards, and visual analyses.
3. Freeform SQL – Through a freeform SQL report interface, hand-written SQL statements access Salesforce.com tables and display the data in a MicroStrategy grid. The freeform SQL queries connect and retrieve data using the Salesforce.com ODBC driver.
4. Freeform XQuery – MicroStrategy freeform XQuery reports use XQuery statements to retrieve information from Salesforce.com's web services. The resulting XML data is transformed into a MicroStrategy report.

Many MicroStrategy BI features, including prompting, pivoting, sorting, adding calculations, and drilling within a modeled schema, as well as security filters and Access Control Lists (ACLs) can be used with Salesforce.com data.

Accessing Data from Hadoop Distributed File Systems

The Hadoop distributed file system provides an efficient way of storing large volumes of data on a reliable and inexpensive system. Additionally, Hadoop lets business users analyze large volumes of data and gain valuable insights from unstructured data, something that was not possible earlier. The MicroStrategy BI platform provides connectivity to a Hadoop file system in three ways:

1. Metadata Model – MicroStrategy connects to Hadoop distributions using Hive ODBC drivers to pass optimized Hive queries to Hadoop. The Hive schema can be modeled as attributes, metrics and hierarchies, and queries are dynamically generated.
2. Freeform Query – Freeform query reports, which use either the Hive or the Pig Latin query language, can retrieve data stored in Hadoop. This data can be viewed in a report or populate an Intelligent Cube.
3. Data Import – MicroStrategy's data import interface connects to Hadoop file systems through the Hive ODBC connector or Hive Thrift connector and imports data directly into an Intelligent Cube. This data can be used in report grids, graphs, dashboards, and visual analyses.

Since MicroStrategy sees Hadoop as any other SQL-based relational source, standard MicroStrategy BI functionality – pivoting, filtering, sorting, and drilling within a modeled schema – is available in Hadoop-based reports.

14.5 JOINING DATA FROM HETEROGENEOUS DATA SOURCES

The MicroStrategy BI platform can merge data from heterogeneous data sources using join relationships defined in the business model or in a dashboard or document. For example, MicroStrategy can display forecasting revenue metrics from an Excel file or Access database with actuals revenue metrics from the data warehouse. In the first method, the merged data appears to come from a single source to the user, and can be presented in the same grid, chart, or visualization on a dashboard or report. Sophisticated calculations can be created using metrics from all the data sources. The second method presents data from different sources on the same page or screen, but in separate grids, charts, or visualizations controlled by selectors on common attributes.

Joining Data Sources in the Business Model

The MicroStrategy BI platform enables BI architects to create a single multi-dimensional business model in the metadata repository spanning multiple data sources. These sources can include any SQL-accessible databases, such as data warehouses, data marts, operational databases, departmental databases, and Excel workbooks, and any of the multi-dimensional cube sources such as SAP BW, Microsoft Analysis Services, Hyperion Essbase, and IBM Cognos TM1. This metadata layer, which consists of abstracted, heterogeneous data sources, provides a single view across multi-dimensional and relational sources.

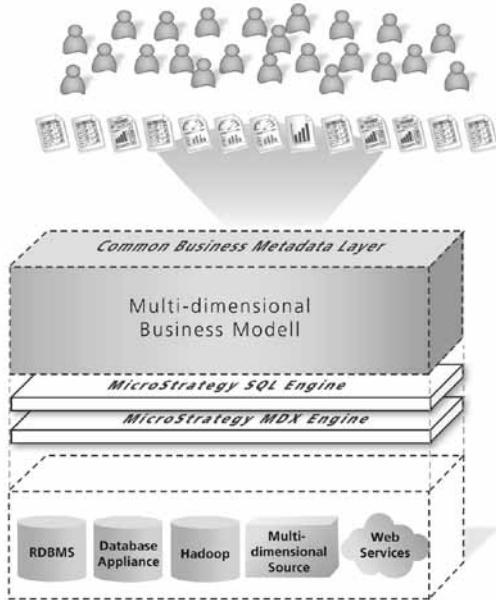


Figure 14-10 MicroStrategy BI platform provides a single view across relational and multi-dimensional sources for reporting.

A business analyst or report developer uses the abstracted metadata objects to build a report in the standard fashion without knowing that these objects map to tables and cubes coming from different data sources. The MicroStrategy Query Engine performs the following four steps when resolving a multisource report request:

1. Generates a series of multi-pass queries for each database needed by the report.
2. Forwards the queries to the corresponding data sources for execution.
3. Joins the final results from each of the queries in a relational database.
4. Retrieves and formats the final merged results.

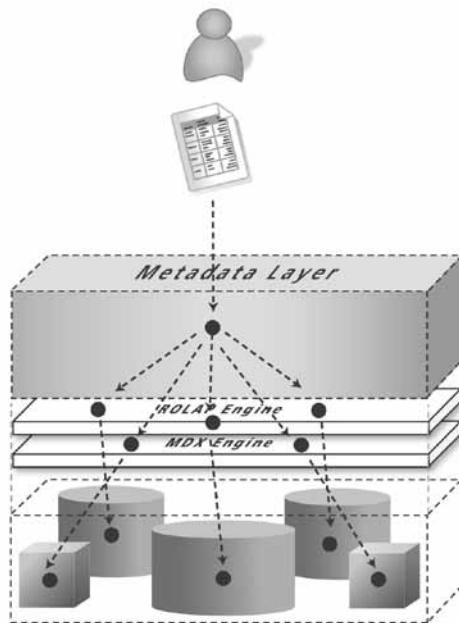


Figure 14-11 The Query Engine generates multi-pass report queries and sends them onto the corresponding data source for execution.

This entire process is transparent to the user, who performs the simple action of executing a report. Since the data sources have been mapped into a metadata business model, standard BI functionality like pivoting, filtering, calculations, and drilling is available with the full security capabilities provided by the MicroStrategy BI platform.

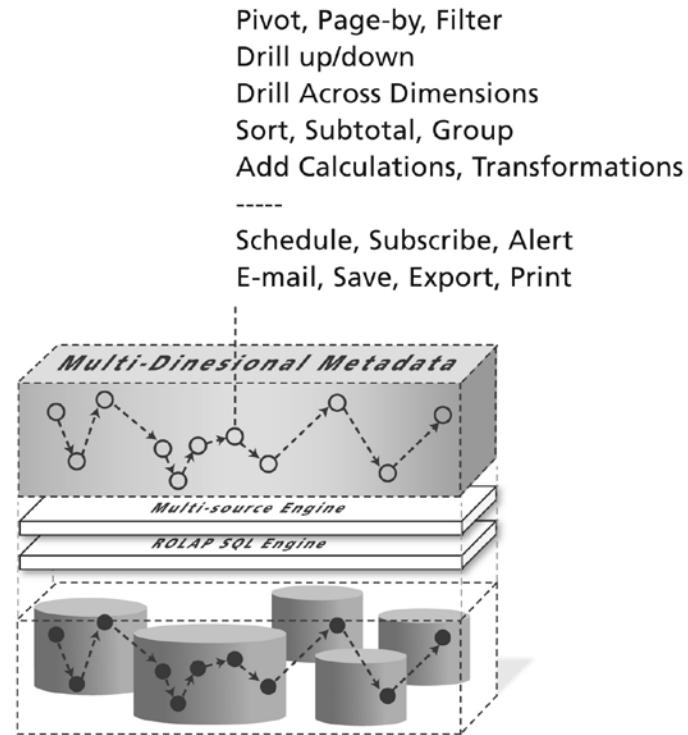


Figure 14-12 All BI functionality is seamless supported in a multi-source report.

Joining Data Sources in a Document or a Dashboard

Some reports display data from unrelated data sources for summary and informational purposes. A full business model is not required to accomplish this in MicroStrategy. Data from different sources can be displayed in separate grids, charts, or visualizations in a document or dashboard. Since this is not a metadata join and does not involve the MicroStrategy Query Engine, data from relational databases, multi-dimensional cubes, Intelligence Cubes, Web services, Salesforce.com, Hadoop, and other operational sources can be displayed together in a document or a dashboard. Data joins on common attributes need only be defined when a selector controls more than one grid, chart, or visualization.

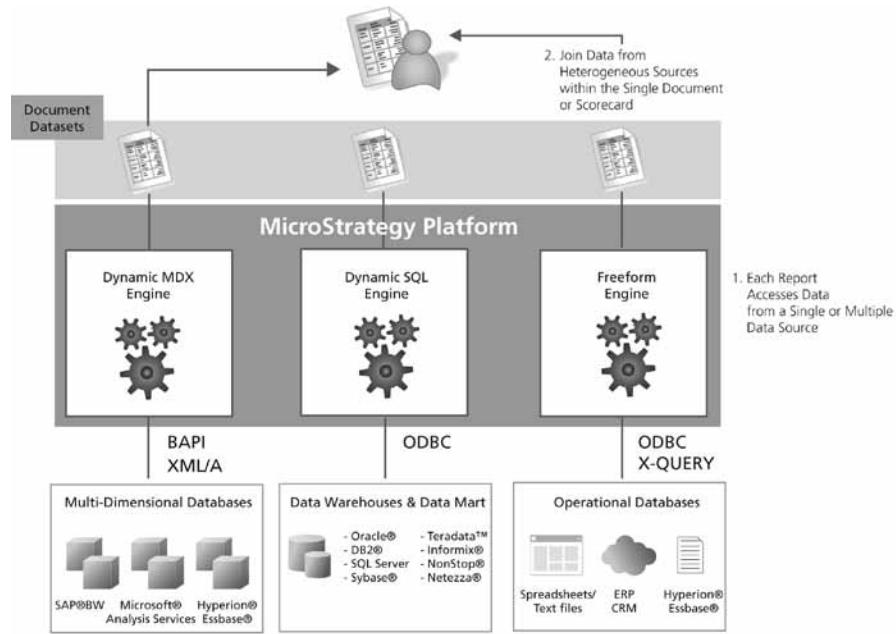


Figure 14-13 Data from a wide variety of sources can be joined at together in a document or a dashboard (the presentation layer).

BI functionality, such as pivoting, sorting, drilling, and calculations, is applied to individual grids or charts rather than for the entire document.

14.6 SUMMARY

As the amount of data available for analysis continues to grow, organizations are forced to use distributed data sources of different types for their storage needs. In the years to come, companies will have parts of their information on local data warehouses, other parts in multi-dimensional cubes and other parts in the Cloud. Since distributed data sources are the norm, organizations need a single BI platform that not only provides reporting on different types of established data sources but also provides the flexibility to access new types.

Recognizing this natural evolution, over the last 20 years, the MicroStrategy BI platform has provided access to a wide variety of data sources – relational databases, data-marts, operational systems, multi-dimensional cubes, Web services and XML sources, Excel files and Hadoop file systems – and will continue to incorporate access to new data sources. The platform ensures that organizations can present a variety of complex data sources as a single, simple, logical business metadata to its end users. This heterogeneous metadata follows all design tenets of the MicroStrategy architecture, including:

- Object oriented metadata objects
- Unlimited data scalability
- Unlimited user scalability
- Independence from hardware, software, and operating environments
- Data Access from any user interface – Desktop, web browsers and mobile devices
- Unparalleled security architecture
- Optimized for every data source
- Sophisticated analytical capability
- Multi-level caching and governing

15

EXPORTING DATA AND REPORTS

Most MicroStrategy users work online, running live reports against the most current information loaded in the data warehouse. MicroStrategy adds analytical context to this data, making it much more valuable for monitoring business processes, enriching enterprise reports, and improving analytical decision making.

15.1 MICROSTRATEGY SUPPORTS FLEXIBLE EXPORTING FOR ALL USES

Enhanced information is often needed outside MicroStrategy's interfaces and applications for:

- Offline Analysis – obtaining data while connected to a network, and then performing analysis after disconnecting
- Application Integration – sharing data with other applications, for example ERP, Supply Chain Management, and Budgeting systems, without human intervention
- Offline Review – generating static files, such as HTML, PDF, MHT/Flash and Microsoft Excel, Word and PowerPoint
- Batch Printing – distributing reports directly to any network printer in an industry-standard printable format
- Report Archives – storing reports and documents of any size in specified file locations for recording purposes

PLATFORM FUNCTIONALITY



Single Report – Multi-page Report – Programmatic Data Refresh – Million Cell Export – Data Marts – Bulk Files

Figure 15-1 Offline use cases arranged by user interactivity, data size, and frequency of export.

15.2 MICROSTRATEGY PROVIDES EFFICIENT EXPORTING OF DATA

Requests for data required outside the MicroStrategy BI platform are initiated differently, depending on how and where the data is going to be used. MicroStrategy provides four main approaches to fulfilling these requests:

- Exporting data directly from the user interfaces
- Saving reports in datamart tables in a database
- Performing bulk exports with Distribution Services
- Integrating with open APIs and Web services

Each of these methods addresses the requirement that the right data is transferred quickly and efficiently in a useful format. Deciding which method to use depends on the following factors:

- Data Accuracy
- Data Volume
- Data Format

Data Accuracy Keeps Export Processing to a Minimum

MicroStrategy contains many ways to control the accuracy of the data in a report. Ensuring that only the necessary data is exported reduces the amount of processing required to package and consume report information. Besides the standard filtering contained in reports and documents, this is accomplished via user input, ad hoc navigation, prompted reporting, and an automated security process.

Ad hoc navigation is performed naturally by a user, who changes the initial context of a report using standard analytical methods such as pivoting, page-by, sorting, and drilling. When this navigation results in a useful set of data for offline analysis, the user can export the data immediately if the result set is relatively small. For very large result sets, the information can be saved as a datamart table or a standard MicroStrategy report that Distribution Services can process.

The screenshot shows the MicroStrategy interface with a right-click menu open over a data grid. The menu options include Drill, Sort, Insert Metric, Move, Filter On..., Remove from Grid, Remove from Report, Advanced formatting..., Thresholds, Alerts, Edit Links..., Percent To Total, Transformation, Rank, Last Month's, Last Quarter's, New..., \$ 50,387 / \$45,3, \$ 68,752 / \$72,3, \$ 76,640 / \$88,1, \$ 146,186 / \$136,4, \$ 41,761 / \$46,765, \$ 116,812 / \$104,858, \$ 85,675 / \$85,164, \$ 114,455 / \$101,288, \$ 24,620 / \$25,916, \$ 20,106 / \$25,133, \$ 9,181 / \$9,767, \$ 33,741 / \$33,275, \$ 26,212 / \$29,189, \$ 20,090 / \$18,066, \$ 37,790 / \$34,045, \$ 43,394 / \$53,394, \$ 58,496 / \$63,112, \$ 21,524 / \$22,698, \$ 51,155 / \$56,445, \$ 40,721 / \$56,445, \$ 65,485 / \$62,159, \$ 67,562 / \$60,800, \$ 28,767 / \$28,767, \$ 121,478 / \$136,723, Normal / Variance, \$ 4,895 / \$57,531, Variance Percentage / \$4,017, \$ 129,987 / \$129,987, \$ 118,853 / \$121,527, \$ 18,934 / \$17,795, \$ 27,453 / \$33,643, \$ 28,457 / \$28,316, \$ 35,626 / \$34,588. The menu also includes options like Drill, Sort, Insert Metric, Move, Filter On..., Remove from Grid, Remove from Report, Advanced formatting..., Thresholds, Alerts, Edit Links..., Percent To Total, Transformation, Rank, Last Month's, Last Quarter's, New..., \$ 50,387 / \$45,3, \$ 68,752 / \$72,3, \$ 76,640 / \$88,1, \$ 146,186 / \$136,4, \$ 41,761 / \$46,765, \$ 116,812 / \$104,858, \$ 85,675 / \$85,164, \$ 114,455 / \$101,288, \$ 24,620 / \$25,916, \$ 20,106 / \$25,133, \$ 9,181 / \$9,767, \$ 33,741 / \$33,275, \$ 26,212 / \$29,189, \$ 20,090 / \$18,066, \$ 37,790 / \$34,045, \$ 43,394 / \$53,394, \$ 58,496 / \$63,112, \$ 21,524 / \$22,698, \$ 51,155 / \$56,445, \$ 40,721 / \$56,445, \$ 65,485 / \$62,159, \$ 67,562 / \$60,800, \$ 28,767 / \$28,767, \$ 121,478 / \$136,723, Normal / Variance, \$ 4,895 / \$57,531, Variance Percentage / \$4,017, \$ 129,987 / \$129,987, \$ 118,853 / \$121,527, \$ 18,934 / \$17,795, \$ 27,453 / \$33,643, \$ 28,457 / \$28,316, \$ 35,626 / \$34,588.

Figure 15-2 Users navigate to the data they need with a right-click menu.

Prompted reports efficiently personalize the data retrieved through simple questions and answers. More accurate data is initially displayed, reducing the amount of ad hoc navigation needed before exporting the data. If the report is saved before exporting, the user's actual prompt answers are remembered for future

use. User permissions add another tier to ensure data accuracy. Security filters and object access control lists are automatically applied to any report requested by a user. Even reports that are sent by Distribution Services on behalf of a user are subject to this security layer.

Data Volume Exported Varies by Use

The data volume exported from the MicroStrategy BI platform ranges from a single row or cell of data to many millions of rows. Small, manageable datasets are usually exported directly from user interfaces. These are typically composed of a few screens or pages, and can span up to several thousand rows of data. Export actions are embedded in all MicroStrategy interfaces. MicroStrategy Office is designed to handle larger datasets than the other user interfaces.

Certain third-party applications, such as data mining tools, require datasets that are several gigabytes in size. These datasets should be saved as tables in a database using dynamic datamarts, or exported using Distribution Services' bulk export feature.

The amount of data that must be transferred and the total number of distinct export requests affect the time taken to retrieve the data from the data warehouse, apply additional analytical processing, format the data, and transmit the data. The MicroStrategy BI platform utilizes Intelligence Server for efficient processing of export requests. Intelligence Server's component, Distribution Services, manages immediate exporting that is optimized for delivering small datasets quickly, as well as bulk information delivery.

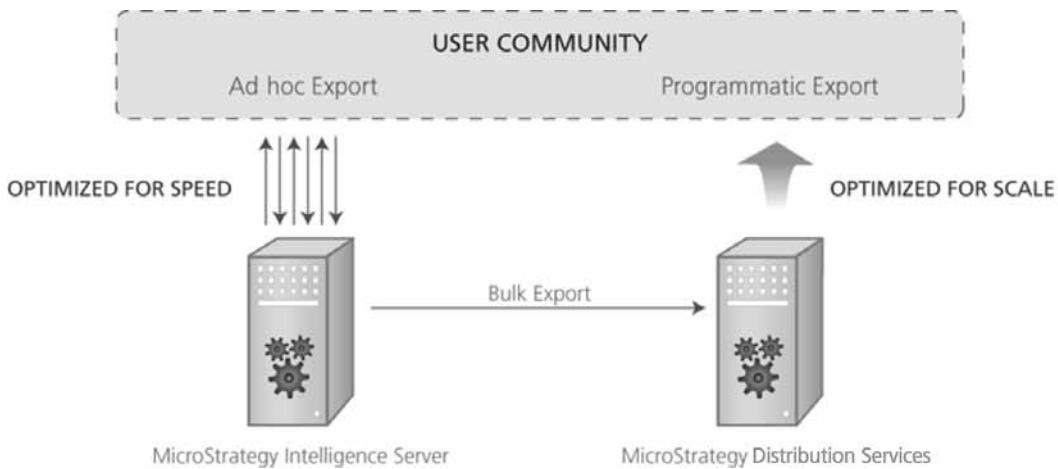


Figure 15-3 Exporting in MicroStrategy is optimized for different use cases.

Report Data is Exported in Many Output Formats

Depending on the use of the data, the required output format varies significantly. Ad hoc offline analysis is often done in Microsoft Excel. Offline review is normally conducted in Microsoft PowerPoint, Microsoft Word, HTML, PDF, and MHT/Flash formats. In most of these cases, the precise formatting of the MicroStrategy report must be kept in the final output. Indeed, MicroStrategy applies the formatting stored in the metadata repository to exported data when required.

When larger amounts of data are exported, detailed formatting of the data itself is generally not necessary. The output file type is of more importance, as large datasets are mostly imported into other applications. File types used for these datasets include database tables, text, and Web services formats.

Distribution Services delivers data and reports in E-mails as HTML or text content, and as PDF, Excel, and text file attachments. The attachments can be compressed automatically before being sent. Distribution Services also delivers reports to printers, and to file systems for offline review and report archiving.

MicroStrategy has invested significant resources into supporting a wide range of output formats for thousands of customers worldwide who use many different RDBMS, and hundreds of thousands of report designs. Along the way, MicroStrategy has addressed significant challenges, including:

- Repeating row and column headers over multiple pages
- Incorporating correct HTML codes
- Optimizing inserts for each supported database
- Handling data formatting issues such as:
 - null values
 - leading zeros
 - negative values
 - spaces
 - double-byte characters
- Inserting page data in separate worksheets

MicroStrategy Office formats data using information in the metadata. This formatting can be changed in the Microsoft documents directly, and will persist if the data is refreshed. MicroStrategy Office supports the widest range of Excel formats:

- Grids
- Charts
- Outlines
- PivotTables®
- PivotCharts®
- Flattened
- Quickgrids

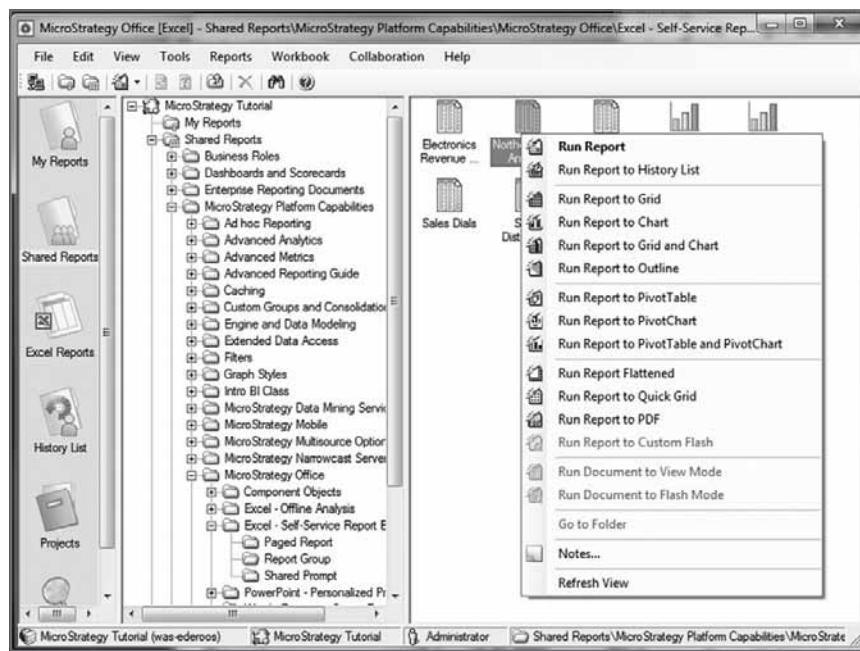


Figure 15-4 MicroStrategy Office gives users many options for running reports with different output formats in Microsoft Office.

Different components in the MicroStrategy BI platform convert report data into the desired data format. For some formats, such as PDF, the formatting can be distributed across more than one component in order to spread the processing across more than one server. Exporting to the following formats is available from MicroStrategy:

Export Format	User Interface
HTML	MicroStrategy Web, Desktop
PDF/Flash	MicroStrategy Web, Desktop, MicroStrategy Office
MHT/Flash	MicroStrategy Web, Desktop, MicroStrategy Office
Excel	MicroStrategy Web, Desktop, MicroStrategy Office
PowerPoint	MicroStrategy Office
Word	MicroStrategy Office, Desktop
Access	Desktop
E-mail	MicroStrategy Web
Web Services	MicroStrategy Office
Data Mart	Desktop
CSV text files	MicroStrategy Web, Desktop, MicroStrategy Office

Figure 15-5 MicroStrategy's user interfaces export report data to many different formats.

15.3 MICROSTRATEGY OFFICE INTEGRATES WITH EXCEL, WORD, AND POWERPOINT

Microsoft Office is the most prevalent desktop productivity suite. MicroStrategy Office runs MicroStrategy reports and documents directly in Excel, Word, and PowerPoint, extending business intelligence to many more people. Simply put, MicroStrategy Office allows Microsoft to be the interface to the business intelligence system with all of the security, robustness, and data consistency of the MicroStrategy BI platform.

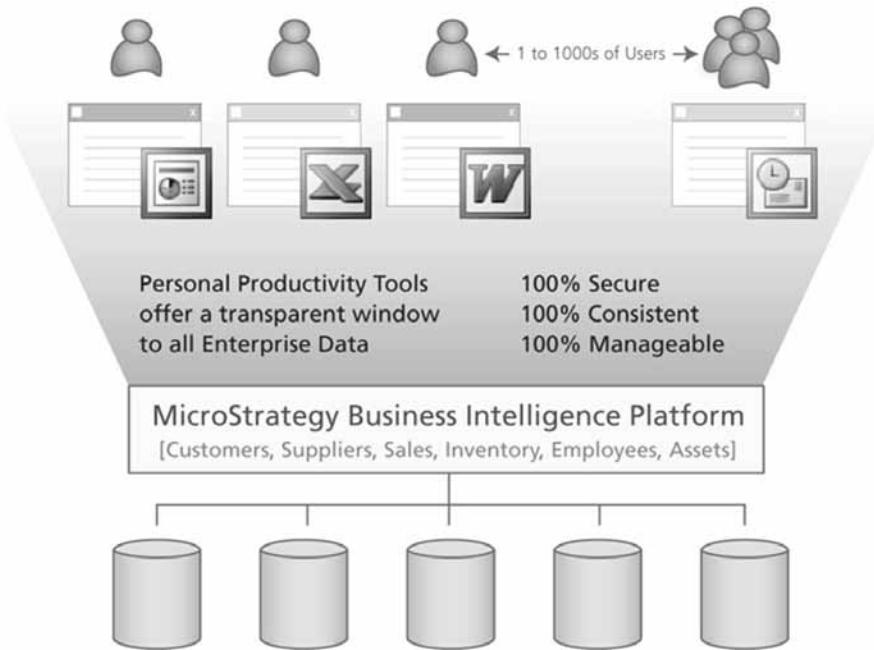


Figure 15-6 MicroStrategy Office runs reports directly in Microsoft Office, extending business intelligence to many more people.

Integration with the office productivity suite is robust, because MicroStrategy Office uses shared Microsoft Office components, adheres to the Microsoft .NET Architecture for Office, and is coupled to the MicroStrategy BI platform using Web Services. Data is transmitted in SOAP packets in XML or CSV formats. The lightweight .NET add-in provides native support for Microsoft Office worksheets, charts, PivotTables, and PivotCharts. The same formatting viewed in MicroStrategy's standard interfaces is applied to reports and documents run from Microsoft Office. Charts can be presented as live Microsoft Office charts or as images from Intelligence Server. Additional formatting added in the Microsoft Office document is automatically retained even when the data is refreshed.

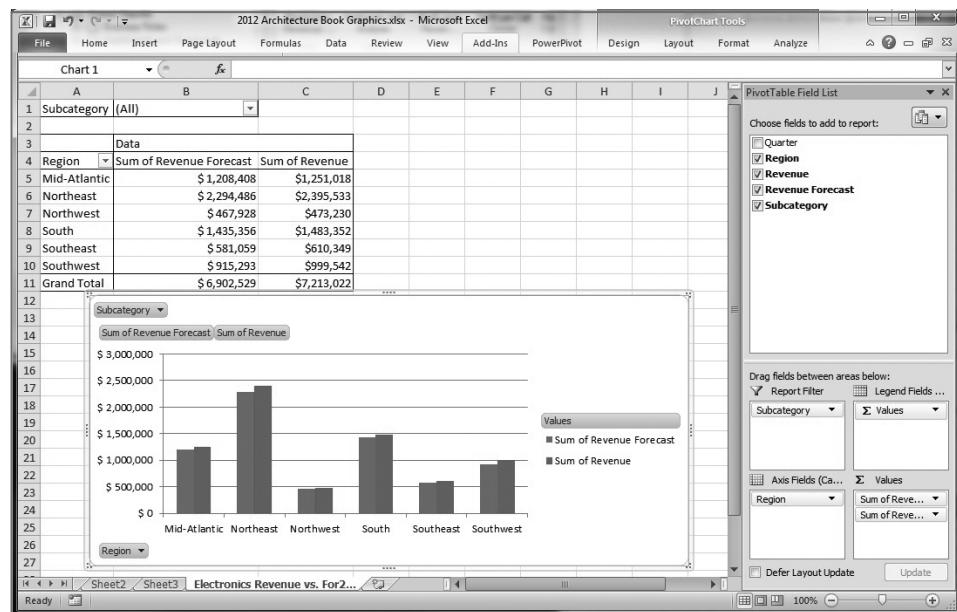


Figure 15-7 A MicroStrategy report can be run simultaneously as a PivotTable and PivotChart. Using the PivotCache, millions of rows can be analyzed offline.

MicroStrategy Office is particularly efficient for exporting workbooks that contain many reports. Persistence data is stored in the Microsoft Office document, and an entire document, potentially containing dozens of MicroStrategy reports, can be completely refreshed with new data with just one click.

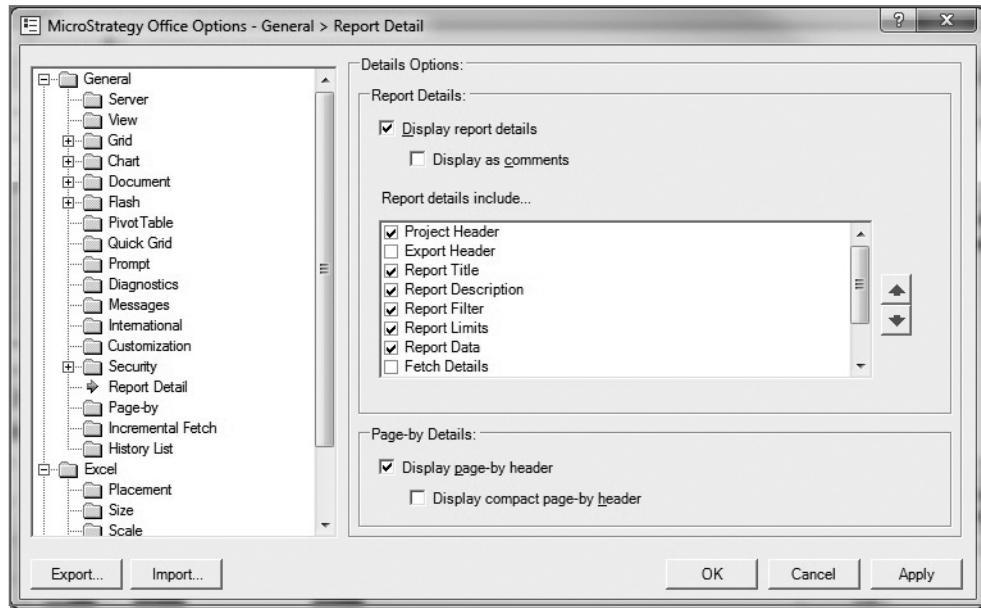


Figure 15-8 MicroStrategy Office options provide the flexibility that the Microsoft Office user requires.

Beyond the simple exporting of reports, the Microsoft Office integration takes advantage of browsing and searching MicroStrategy reports, accessing the History List, and re-prompting. The user can also access many projects in series. All enterprise security enforced by MicroStrategy Intelligence Server is also enforced for Microsoft Office users. For developers interested in extending these capabilities, MicroStrategy Office has a complete programming API.

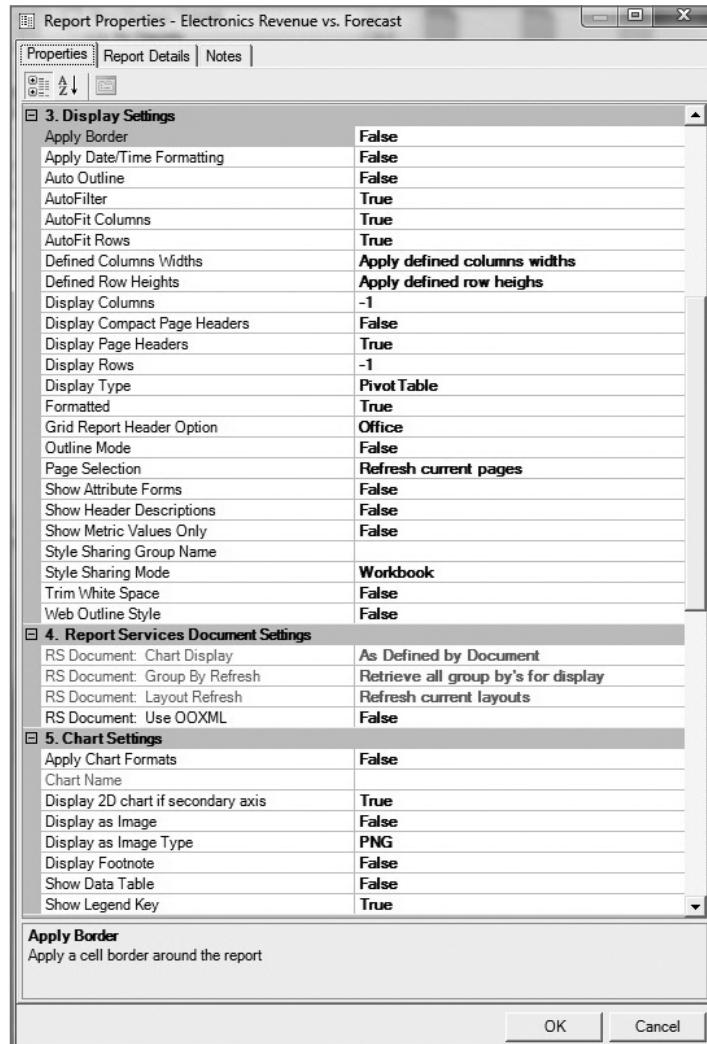


Figure 15-9 MicroStrategy Office persistence data stored in the Microsoft Office document.

15.4 SUMMARY

The MicroStrategy BI platform offers the broadest, most scalable exporting options for extracting analytical information from data warehouses and other data sources. MicroStrategy Web, Desktop, and MicroStrategy Office users can export most report data volumes directly from their preferred interfaces into an output format of their choice. Distribution Services sends report information to E-mail systems, file systems, and printers, and can handle very large files. For the dominant use cases in exporting and offline use, MicroStrategy offers an efficient, robust, and scalable solution.



USER INTERACTIVITY

16

END USER EXPERIENCE

The end user experience with a BI technology is often the most important factor in determining the success of the Business Intelligence application. Business Intelligence applications are commonly accessed by all users within an organization – from executives to business analysts to managers to casual business users.

16.1 CONSISTENT REPORTS, FUNCTIONALITY, AND SECURITY THROUGH MANY USER INTERFACES

The MicroStrategy BI platform has been architected from the ground up to be intuitive for the business user. With consistent interfaces spanning the Web, mobile devices, Microsoft Office, and Windows, MicroStrategy's user interfaces require minimal end user training and lead the industry in terms of end user adoption. Users can move seamlessly between various styles of BI applications without switching interfaces or paradigms, leading to high productivity. MicroStrategy's end user interfaces are also heavily self-service oriented, allowing even casual business users to format and personalize report content. This characteristic drastically reduces iterative interactions with IT developers.

The MicroStrategy BI platform provides a wide variety of interfaces that support all levels of users for any business intelligence application. Since all of the interfaces share the same metadata, users can move from one interface to another without any re-configuration or administrator intervention. A report created over the Web can be viewed in Windows, Office, on a mobile device, and in E-mails without the loss of functionality, security, or object privilege settings.

USER INTERACTIVITY

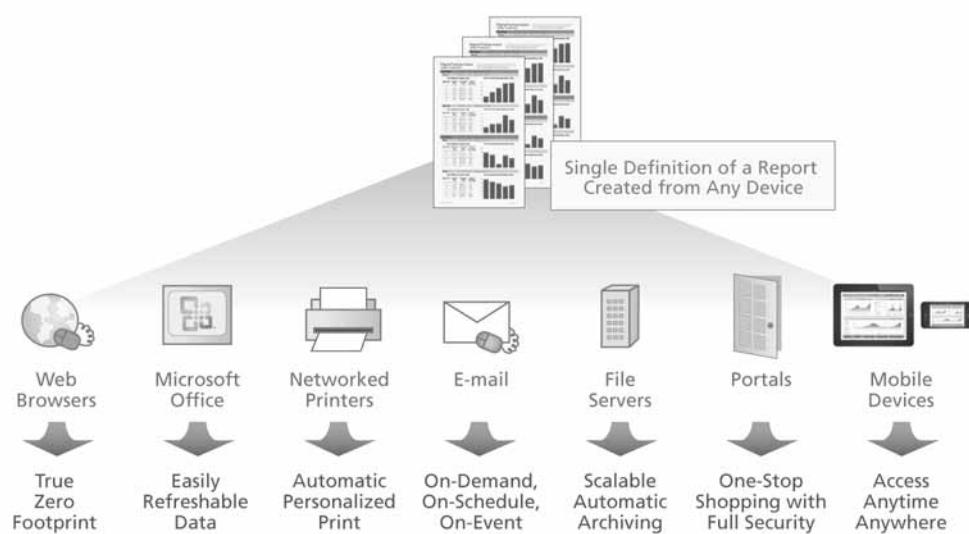


Figure 16-1 Reports are expressed identically through various MicroStrategy interfaces.

The MicroStrategy BI platform architecture includes four primary interactive user interfaces.

1. MicroStrategy Mobile
2. MicroStrategy Web
3. MicroStrategy Desktop
4. MicroStrategy Office

In addition, the platform includes MicroStrategy Distribution Services, which is an interface for distribution to E-mail, printers, and file servers.

INTERFACE	MICROSTRATEGY PRODUCT
Smartphones and tablets	MicroStrategy Mobile
Web Browsers	MicroStrategy Web
Microsoft Excel	
Microsoft PowerPoint	MicroStrategy Office
Microsoft Word	
Windows	MicroStrategy Desktop
E-mail	
Network printer	MicroStrategy Distribution Services
File server	

Figure 16-2 The MicroStrategy BI platform interacts with a number of applications and devices.

One of the fundamental design tenants of the MicroStrategy platform has been to ensure that any report can be run through any interface and have the same look and feel. In addition, user actions such as pivoting, page-by, sorting, and drilling are available across all interfaces. As a result, a user can seamlessly transition from one interface to the next with minimal training. The next sections discuss the various MicroStrategy user interfaces in more detail.

MicroStrategy Mobile

MicroStrategy Mobile is the interface of choice for users who conduct business on the go. It is the most portable method of connecting to MicroStrategy reports, graphs, and dashboards and is appropriate for all levels of users. Any MicroStrategy Mobile user can securely connect to any MicroStrategy Mobile server to access interactive reports and dashboards. With MicroStrategy Mobile, users can browse reports and dashboards, answer prompts, and even drill and filter data. BI administrators can customize the look and feel of the MicroStrategy Mobile homepage so users can easily identify and access the most important information quickly. Using some of the features inherently available on the mobile device, users can also access sensory inputs, such as reading bar codes with the camera and GPS location services.



Figure 16-3 With MicroStrategy Mobile, users can access BI reports through mobile devices.

MicroStrategy Web

The Web browser is often the interface of choice for enterprise applications. MicroStrategy Web is the easiest-to-use BI interface in the industry and was designed to be highly intuitive. MicroStrategy introduced the first interactive Web interface for business intelligence applications in 1996 and today provides the most comprehensive set of end-user functionality over the Web. In fact, few business intelligence products, irrespective of interface, can match the versatility of MicroStrategy Web in terms of report creation, interactivity, formatting options, data manipulation, and ability to integrate with third-party applications.

MicroStrategy Web is suitable for all levels of users from high-level executives, to power users and analysts, to information consumers. All of this functionality is provided through a browser-independent, zero-footprint Web interface that does not use ActiveX components or Java applets.

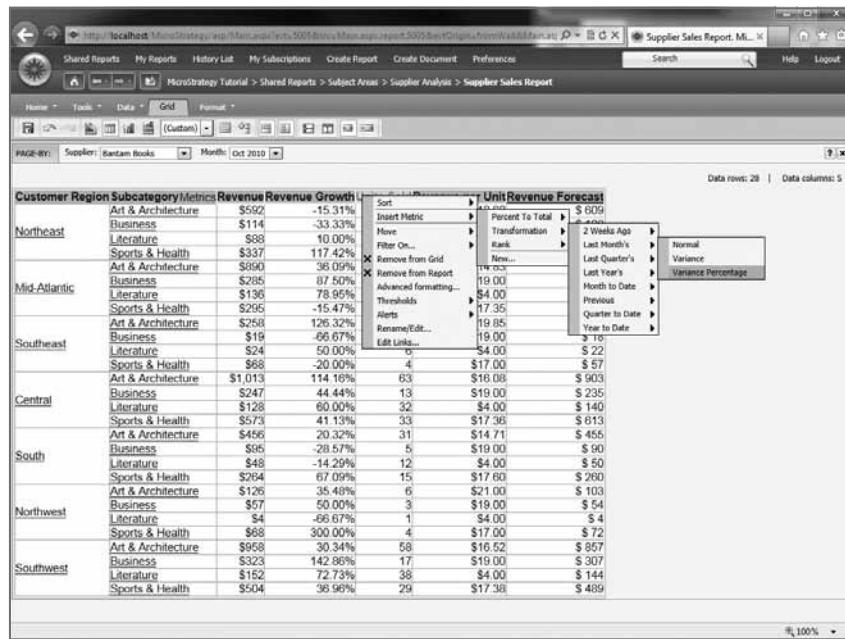


Figure 16-4 Complete report interactivity is available in MicroStrategy Web.

MicroStrategy Web is the most widely deployed MicroStrategy interface, with over 80% of MicroStrategy users accessing their reports and analytics through a Web browser. It has the highest user adoption rate in the BI industry for a number of reasons:

- MicroStrategy was the first BI vendor to introduce a full-function BI Web interface in 1996. This early introduction of the Web paradigm has allowed MicroStrategy to refine Web interactions and make interfaces ideally suited to end users.
- MicroStrategy Web was the first zero footprint multi-browser Web interface. The lack of client dependencies makes it easy to deploy applications to large user populations. Thin client interfaces ensure that security considerations with firewalls are not an issue. MicroStrategy Web achieves a stateless connection without the use of cookies, further simplifying deployments in cases where corporate security policies do not allow cookie storage on user machines.
- MicroStrategy Web is based on the latest AJAX techniques, allowing for an unprecedented level of Windows-like user interactivity through a Web browser. Windows-on-the-Web interactivity includes drag-and-drop, dropdown menus, context sensitive right click actions, mouse over tool tips, and undo/redo actions. As a result, MicroStrategy Web can instantly be used by anyone familiar with internet browsing and Windows application paradigms.

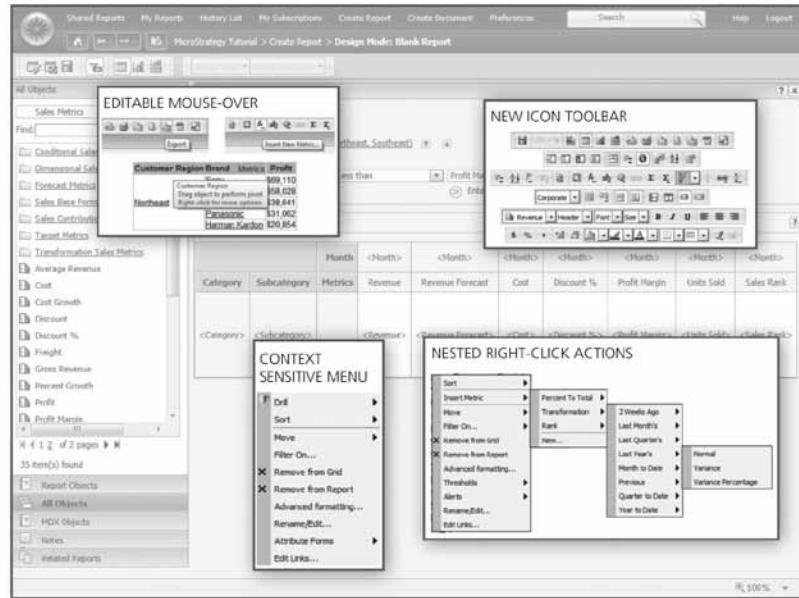


Figure 16-5 MicroStrategy Web's easy-to-use interface is optimized for the business user.

MicroStrategy Office

The MicroStrategy Office interface is designed for business users who are already familiar with Microsoft Excel, PowerPoint, and Word, and need to be able to insert business intelligence into their Microsoft Office files. Once a report is inserted into a file, it can be refreshed from the file itself. All formatting, prompting, security, and other features of the MicroStrategy BI platform are automatically applied and the other parts of the Microsoft file are preserved. As a result, users can take advantage of the best functionality available in both MicroStrategy and Microsoft Office.

The screenshot shows a Microsoft Excel spreadsheet titled 'Category Sales and Profit Performance'. The report contains data for 'Category: Electronics' and 'Subcategory: Computers' across four quarters. The data includes columns for Current Month, Last Month, Last Quarter, and Last Year, along with detailed metrics like Sales, Cost, Profit Margin, and Profit. The report is styled with bold headers and color-coded cells. The bottom of the report shows a summary: 'Quarter Sales \$2,552,210' and 'Quarter Sales (\$30,112)'. The entire report is embedded within an Excel worksheet, with the formula bar showing 'MicroStrategy' and the ribbon tabs including 'File', 'Home', 'Insert', 'Page Layout', 'Formulas', 'Data', 'Review', 'View', 'AddIns', and 'PowerPoint'.

Figure 16-6 BI reports are accessible in Microsoft Excel through MicroStrategy Office.

MicroStrategy Desktop

MicroStrategy Desktop provides a powerful environment for creating BI applications, building the business abstraction layer, designing and running reports and documents, and monitoring and administering the BI environment. End users can build sophisticated reports, modify them on the fly, and drill anywhere for train-of-thought analysis. The reporting flexibility of MicroStrategy Desktop allows users to change any parameter of a report at any time during analysis and save any portion of the result — the complete report, report filter, or report template. Users can also combine reports and filters to generate sophisticated qualifications for set analysis or custom groups. Power users can develop and deploy customized workflows and drill paths for navigating through scorecards, dashboards, and other analytical reports.

MicroStrategy Desktop is the primary designer and development interface in the MicroStrategy platform. It is a Windows-based application that offers full system architecting, application design, system configuration, and administrative functionality.

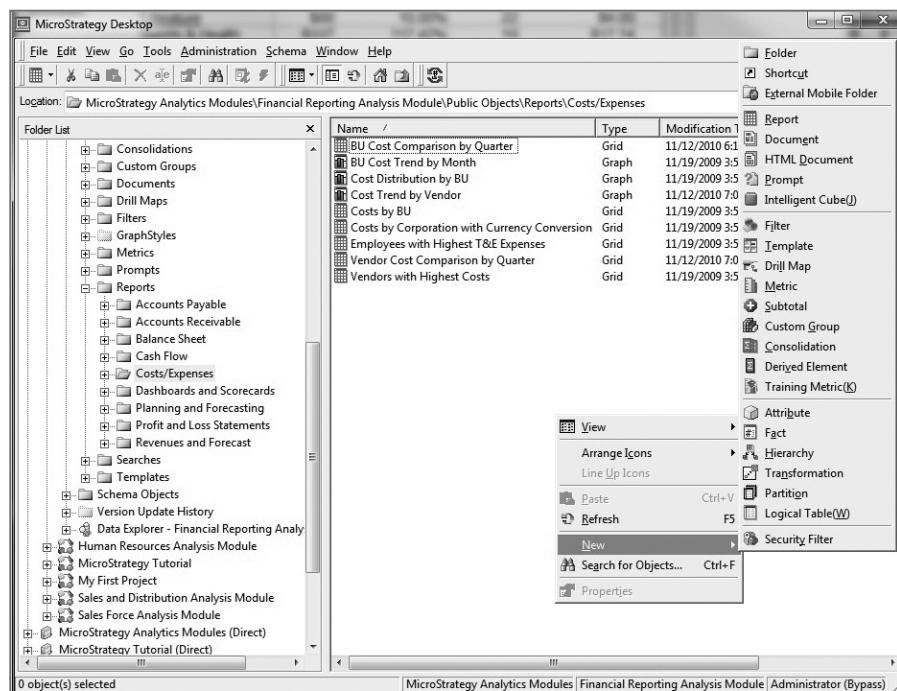


Figure 16-7 Powerful development and analysis capabilities are available in MicroStrategy Desktop.

MicroStrategy Distribution Services

In addition to the four primary interactive end user interfaces mentioned above, the MicroStrategy BI platform features a distribution/subscription interface that delivers information to user-specified devices on an event or time-driven schedule.

MicroStrategy Distribution Services distributes reports and documents via e-mail or to network printers and file servers. These reports can be delivered on a periodic schedule or when an alert condition is triggered; for example, they can be delivered every Monday morning, at the end of the month, or when the inventory for a particular product falls below a predetermined level. Users can subscribe to reports, specifying whether the reports are personalized and to which E-mail accounts or other devices they should be sent. Alternatively, an administrator can schedule report distributions. The common metadata architecture of the MicroStrategy BI platform makes reports that are created in either the Web or the Windows environments immediately available for delivery and subscription in Distribution Services.

MicroStrategy Distribution Services can distribute information to user groups ranging from 10's to 1,000,000's of people while simultaneously personalizing the information delivered to each recipient according to their preferences and security profiles.

16.2 THE BUSINESS INTELLIGENCE APPLICATION SPECTRUM

As business intelligence has evolved from hard-coded back office applications to pervasive enterprise deployments, user functionality has become an essential consideration. Many companies acquired a diverse collection of business intelligence technologies because each technology offered a particular strength in a single area. Broadly speaking, business intelligence applications can be grouped into five application areas, or styles of BI:

1. Data Mining and Advanced Analysis — Fully investigate queries with set analysis, statistical and trend analysis, and data mining
2. Visual and OLAP Analysis — Slice-and-dice analysis with visualizations, drilling, pivoting, and other investigative features
3. Enterprise Reporting — Print-perfect operational and business reports with interactive content
4. Dashboards and Scorecards — Highly graphical reports designed to monitor corporate performance
5. Mobile Apps and Alerting — Business apps on mobile devices and the scanning of data for exception reporting

USER INTERACTIVITY



Figure 16-8 BI vendors have produced different technologies to support the 5 styles of BI applications. Only MicroStrategy delivers all 5 styles of BI within a single unified architecture.

Traditionally, organizations have had to maintain multiple different toolsets to provide users with functionality across these application areas. The MicroStrategy BI platform gives users a single platform and interface from which to access all of these applications and the ability to navigate seamlessly between different application areas. For example, users can view a scorecard formatted for viewing and printing, drill from the scorecard to obtain more information from an OLAP cube, perform advanced statistical analyses, and collaborate with other users. All of these steps are achieved without users having to switch interfaces or applications, allowing them to easily maintain context.



Figure 16-9 Users move seamlessly between any and all Styles of BI in any MicroStrategy interface.

16.3 MICROSTRATEGY USER EXPERIENCE DESIGN TENETS

The MicroStrategy BI platform was designed specifically to eliminate the need for organizations to maintain multiple disparate architectures for BI. The design tenets for MicroStrategy's BI user functionality were simple:

1. Provide the complete range of business intelligence functionality in one platform.
2. Design interfaces that are intuitive to the business user.
3. Provide interfaces tailored to user needs by allocating BI functionality on a user, user role, or user group level without customization or coding.
4. Allow users to seamlessly acquire additional BI functionality as their experience and needs increase.
5. Offer BI functionality uniformly through all popular interfaces — Web browser, Windows, Microsoft Office, mobile devices, and e-mail.
6. Deliver Web functionality through a pure HTML/DHTML browser—without using ActiveX or Java applets—to ensure airtight security and scalability to millions of users.

The MicroStrategy BI platform is the crowning achievement of these goals. On a single platform, users can access multiple styles of business intelligence and seamlessly move from style to style in the same interface. In addition, as users demand more functionality, the BI interface is automatically adjusted from simple to advanced, to fit their BI requirements and permissions.

Powerful WYSIWYG Formatting and Layout Options¹⁰

The MicroStrategy BI platform provides extensive formatting and layout features to ensure that the presentation of reports and scorecards satisfies end user and corporate specifications. Out-of-the-box data formatting options include:

- Font — Size, font face, color, bold, italics, underline
- Number format — Decimals, currency, number of digits, date and time formatting
- Thresholds and indicators — Conditional formatting, stop-lights, gauges
- Grid formats — Borders, cell background, shading for headers, headers format, alignment
- Graph formatting — Number of series, legends, axis, title, series along rows or columns, scale, fonts, graph types
- Visualizations — Visual data display: heatmaps, interactive bubble charts, microcharts, bullet charts, graph matrix
- Text boxes, tables, images — Ability to add text boxes, and insert static images into documents
- Outline mode — Collapse and expand reports at various levels for easy aggregation

¹⁰See Appendix D for a list of charts and visualizations

- Panels and panel stacks —Group visualizations into logical pages with distinct controls
- Selector controls —Interactive data filters using drop downs, check boxes, radio buttons, sliders
- Zone-based reports — Freeform layout of documents with images, text, grids, and graphs
- Banded reports — Hierarchically organized reports with banded groups of information

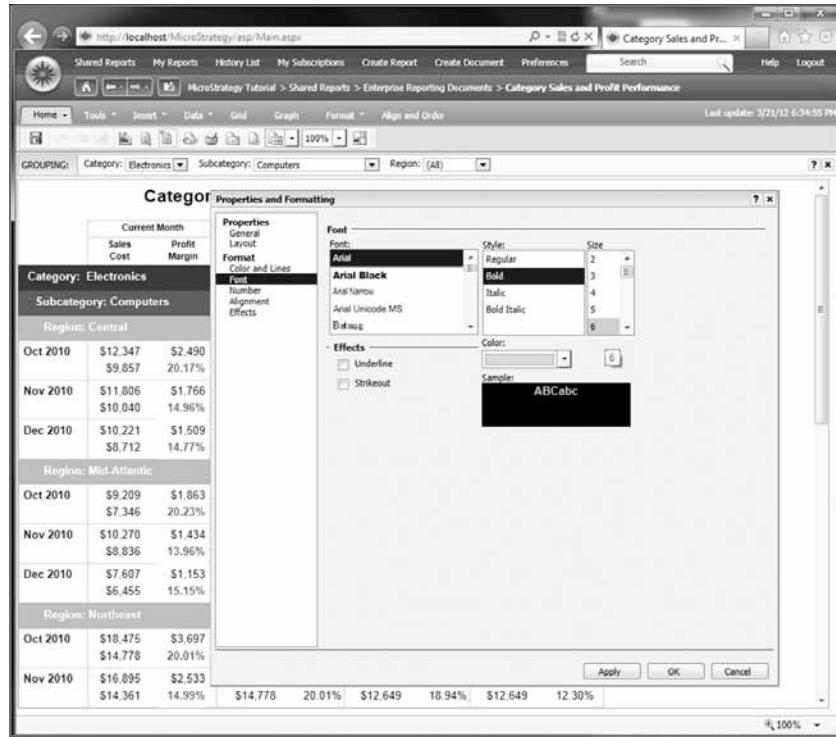


Figure 16-10 Standard Excel-style formatting and functionality make it easy for users to format data.

Intuitive and Comprehensive Report Manipulations

MicroStrategy users can manipulate reports in a number of ways. Users can intuitively access these options through:

- One-click icons in toolbars with tool tips
- Drop down menu option
- Context sensitive right mouse click actions

These options are presented in the graphic below and include OLAP, formatting, export, and collaborative actions.

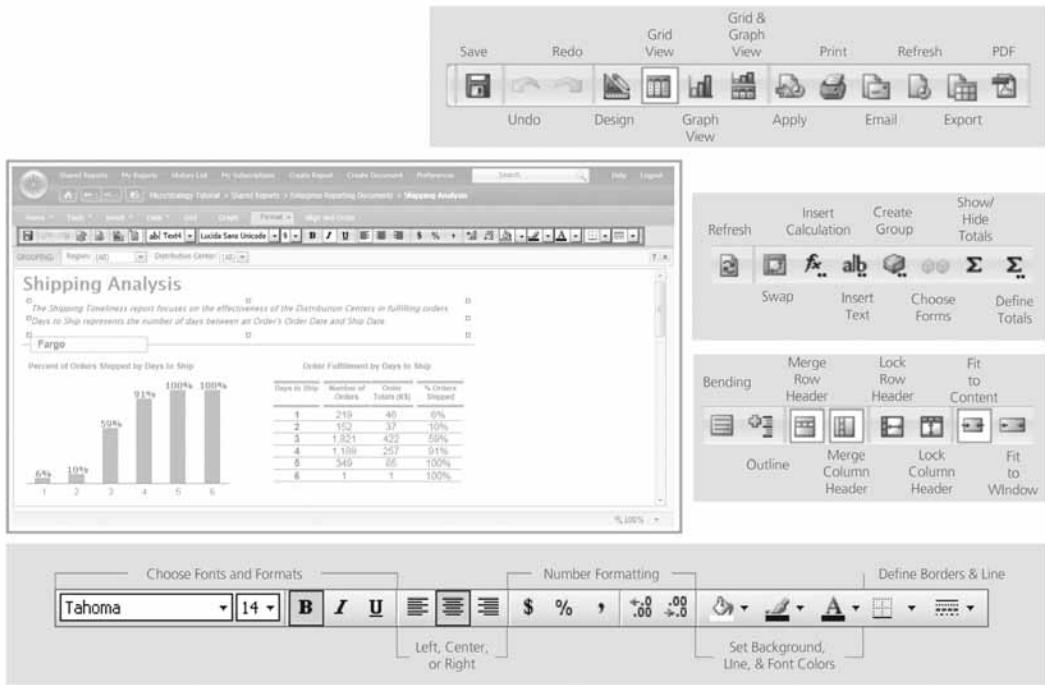


Figure 16-11 MicroStrategy Web contains a large number of report manipulation options.

Extensive Report Export and Collaboration Options

Once a report has been created, viewed, analyzed, modified, or manipulated, a user has several options to save, share, or distribute the information. The report actions supported by MicroStrategy enable enterprises to deploy analytic applications that promote collaboration and increase productivity. These report actions, which are available to both Web and Windows users, include the ability to:

- Print. Print-perfect reports can be printed to any network printer. Users can dynamically change the page layout, apply fit-to-page functionality, and customize headers and footers for any report.
- E-mail. Reports can be sent directly to other users to view and analyze with all the formatting of the original report in a variety of output formats:
 - HTML report image
 - PDF attachment
 - Excel attachment
 - MHT attachment
- Subscribe. Receive reports on a regular schedule or based on an event. Reports can be delivered via E-mail, to a printer, or to a file server.
- Save to personal folder. A report can be saved to a user's personal folder after data manipulation and formatting.
- Save and publish a shared version. Reports can be saved in public folders for use by other business users. All security settings will be applied when other users run these reports.

- Export. The report information can be exported, with or without formatting, to any of the following:
 - HTML
 - Microsoft Excel
 - Microsoft Word
 - Microsoft Access
 - PDF file
 - Text file
 - Flash/MHT
- Schedule. Report caches can be refreshed from the data warehouse at off-peak hours, reducing the frequency of data source access via SQL or MDX.
- Create a datamart table. Automatically save the results from any report to a relational database table. This allows data-level integration with third-party applications like data mining or marketing automation systems.
- Notes. Users can add and edit comments to reports and documents. Each comment is marked with the user's ID and a time stamp.

16.4 DASHBOARDS AND SCORECARDS

Dashboards and scorecards are used to measure and monitor operational performance by presenting key measurement indicators from various areas in an easily understandable manner. Dashboards are visually intuitive and interactive displays of data for monitoring personal or work group contributions to overall goals of the business. Dashboards allow for rapid visual assessment of performance and make it easy for a user to consume a large amount of information quickly. Scorecards help users identify areas to improve corporate performance and often adhere to a major management methodology such as Balanced Scorecard, Activity Based Management, or Six Sigma. With scorecards, users can track key performance indicators (KPIs) and view actual values against targets or benchmarks.

With MicroStrategy, the types of dashboards and scorecards a user can create and interact with are endless. A few examples of dashboards and scorecards a user might create with MicroStrategy are:

- Revenue Performance Management Dashboard – View revenue, profit, and profit margins for each geographic region where the company conducts business. See how current revenue growth measures up against target values.
- Customer Support Analysis Dashboard – Track the status of all customer service interactions and ensure each representative is resolving issues in a timely manner. View trends in the number of interactions over time.
- Product Inventory Analysis Dashboard – Access information about product inventory across all product categories. Track product performance over time. View information such as order count, units received, and end on hand.
- Enterprise Strategy Map Scorecard – Gain insight into all aspects of the business and how each piece affects the others. See all perspectives – Financial, Customer, Learning and Growth, and Internal Processes.

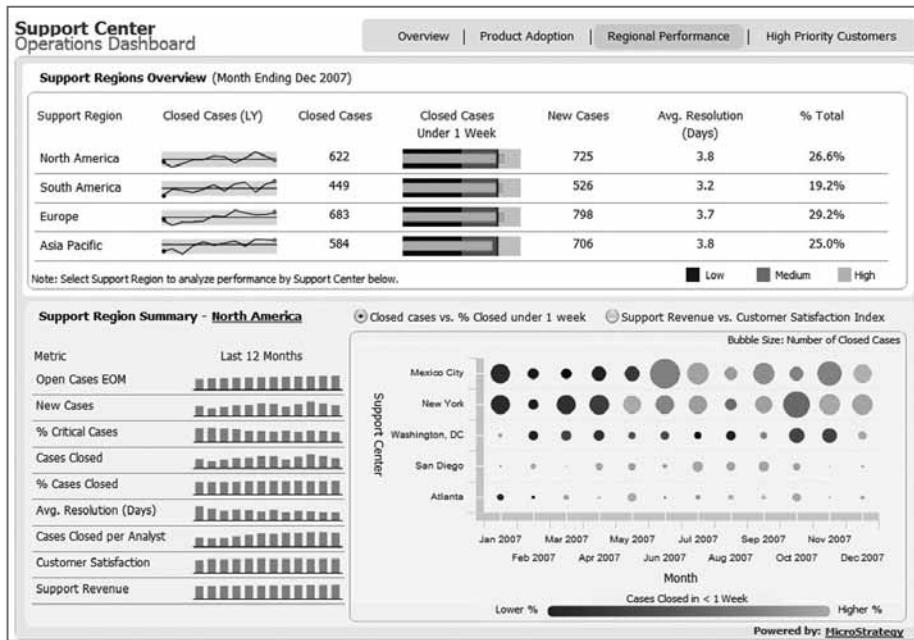


Figure 16-12 MicroStrategy dashboards are a convenient way to monitor operational metrics and key performance indicators.

Fully Interactive Scorecards and Dashboards

MicroStrategy provides full interactivity for all dashboards and scorecards. Business users can rearrange the layout, formatting, and content of any report with simple drag-and-drop actions or by clicking on toolbar icons to get entirely new views of the data without requiring assistance from IT. Users can interact with scorecards and dashboards in many ways, some of which include:

- Performing OLAP-type manipulations such as sorting, pivoting, drilling, and filtering
- Flipping between different “tabs” or pages of information
- Interacting with Flash data visualizations, such as heat maps, that offer even more specialized views of the data
- Using selector controls, such as dropdown menus, to change the information shown
- Viewing and clicking on content from external websites through HTML containers on the document
- Creating custom attribute element groups, or derived elements, on grids and graphs
- Exporting to standalone files

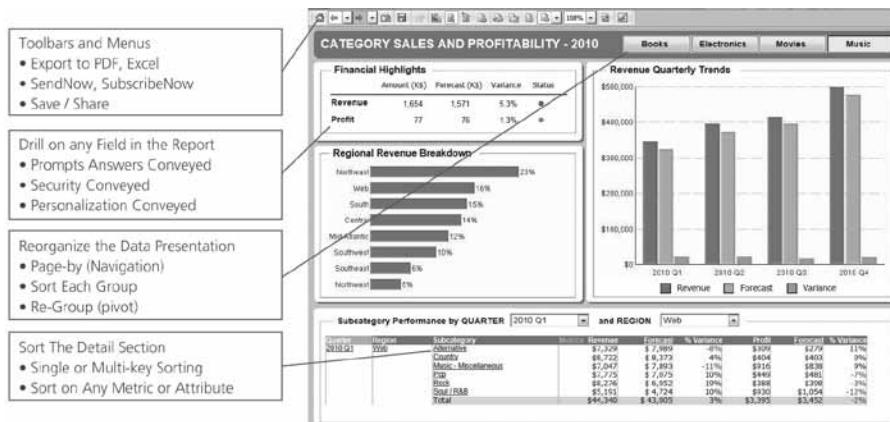


Figure 16-13 MicroStrategy Web offers complete interactive reporting for business users.

WYSIWYG (What-You-See-Is-What-You-Get) Formatting for All Users

MicroStrategy Web extends report design capability beyond specialized report developers to include business users. Business users can design and refine reports in a full WYSIWYG view through a zero-footprint Web interface. WYSIWYG design uses the innate skills needed to design or refine a MicroStrategy report that most business users already possess from their experience with Microsoft PowerPoint or Excel. Business users can edit dashboards to alter content and formatting. They can change the depth of data displayed, modify filtering options, add visualizations, and even make it possible to view information from other websites directly on the dashboard. This provides a radically simpler model for developing reports and makes it possible for average business users to “tune” reports to their exact structure and formatting needs without time-consuming iterations with professional report designers.

With MicroStrategy Web’s WYSIWYG design, available to both business users and report developers, companies achieve drastically faster time-to-deployment for new scorecards, dashboards, and enterprise reports. Report developers can design reports quickly and then send them to business users to add context and details that make the report most meaningful. This saves time for report developers and business users, freeing both to perform their primary job functions most efficiently.

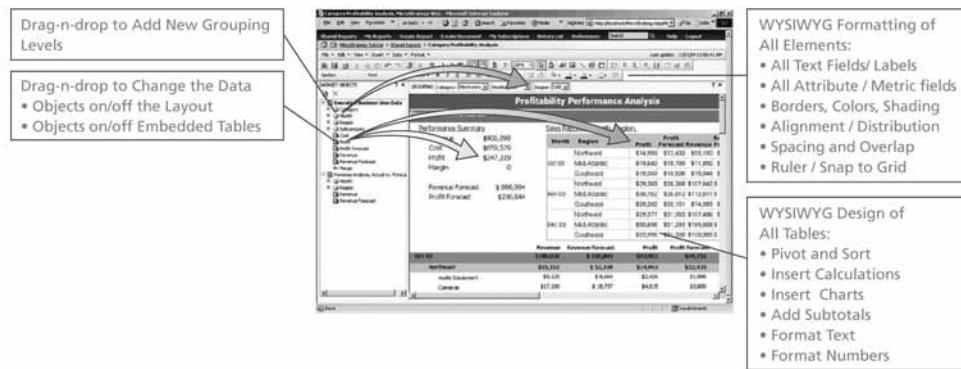


Figure 16-14 Users have full control of the report layout using WYSIWYG design techniques.

16.5 VISUAL INSIGHT ANALYSIS AND DATA DISCOVERY

With Visual Insight analyses, users can explore large sets of business data without requiring dashboard design skills or direct support from IT. These customized, interactive displays give users graphical ways of investigating data through the Web and on mobile devices. Users view and interact with data in one of several powerful visual representations, or visualizations. Visualizations can be interchanged at any time and onscreen menus allow users to manipulate and filter data. In addition, users can save and share their visualizations.

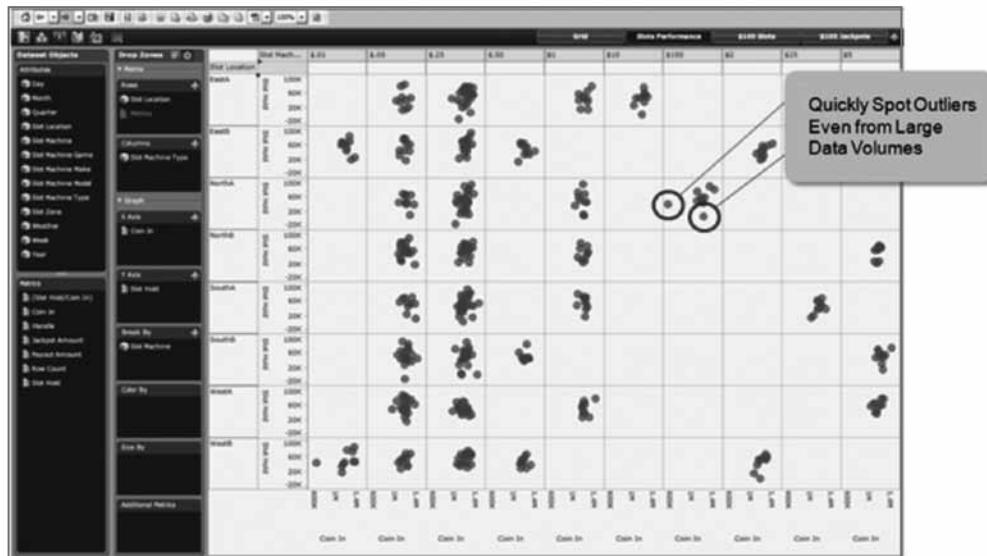


Figure 16-15 MicroStrategy Visual Insight provides users with graphical data investigation in a graph matrix visualization and a visual data discovery and analysis tool.

The different types of visualizations available for analyses are:

- Grid visualization: An interactive grid. Users can pivot, sort, move, drill, filter, and perform additional manipulations on data displayed in the grid
- Graph visualization: An interactive graphical format. Users can choose between a variety of different graphs, such as an area graph, line graph, or pie chart
- Graph Matrix visualization: A chart containing one graph for every combination of the specified data. Users can examine the data for each combination individually
- Map visualization: Geographic locations on a map. Users can change the color, size, and display of markers based on the value of a metric, to allow users to quickly grasp relationships between different locations
- Heat Map visualization: Combination of colored rectangles where each rectangle represents an attribute element and is colored and sized according to the value of metrics in the visualization

16.6 TRANSACTIONAL APPS TO CAPTURE DATA

Transactions allow users to act on their decisions, change data on the fly, and view updated reports. When users input information, it is captured by the Business Intelligence application using MicroStrategy's write-back capabilities. The updated data is then immediately accessible to any report. For example, a store manager may access a BI report to review store inventory and want to make notes about products that must be reordered or discontinued. Using MicroStrategy's transaction capabilities, the manager can make notes directly on the report, alerting anyone running the report that action is required.

MicroStrategy Mobile also makes it possible to imbed transactions into mobile apps. With transactional apps, users can input a wide range of information into the app, such as pictures, text, and location information and it is captured by the Intelligence Server.



Figure 16-16 Users can input data from mobile devices and from Web browsers.

MicroStrategy provides a graphical application development environment to quickly build mobile and Web transactional apps to trigger new business processes, reallocate resources, and accelerate business activity. MicroStrategy uses a wide range of input forms and takes advantage of a flexible drag-and-drop design template. Change or add new data using:

- Text Inputs
- Number Inputs
- Calendar,
- Camera
- GPS
- Switches
- Steppers
- Sliders

16.7 ENTERPRISE REPORTING

Enterprise reporting applications provide users with pixel perfect formats that deliver detailed information as operational reports, business reports, invoices, and statements. Standard operational reports are used extensively in all types of analytical applications. Standard reports are the primary reports that are available in any reporting application. Users often customize these reports to meet their own specifications. The MicroStrategy BI platform provides the highest degree of flexibility in analyzing standard reports, so that a user can tailor a report view to the exact level of information and format required. Sophisticated caching within the MicroStrategy BI platform ensures that users quickly receive their most complex reports at the most granular level of detail necessary.

Category Sales and Profit Performance									
Current Month			Last Month		Last Quarter		Last Year		
Sales Cost	Profit Margin	Sales Cost	Profit Margin	Sales Cost	Profit Margin	Sales Cost	Profit Margin		
Category: Electronics									Quarter Sales \$2,552,210
Subcategory: Computers									Quarter Sales \$208,559
Region: Central									Quarter Sales \$34,373
Oct 2010	\$12,347 \$9,857	\$2,490 20.17%	\$11,273 \$9,148	\$2,124 18.85%	\$11,813 \$9,863	\$1,950 16.51%	\$7,789 \$9,863	\$1,544 19.83%	
Nov 2010	\$11,806 \$10,040	\$1,766 14.96%	\$12,347 \$9,857	\$2,490 20.17%	\$10,210 \$8,275	\$1,934 18.95%	\$7,750 \$8,275	\$1,080 13.94%	
Dec 2010	\$10,221 \$8,712	\$1,509 14.77%	\$11,806 \$10,040	\$1,766 14.96%	\$11,273 \$9,148	\$2,124 18.85%	\$7,090 \$9,148	\$1,143 16.13%	
Region: Mid-Atlantic									Quarter Sales \$27,086
Oct 2010	\$9,209 \$7,346	\$1,863 20.23%	\$7,568 \$6,145	\$1,423 18.80%	\$7,912 \$6,600	\$1,312 16.59%	\$8,763 \$6,600	\$1,753 20.01%	
Nov 2010	\$10,270 \$8,836	\$1,434 13.96%	\$9,209 \$7,346	\$1,863 20.23%	\$6,044 \$4,931	\$1,113 18.42%	\$6,910 \$4,931	\$964 13.95%	
Dec 2010	\$7,607 \$6,455	\$1,153 15.15%	\$10,270 \$8,836	\$1,434 13.96%	\$7,568 \$6,145	\$1,423 18.80%	\$7,366 \$6,145	\$957 13.00%	
Region: Northeast									Quarter Sales \$57,531
Oct 2010	\$18,475 \$14,778	\$3,697 20.01%	\$18,194 \$14,862	\$3,332 18.31%	\$17,196 \$14,382	\$2,814 16.36%	\$14,014 \$14,382	\$2,776 19.81%	
Nov 2010	\$16,895	\$2,533	\$18,475	\$3,697	\$15,605	\$2,956	\$15,078	\$1,855	

Figure 16-17 Densely populated operational enterprise reports are easy to understand with MicroStrategy.

Enterprise reports quickly identify when key performance metrics fall outside a desired range, and how they track over time. The most detailed enterprise reports contain many metrics to gain deeper insight into the status of a business process. Users can use MicroStrategy's drill anywhere capability to fully investigate the data, and discover why a particular indicator has fallen outside an acceptable range.

16.8 AD-HOC REPORTING AND OLAP ANALYSIS

Ad hoc reporting capabilities reduce development and administrative burdens by transferring report creation to the end user. The MicroStrategy BI platform provides extensive ad hoc capabilities for both Web and Windows so users can build, modify, and save reports and dynamically change the attributes and metrics on a report.

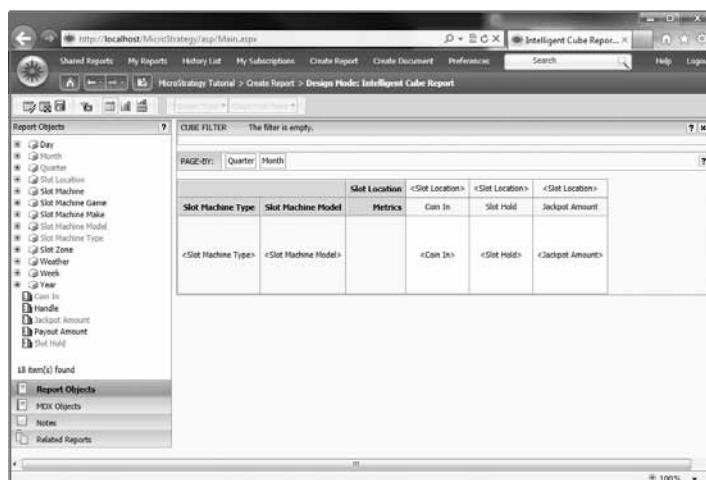


Figure 16-18 Ad hoc report creation over the Web with MicroStrategy.

Report Wizards and Workflows for Guided Reporting and Analysis

Report wizards and navigation workflows in the MicroStrategy BI platform enhance the ease-of-use by guiding users through report creation and through predefined drilling analysis. Users also have the option of building their own custom wizards and workflows. The types of wizards and workflows available are:

- Intelligent Cubes

MicroStrategy OLAP Services enables administrators and users to create sandboxes of analyses consisting of related characteristics, hierarchies, metrics, key figures, and attributes. These are available to users as "Intelligent Cubes" which are stored in memory on the Intelligence Server in a multi-dimensional format. Designers can create a number of Intelligent Cubes for various areas of analysis, such as Customer Analysis or Financial Analysis, and make these available to end users over the Web. Users can then create their own analyses and reports based on these cubes by dragging and dropping the attributes and metrics of interest on to a report. This option allows users to create analyses in a tightly defined area with high performance due to the in-memory nature of the Intelligent Cubes.

- Report Builder Wizard

This multi-step wizard guides a user through the steps required to build a report over the Web. Reports built using the report builder wizard are fully interactive and can be saved, manipulated, shared, and distributed. A report builder wizard can contain any combination of attributes and metrics or be directed to build a specific type of report. A BI application can have any number of report builder wizards. For instance, a financial reporting application may have both a profit and loss report builder and a balance sheet report builder. The profit and loss report builder will limit the user to only income and expense information while the balance sheet report builder will only access asset, liability, and capital information.

- Ad hoc Report Wizard

The ad hoc report wizard allows a user to create and run an ad hoc report over the Web interface by combining an existing report template with an existing report filter. By allowing users to mix-and-match templates and filters from different reports, the ad hoc wizard greatly increases the reporting permutations available to the end user without needing a new report for each permutation. Reports run using the ad hoc report wizard are fully interactive and can be saved, manipulated, shared, and distributed. For example – a list of filters may include top decile of customers by revenue and an available template may include regions and profit. Combining these together creates a report that shows the regional profit for the top decile of customers.

- Customized Drill Paths

The MicroStrategy BI platform provides a drill map editor to define specific drill paths with which to navigate through data in the report. The drill path specifies the attributes, hierarchies, and templates that users can drill to from anywhere on the report for further analysis. This functionality provides users with an entire path of analysis within a given report. The user also has the option of saving a fully interactive report during any point of analysis.

Parameter-Driven Reports

Parameter-driven reports accept user input at runtime to generate a customized view of a report. The MicroStrategy BI platform contains a sophisticated prompting engine that allows the selection of filtering criteria using attributes, such as a time period or a geographical attribute, and metrics, such as top 10% or bottom 10 by revenue. Report content becomes completely dynamic with object prompts. Users can choose the specific attributes and metrics from a list that are available on a report. This results in controlled, guided ad hoc analysis that can be deployed to all end users.

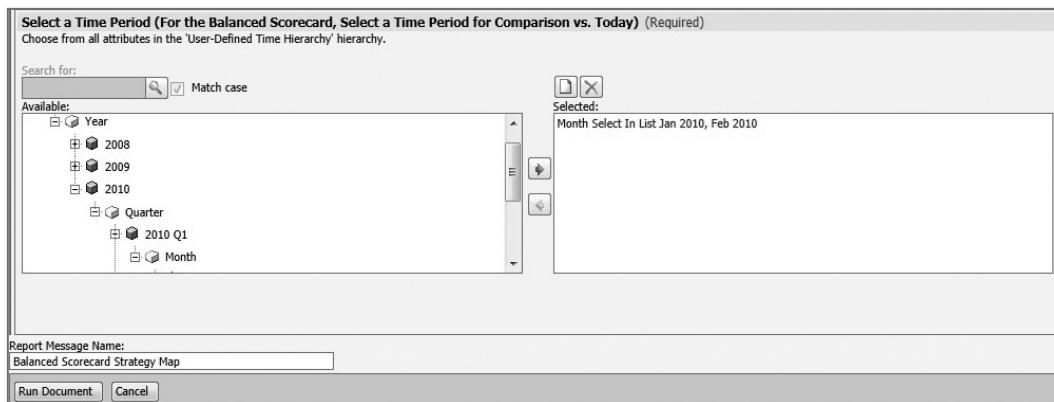


Figure 16-19 Hierarchy Prompts allow users to select from various levels in a hierarchy.

MicroStrategy supports more types of user prompts than any other BI platform. User prompts allow the users to input or make selections at runtime to alter the contents of the report. Simple prompts may allow users to pick from a list of years to filter down time criteria for a report; more sophisticated prompts may allow users to select the level of a hierarchy and the metrics they would like to see on the report. MicroStrategy supports the following set of prompts.

- Hierarchy Prompts allow users to choose elements from a tree-like hierarchy
- List Prompts include shopping carts, radio buttons, check boxes, and multi-select boxes formats
- Object Prompts allow users to choose which objects to place on a report at runtime
- Value Prompts allow users to specify values or thresholds for specific reports
- Level Prompts allow users to choose dimensionality for the metrics on a report

For added convenience, users can save their answers to any prompt and reuse those answers any time the prompt is used on any report, making it easy to answer a series of prompts quickly and reducing the number of saved static reports.

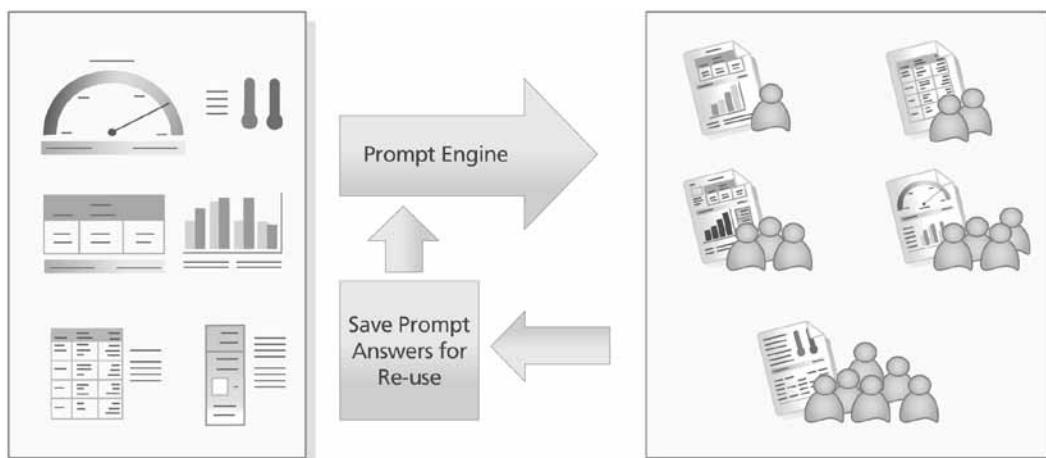


Figure 16-20 Using prompts, MicroStrategy users can create reports that are automatically personalized at run-time.

Data Manipulation for OLAP Analysis

OLAP analysis refers to applications that require users to surf data and make analytical deductions based on investigative reasoning. These applications require users to be able to access large amounts of data in an intuitive and rapid manner. The ability to drill, pivot, and slice-and-dice the information directly from a report is referred to as data manipulation. Data manipulation is central to any investigative business intelligence application since it allows the user to delve deep into a report to identify and discover important trends and patterns in the data. The MicroStrategy BI platform provides an extensive set of data manipulation options to Mobile, Web and Windows users through easily navigable interfaces. Using simple gestures, drag-and-drop, toolbar, and right-click mechanics, users are able to maximize the value of information from any given report. Report data can be manipulated, saved as a new report, and shared with other users.

In many cases, the data needed for these actions are contained in the original report cache and no new SQL or MDX statements are executed in the data source. If any objects are requested that are not in the original report, MicroStrategy Intelligence Server will automatically retrieve that information from the data warehouse and create a new report.

The following data manipulation functionality is available to Mobile, Web and Windows users. Microsoft Office users make use of the functionality built into Excel, PowerPoint, and Word.

- **Sorting**

Specify the sequence of one or more attributes or metrics directly on a report. The sort order, once defined, is preserved through any subsequent report manipulation steps. MicroStrategy supports single- and multi-key sorting on both columns and rows for complex reports with many attributes and metrics. For example, on a customer sales report, users may first want to sort on customer region and then order the customers by revenue within those regions. This is an example of a multi-key sort.

- **Pivoting**

Re-arrange the order and placement of attributes and metrics on both tabular and cross-tabular reports. Users can pivot their data between rows, columns, and pages using either drag-and-drop or single click actions. Pivoted reports can be saved in their new formats by the end user. For example, a report may have sales information with regions and years in columns. Users can switch or pivot the years to rows for a cleaner layout.

- **Page-by**

Create subsets of a report grouped by specific data elements. The page-by functionality is useful when viewing reports with many attributes and metrics. For example, a user can page-by region and month on a report to view specific information for a particular region and month. The user is then able to switch between different region and month values to change the focus of the report. MicroStrategy does not limit the number of attributes or combination of attributes that can be used with page-by to create the desired subsets of data.

- **Drill Down**

Iteratively increase the level of detail of a summary report. MicroStrategy makes it easy for end users to view granular level information contained in a report. For example, a user can drill down from a report at the quarter level to month, week, or date in a single step.

- **Drill Anywhere**

Perform train-of-thought analysis on any report. The drill anywhere functionality does not limit the drill path to a specific dimension in a given report. For example, a user may be viewing a sales report at the regional level. The drill anywhere functionality will allow the user to drill from a given region to the supplier to view vendor performance in each or all regions. Drill anywhere is a powerful feature that enables true 360-degree analysis of the information in a report down to the most granular level of data.



Figure 16-21 Data manipulation — drill anywhere functionality on graph and grid reports for train-of-thought analysis.

- Totals and subtotals

Add totals and subtotals to metrics on a report. Totals and subtotals are supported for all types of metric calculations, including averages and percentages and can be calculated on both rows and columns. In the case of averages or percentages, the total is calculated using the formula of the metric rather than simply summing up the rows and columns. When totaling, users can choose from a variety of functions to calculate the subtotals and grand totals for the original metric calculations. New subtotal functions can also be added to the BI application. The built-in subtotal functions are:

- Average
- Count
- Geometric Mean
- Maximum
- Median
- Minimum
- Mode
- Product
- Standard deviation
- Sum
- Variance

- Derived Metrics

Create new metrics on the fly using existing metrics in the report. Functions available to users to build these on-the-fly metrics include common arithmetic operations and functions to calculate moving averages, variances, standard deviations, and ranking operators. This functionality makes it possible for users to add a percent-to-total metric or a forecasting metric to a report. The new metrics are saved as part of the report definition for future use.

- One Click Metrics

Metrics, such as ranking and percent to totals, as well as time series analysis metrics like “last year’s revenue” and “year to date revenue”, can be quickly added to any report through a simple right click menu action.

- Derived Elements

Create custom combinations of attribute elements on the fly. This functionality makes it possible for users to spontaneously combine different elements of an attribute and use that combination in filters, groups, lists, and calculations. For example, a user may see profit information for the months of March, April, and May in a report and using derived elements combine these months into a group called “Spring”, providing a whole new perspective of the data.

- Quick Data Filtering (View Filters)

Large reports and analyses often contain many rows of information with many measures to analyze. View Filters allow users to quickly narrow down large sets of information based on criteria specified by the user. For example, users looking at a report with sales information for all regions may only want to see the north region information where revenue is greater than forecast by 20%. Users can quickly create a combination of conditions to get this information.

- Lock Row and Column Headings

For large reports, users can lock the row and column headers so that they do not lose context as they scroll vertically or horizontally.

- Switch to Grid, Graph, or Grid and Graph Views

Users can quickly switch views between grid and graph views or display both grid and graph information for a report at the same time.

- Fit to Contents or to Window

Users can choose how they would like report data to be presented on the screen. Fit to Contents shrinks the columns to the standard width, whereas Fit to Window maximizes the column width to fill the screen.

- Seamless Integration of OLAP Analysis into Scorecards and Dashboards

Through years of focused design effort and the development of proprietary algorithms and patents, MicroStrategy's highly refined relational OLAP engine meets the most demanding BI requirements.

MicroStrategy Web makes this same powerful analytic capability available directly from every report. It delivers analytic ability to business users in two ways:

- OLAP analysis can be performed directly on grids embedded in scorecards, dashboards, documents, and reports.
- Drilling is possible from an enterprise report, scorecard, or dashboard to a dedicated view that is optimized for conducting detailed analysis from the same unified Web interface.

In both cases, the integration of reporting with analysis is automatic and entirely seamless, enabling self-service reporting, analysis, and monitoring by all business users. Business users can answer their own follow-on questions as they occur, without need of further IT department support.

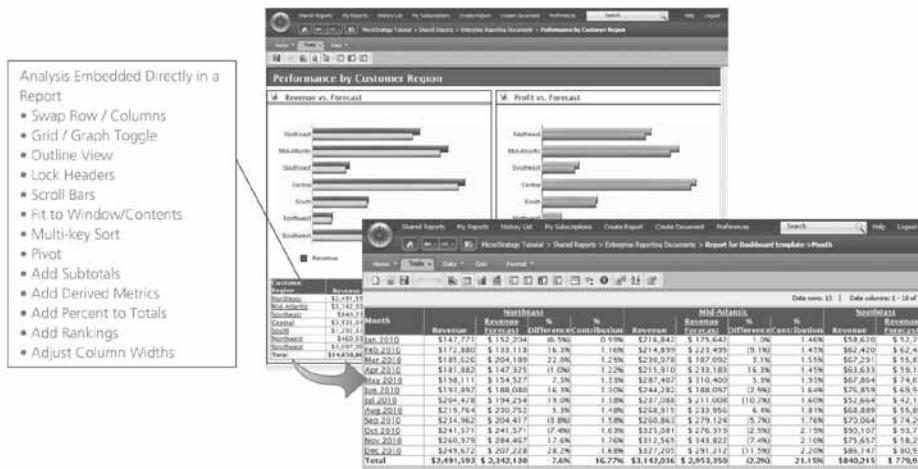


Figure 16-22 Further analysis is possible on any report, scorecard or dashboard.

16.9 ADVANCED AND PREDICTIVE ANALYSIS

Advanced analysis and data mining reports deliver complex calculations and predictive metrics to all end users, enabling more accurate data-driven decision making. Metrics using built-in statistical, mathematical, financial, and data mining algorithms add analytic richness to any BI application. These reports and analyses are available in all MicroStrategy BI interfaces without any additional programming or knowledge of how the functions and algorithms work.

MicroStrategy offers several predictive models for use within any report. MicroStrategy can create predictive models in addition to users importing their own. Users can create these predictive models with MicroStrategy:¹¹

- Linear Regression
- Exponential Regression
- Logistic Regression
- Tree Regression
- Decision Tree
- Clustering
- Time Series
- Association Rules

While MicroStrategy does not support the creation of the following predictive models, users can import and use them:

- Neural Networks
- General Regression
- Ruleset
- Support Vector Machine

¹¹See Appendix C for a full list of functions available in MicroStrategy

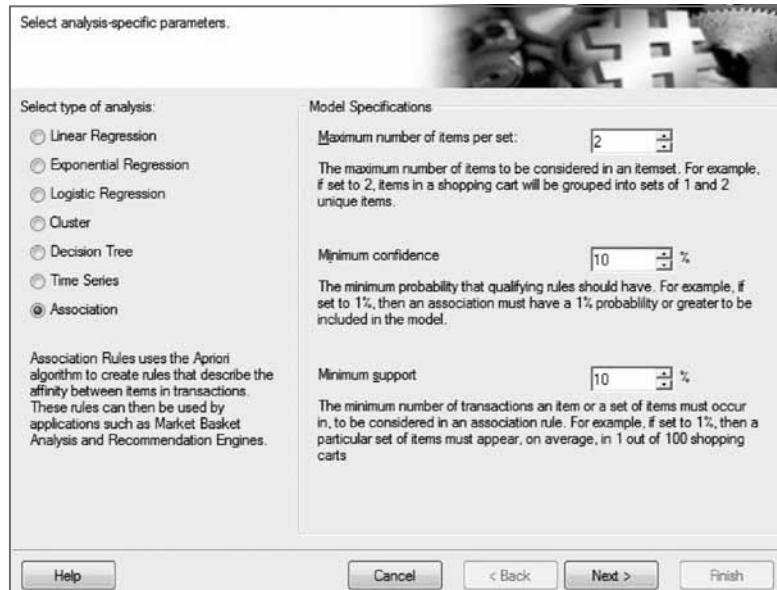


Figure 16-23 MicroStrategy can create, train, and deploy predictive and data mining models.

16.10 ALERTS AND PROACTIVE NOTIFICATIONS

Collaboration applications integrate corporate-wide information so that all users, both within and outside an enterprise, have the same version of the information at all times. Collaborative analytic applications must therefore account for varying levels of user expertise, multiple interface and interactivity requirements, and the need for on-demand and proactive information delivery. The MicroStrategy platform provides users with an extensive array of interface choices and information delivery options. MicroStrategy's sophisticated reporting capabilities also allow end users to perform complex collaborative tasks, such as analyzing information, exporting data to other third-party applications, writing back to the database, and sending e-mail alerts. All of this can be done through a single user interface. Collaborative applications can operate in 2 modes:

1. Administrator-led push applications
 - Information is distributed to large user populations via print, E-mail or file servers, with each recipient automatically receiving personalized information based on their security profiles. Users receive information either on a time or event-driven schedule.
2. User-led proactive applications
 - Subscribe to reports to be sent to them via E-mail or to a printer or file server based on a schedule or event.
 - Send reports to other users via E-mail, print or file servers.

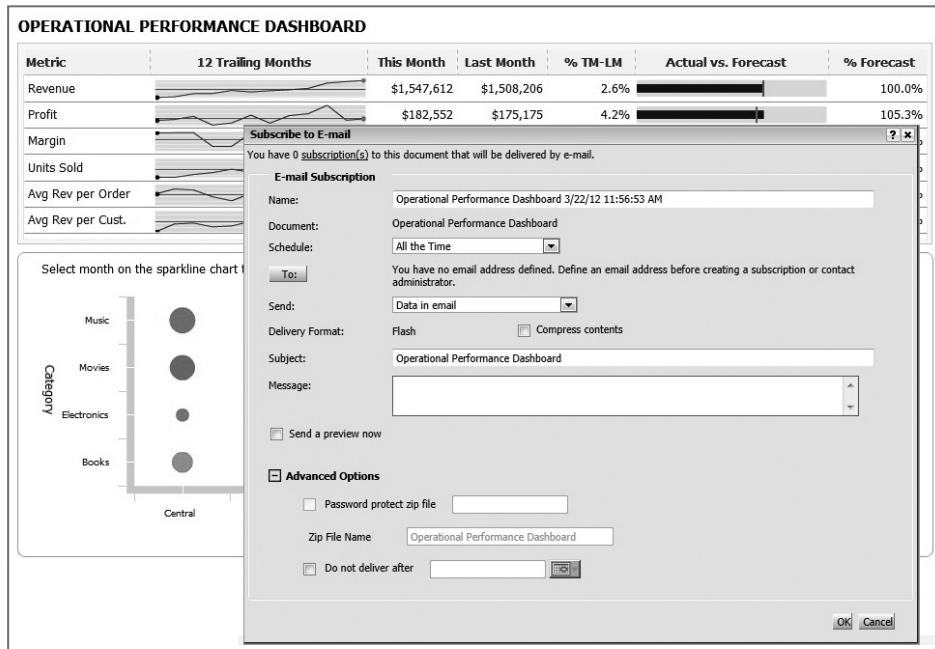


Figure 16-24 MicroStrategy Web allows the subscription of reports using predefined schedules.

16.11 USER FUNCTIONALITY BY USER ROLE

In addition to growth in the number and types of business intelligence applications, as detailed above, the number and types of users of these applications has expanded. Architectures for enterprise business intelligence must be able to accommodate the needs and functionality of all types of users, from analysts and power users to information consumers, in a scalable and manageable manner that does not require extensive customization or coding. Furthermore, reports and analyses developed by any user must be shareable.

The Business Intelligence User Spectrum

The main types of users and their relative distribution amongst the user population in a typical enterprise are shown in the diagram below:

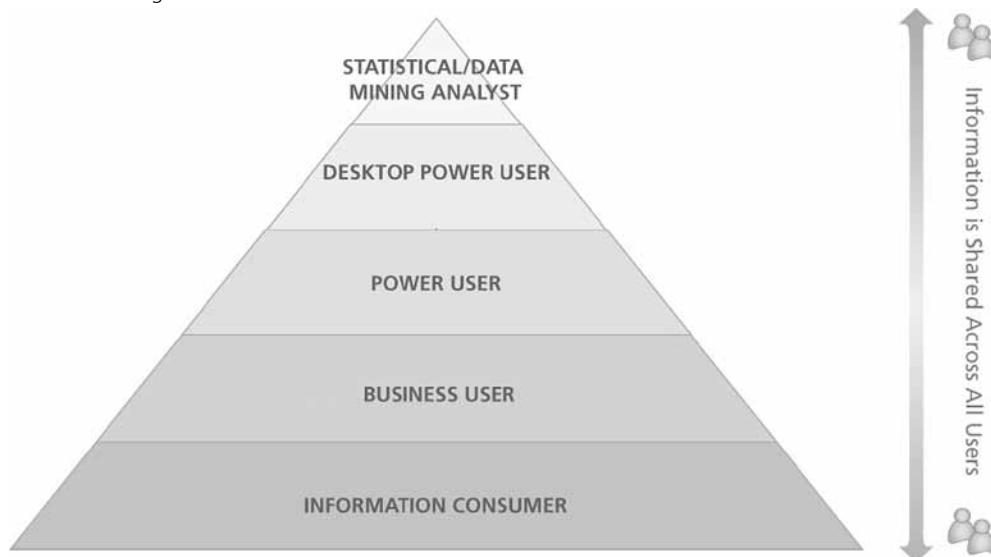


Figure 16-25 The number of BI users in an enterprise is defined by the functionality they use.

1. Information Consumer

This user receives static information either via mobile devices, the Web or via scheduled delivery to E-mail. The information is well-formatted, and designed to be consumed quickly and easily, with little interactivity.

2. Business User

This user uses a mobile device or Web interface to run reports and documents. The reports allow some OLAP actions, such as pivoting, sorting, and drilling, and can be run from or exported to Microsoft Office applications. The ability to reformat and edit reports in a WYSIWYG manner is often also required.

3. Power User

This user runs both standard and ad hoc reports and documents from MicroStrategy Mobile and MicroStrategy Web, and performs advanced OLAP actions such as drilling anywhere to transaction-level data, adding time-series analysis metrics to reports, and saving documents and reports that other users can consume.

4. Desktop Power User

This user needs advanced functionality that defines the metadata objects used to build documents, reports, and analyses. This includes creating sophisticated metrics, arranging data into custom groups to satisfy particular groupings, and defining filters for set analysis.

5. Statistical / Data Mining Analyst

This user develops advanced statistical and data mining models, using neural networks, regression, and clustering algorithms, to create scoring and confidence predictive metrics. These predictive metrics can be incorporated into any document or report, and are available to all users.

MicroStrategy provides the functionality to satisfy all reporting requirements for all BI users. More importantly, BI administrators retain a very precise degree of control over exactly what functionality is given to a specific user, a group of users, or a user role. Users from two different groups can access the same report through any interface with the precise functionality that is appropriate for each user. For example, a user from the "Business User" group might have more limited interactive capabilities than a user from the "Analyst" group. This protects the BI system, the database, and the users themselves from unnecessary complexity while still providing the full range of easy-to-use interactivity to those users who benefit from it. Even within MicroStrategy's end-user products, BI administrators can further adjust and personalize an individual user's, security role's, or user group's access to MicroStrategy functionality.

User Functionality Governed by Privileges

The MicroStrategy architecture allows all of MicroStrategy's interfaces to be customized to the target user population, from novice to expert, through extensive use of privilege controls. Users or groups of users can be assigned varying levels of functionality such as pivot, print, and send now on a per-application or project basis. In addition, users can be assigned security profiles to ensure that they can only see the data and objects in folders that they can access. Thus, without writing a single line of code, administrators can configure an application and user interfaces to suit all user populations.

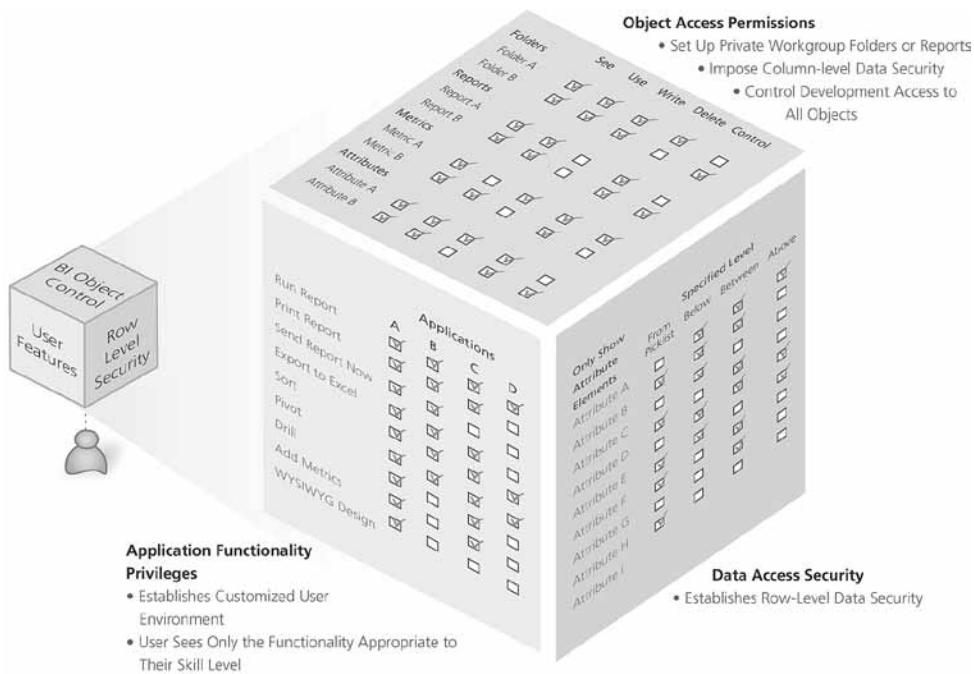


Figure 16-26 User functionality is managed through application privileges, object access permissions, and data access security.

Personalized reports are generated from a single report definition so that report contents are restricted to information that an end user is allowed to see. For example, the same Sales Report will show only West Region details to the Western Regional Manager and show only East Region details to the Eastern Regional Manager. This personalization is applied in on mobile devices, Web, Windows, Microsoft Office interfaces, and when users subscribe to a report, ensuring both data security and delivery of relevant information to the user.

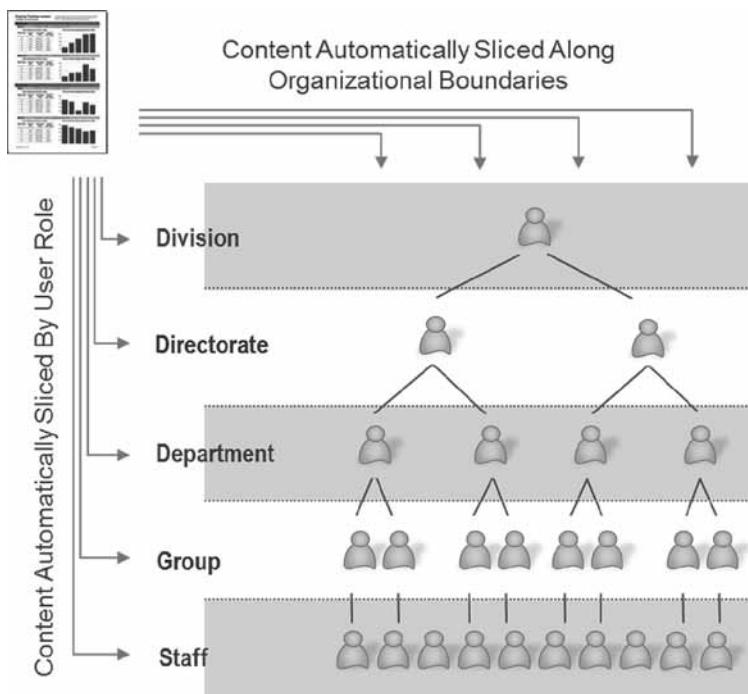


Figure 16-27 A single MicroStrategy Report Is automatically personalized using role-based filters, prompts, and security privileges.

16.12 CUSTOMIZING MICROSTRATEGY FUNCTIONALITY FOR THE END USER

Although the MicroStrategy BI platform provides a comprehensive set of analytical features and functionality through a single interface, enterprises can add their own custom functionality. MicroStrategy's open architecture and APIs, which are documented extensively in the MicroStrategy SDK, make it especially suitable for developing custom functionality or integrating with other applications. The SDK describes how to integrate with enterprise portals, external security systems, and Web services applications. The goal of the MicroStrategy platform architecture is to enable enterprises to fully leverage their entire information assets across the enterprise.

16.13 SUMMARY

The MicroStrategy BI platform has been architected from the ground up to provide an intuitive, consistent, and easy-to-use experience for users. The hallmarks of the MicroStrategy BI platform for user experience are:

- Consistent and intuitive user experience across all interfaces including MicroStrategy Mobile, MicroStrategy Web, MicroStrategy Desktop, MicroStrategy Office, and E-mail
- Support for all 5 styles of BI and seamless user navigation between any of these styles of BI within the same interface
- Powerful distribution and subscription reporting based on alerts or schedules
- Interfaces governed by privilege settings to easily configure for all users ranging from novice to power users
- Comprehensive reporting, OLAP, and formatting options from all interfaces to allow users to create, edit, and modify reports

17

DEVELOPER EXPERIENCE

A key component to building successful BI applications is having a robust set of developer tools to enable developers of all levels to create reports and other metadata objects rapidly, intuitively, and in a scalable, manageable manner.

17.1 MULTIPLE INTERFACES FOR BI DEVELOPMENT

The MicroStrategy BI platform offers several interfaces and tools for the BI developer; all tied together with a secure and object-oriented metadata. The following products and interfaces are available to create and maintain BI applications:

1. MicroStrategy Architect – Used to map data models (in relational databases, SAP, and other cube databases) to business models and to create attributes, facts, and hierarchies.
2. MicroStrategy Desktop Designer – Windows-based design interface used to develop filters, metrics, reports, prompts, consolidations, and custom groups.
3. MicroStrategy Web Professional – Thin-client browser interface that allows users to create reports, dashboards, and scorecards over the Web using an intuitive drag and drop interface. Filters, prompts, and custom groups can also be defined in Web Professional.
4. MicroStrategy Distribution Services – Batch and alert reporting product that allows developers to create exception and alert-based applications to deliver information via portals, E-mails, file servers, and network printers.
5. MicroStrategy SDK – The MicroStrategy SDK (Software Developer Kit) allows developers to customize the look, feel, and content of the MicroStrategy interfaces and extend the functionality available within the platform.

USER INTERACTIVITY

17.2 MICROSTRATEGY ARCHITECT

MicroStrategy Architect is a BI development tool that maps the physical structure of a data source to a logical, object-oriented model of the business through an intuitive, graphical interface. Architect provides a unified environment for creating and maintaining business intelligence applications. BI architects use Architect to:

- Map the physical database schema into a logical business model using wizards and editors
- Model the application using business processes and terminology
- Build complex hierarchies that reflect the relationship between business entities using one-to-one, one-to-many, and many-to-many relationships

The business rules and schema abstraction defined using MicroStrategy Architect are stored in the central metadata repository, allowing MicroStrategy BI products to reuse the definitions and build upon them. These are stored as schema abstraction objects in the metadata repository.



Figure 17-1 MicroStrategy Architect is used to create the abstraction of the data sources used by the BI applications.

Object Creation in MicroStrategy Architect

MicroStrategy Architect provides a graphical interface that enables developers to visually and interactively build data abstraction objects. To ease application creation, MicroStrategy Architect provides a Heuristics Engine that enables automatic recognition and object creation. The Heuristics Engine uses database table information such as primary and foreign keys, cardinality, and data-types along with predefined rules to automatically identify and define objects. By connecting to the Warehouse Catalog and simply dragging and dropping tables into the project space, developers can easily begin to map the system dimensions.

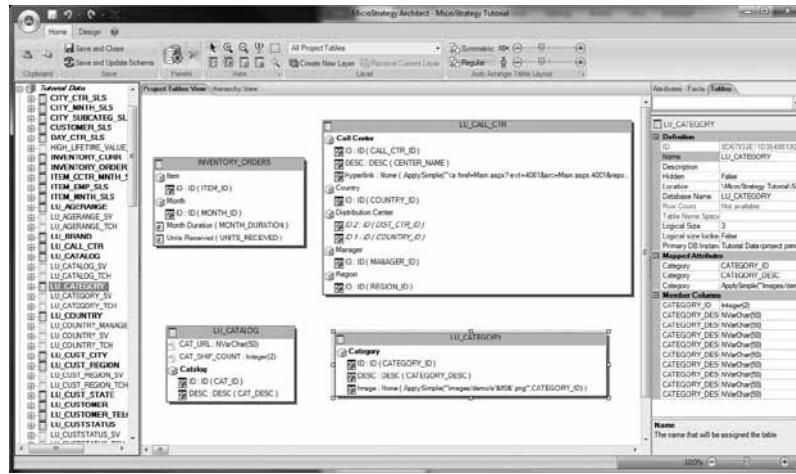


Figure 17-2 MicroStrategy Architect data sources to the business model used by higher level metadata objects.

In MicroStrategy Architect, developers access tables that are visual representations of the tables in the underlying data source. Using these tables and the Heuristics Engine, MicroStrategy Architect automatically recognizes and defines data abstraction objects.

The following are the data abstraction objects that can be created using MicroStrategy Architect:

- Attributes
Attributes define descriptive information in the data, such as Customer Name, Product Category, or Current Date. Architect defines Attributes based on database table column names. Developers can also specify the parent-child relationship between attributes as well as create expressions based on multiple data fields. These are discussed in further detail in later sections.
- Facts
Facts are used to define measures such as Order Amount, Number of Customers, or Current Inventory on Hand. These are usually based on numerical data sources. Facts are mapped to different database tables, providing automatic aggregation awareness. Similar to attributes, users can create sophisticated expressions to define complex facts from multiple data source inputs.
- Hierarchies
Hierarchies are used in the MicroStrategy environment to logically group attribute elements into areas of commonality. Typical examples include the Time Hierarchy, Sales Organization Hierarchy, or Product Hierarchy. There are two types of hierarchies in the MicroStrategy environment:
 - i. System Hierarchy – This hierarchy is created automatically when parent-child relationships are defined between attributes. The system hierarchy mirrors the structure of the data in the data source.
 - ii. User Hierarchies – These hierarchies can be created by users independent of the data source structures. Attributes can be grouped together to form logically coherent groupings based on business models to simplify reporting for end users. There is no limit to the number and sophistication of user hierarchies.

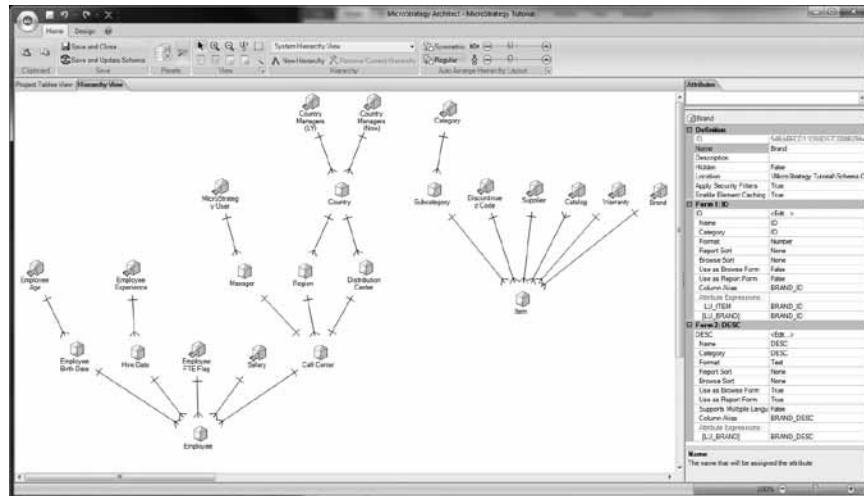


Figure 17-3 Hierarchies can be viewed graphically in Architect.

- Transformations

Transformations are a powerful way to calculate offset values between sets of attributes. For example, architects can build time transformations such as “Last year’s Revenue” or “Quarter to Date Revenue.” These can then be applied to metrics by end users with a single click to get the relevant data transformed over time.

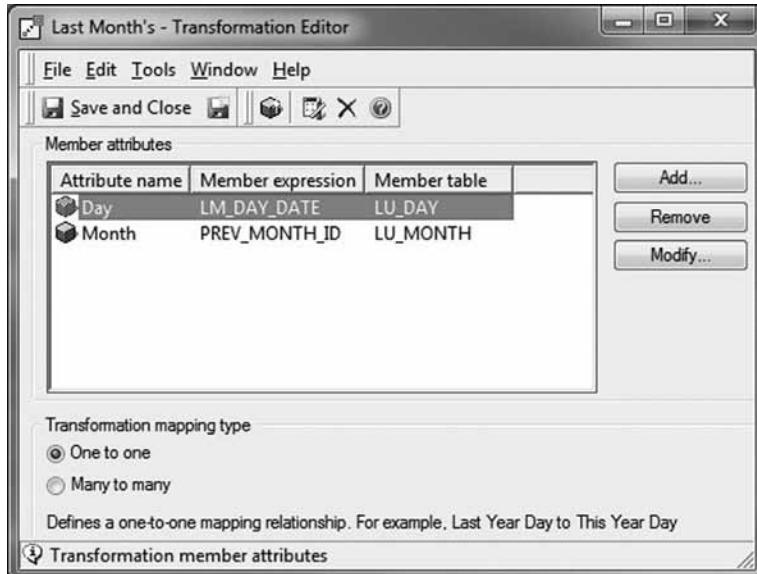


Figure 17-4 Transformations are mainly used for time-series analysis and have 1-to-1 or many-to-many mappings.

- Partitions

When dealing with large data volumes, data is often partitioned in databases to improve performance. For example, instead of storing detailed sales order records for all months in a single table, database administrators often break this table down into a set of tables partitioned by month so that any one table contains only orders for a single month. Most databases allow for partitions to be created and the optimizer automatically routes the query to the correct table. MicroStrategy Architect supports this partitioning

functionality natively. In addition, users can create logical partitions within MicroStrategy that perform the same functions described above without relying on the database to perform the query optimization. MicroStrategy's advanced multi-pass SQL engine creates automatic queries based on partition logic.

- Logical Tables

Logical tables are powerful tools that allow the OLAP architect to create views of information without creating and maintaining these views in the database. Similar to database views, logical tables are created by typing in any valid SQL statement against a database to create the view. Once this view is created, it can be used just like any other table within MicroStrategy. This offers some unique advantages including:

- Defining attributes and facts from expressions spanning multiple tables.
- Flattening non-standard recursive hierarchies.
- Reducing the number of database views.
- Performing complex analysis, such as as-is versus as-was for slowly changing dimensions.

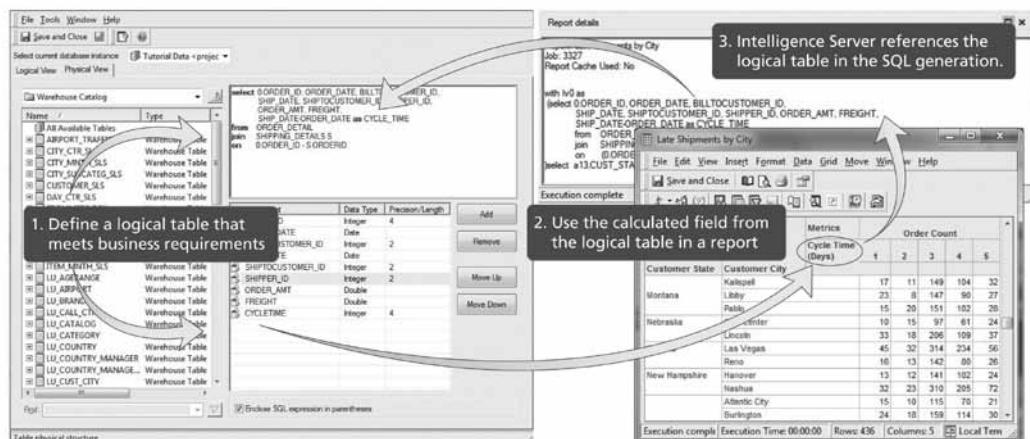
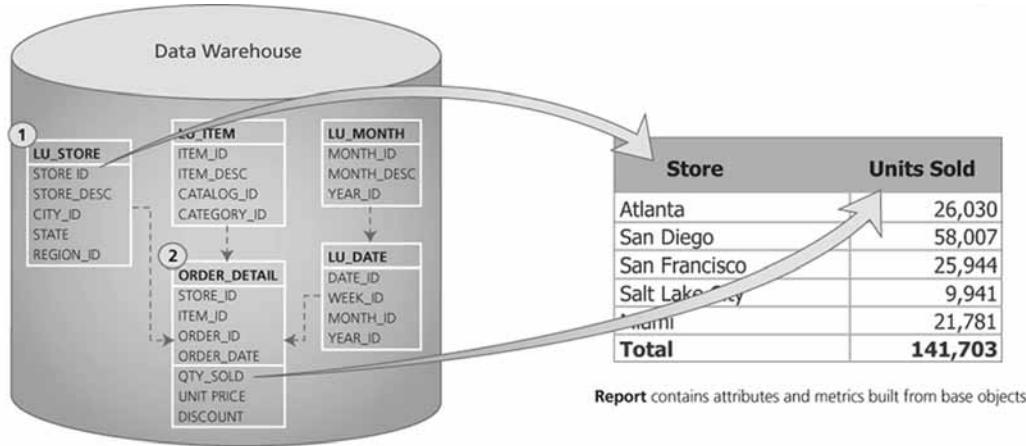


Figure 17-5 A logical table using freeform SQL embeds the SQL statement as a derived table in the dynamically generated report SQL.

17.3 MAP DATABASE STRUCTURES TO QUALITATIVE AND QUANTITATIVE BUSINESS TERMS

MicroStrategy Architect reads the system catalog from the data warehouse, operational datamarts, and databases, as well as cube databases such as SAP BW, and translates the definitions of these data source structures as logical tables and columns. The table and column abstraction objects are the basis for other data abstraction objects such as facts, attributes, hierarchies, and transformations.

This schema abstraction of the physical database tables defines a logical model of the business requirements rather than just the database structures. Therefore, if the underlying physical data model changes, BI architects can quickly and easily incorporate these changes without affecting the higher level objects that use the schema layer. This insulates the BI application from ongoing database modifications.



- 1. Lookup tables** store descriptive information
- 2. Fact tables** store both qualitative and quantitative

Figure 17-6 Architect allows BI architects to map data source columns to business terms using attributes and facts.

User Hierarchies Personalize Business-Focused Data Exploration and Investigative Analysis

BI architects model the relationship between attributes as hierarchies. The system hierarchy represents the true relationship between attributes necessary to generate the SQL when users run reports. The system hierarchy is automatically defined by MicroStrategy Architect when attributes are created and their physical database relationships to other attributes are specified. When users run reports or drill through data, the Intelligence Server reads the system hierarchy in order to generate the appropriate SQL to satisfy the requests.

MicroStrategy Architect provides a unique business abstraction for groups of attributes in a single BI application called user hierarchies. These allow users to browse and drill through the data as required by business needs, as opposed to how the data is physically stored in the data source. User hierarchies can be created through graphical viewers and editors. The visual map displayed in these viewers allows developers to quickly identify and create relationships between data elements. Additionally, business users can graphically visualize the business model.

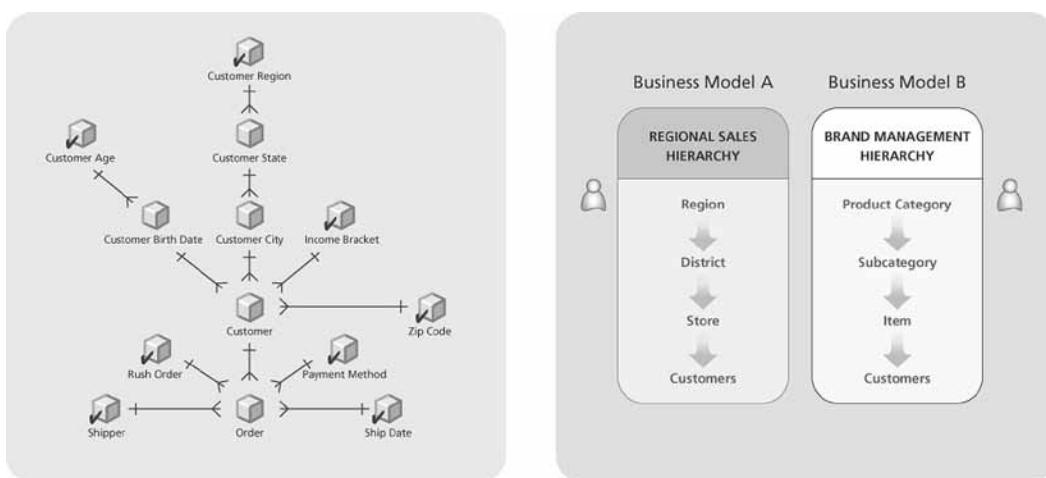


Figure 17-7 The system hierarchy maps the physical data structure. Many user hierarchies define logical business hierarchies that are often different to the way data is stored.

Support Any Data Schema¹²

The MicroStrategy architecture supports a wide range of physical data schemas including common ones such as star and snowflake schemas that have sparse aggregate tables, partitioned tables, and split fact tables. It can also accommodate direct access to operational data stored in third-normal form. Support for a broad range of relational and multi-dimensional data warehouse schema designs ensures that reporting applications easily integrate with an organization's existing data sources.

Leverage Existing Investment in Data Warehouses

Some challenges faced by BI architects while modeling BI applications that leverage existing data warehouses include:

- Inconsistent naming of database fields that represent the same entity
- Entities stored across multiple database fields and tables
- Unavailability of data at levels expected by the business users

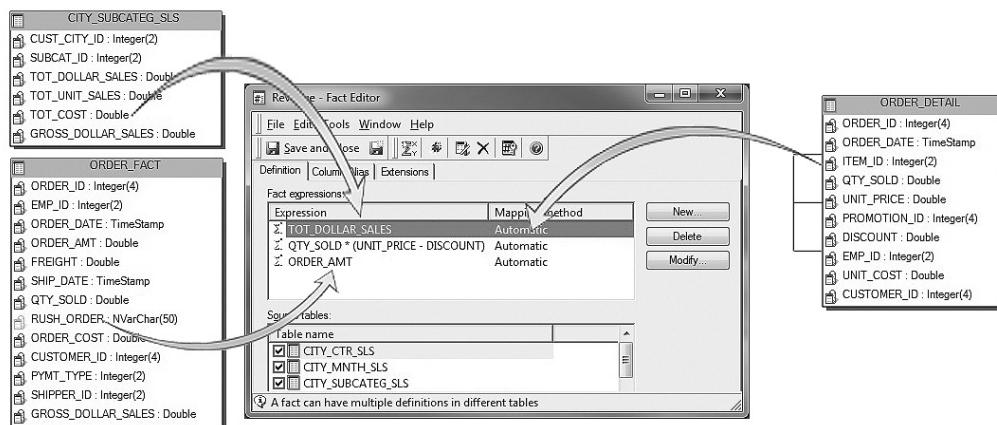


Figure 17-8 The same MicroStrategy object can have different definitions based on the source of the data.

BI architects can easily make available to users different representations of the same attribute by defining attribute forms. Non-textual descriptors, such as images, HTML tags, and URLs, are supported attribute form types that provide great visual appeal and report interactivity.

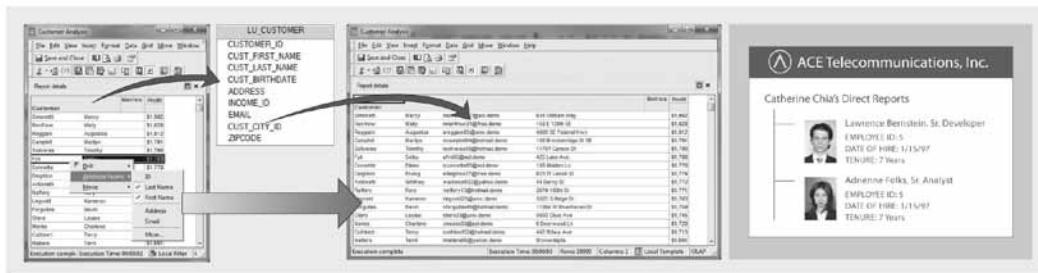


Figure 17-9 Increase visual appeal of reports by defining attribute forms that use dynamic images or URLs.

¹²See Appendix A for more information on Schema Support

Define Multiple Attributes from the Same Database Field Using Attribute Roles

BI applications often need to define many attributes that use the same database table and column. For example, an airline may have a single lookup table that lists all the airport locations served by the airline. To report on timeliness or traffic volumes between airports, the reports need to display airports by the flight origination and destination. MicroStrategy Architect provides unique data modeling options that allow the BI architect to define multiple attributes based on the same lookup table without resorting to database views that may add to the database administration and maintenance burden.

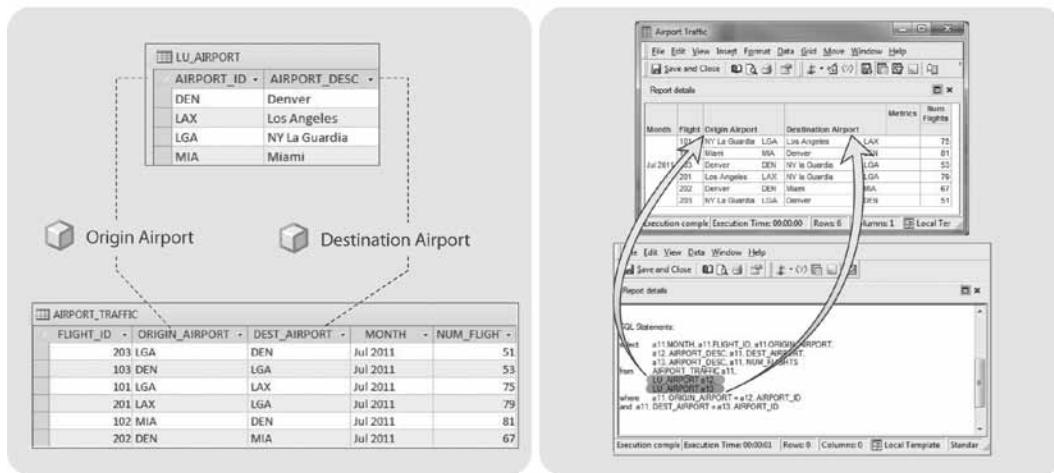


Figure 17-10 Fields in a single lookup table play multiple roles: An airport can either be the origin of a flight, or the destination of the flight as stored in the fact table.

17.4 MICROSTRATEGY DESKTOP

MicroStrategy Desktop is a powerful development interface that includes integrated performance monitoring, enterprise reporting, ad hoc analysis, and decision support workflow. Desktop provides the means to easily access and share critical corporate information from a variety of corporate data sources such as data warehouses, data marts, operational databases, Web services, Hadoop distributions, and cube databases such as SAP BW, Microsoft Analysis Services, Hyperion Essbase, and IBM Cognos TM1 in order to better manage business processes.

With Desktop, report developers get a comprehensive tool to develop the most sophisticated BI applications:

- Build analysis objects using the metadata abstraction layer provided by the data abstraction objects and other analysis objects
- Design and publish reports and documents by combining analysis objects into report objects for powerful analysis

All the objects and business rules defined with Desktop are stored in the central metadata repository, providing a consistent definition and view across all MicroStrategy BI.

Objects Created in MicroStrategy Desktop

BI developers use MicroStrategy Desktop to create two types of metadata objects:

1. Business Abstraction Objects
2. Report Objects

Report objects are built using a combination of business abstraction objects. Grid reports, graph reports, scorecards, and dashboards are types of report objects.



Figure 17-11 MicroStrategy Desktop allows organizations to design business abstraction and report objects used to create sophisticated BI applications. All of these are built on top of the data abstraction of the database schema created using MicroStrategy Architect.

17.5 BUSINESS ABSTRACTION OBJECTS

Business abstraction objects, also called application objects in MicroStrategy, are base objects that are building blocks for reports, scorecards, dashboards, and Mobile apps. These include metrics, filters, templates, and custom groups. Application objects can be shared by multiple reports.

- **Metrics**

Metrics define the analytical calculations to be performed against data in the data warehouse. A metric definition must contain a formula, which determines the data to be used and the calculations to be

performed on the data. It can also contain additional properties that provide great flexibility in defining key performance indicators:

- The level, or dimensionality, determines the level of calculation for the metric. For example, users can choose to calculate profit at the month level or the region level.
- Conditionality associates a filter to the metric calculation. This filter can be controlled to work in conjunction with the report filter.
- A transformation applies offset values, such as “four months ago,” to the selected attributes.

Metrics allow complex business-specific calculations to be presented on reports or documents. Report developers can use one or more functions from an analytic function library, including aggregation, OLAP, statistical, financial, mathematical or ranking functions, as well as logical operators. The object-oriented nature of the metadata allows metrics to reuse other schema and analysis objects, such as attributes, facts, metrics, filters, and transformations, to define new metrics.

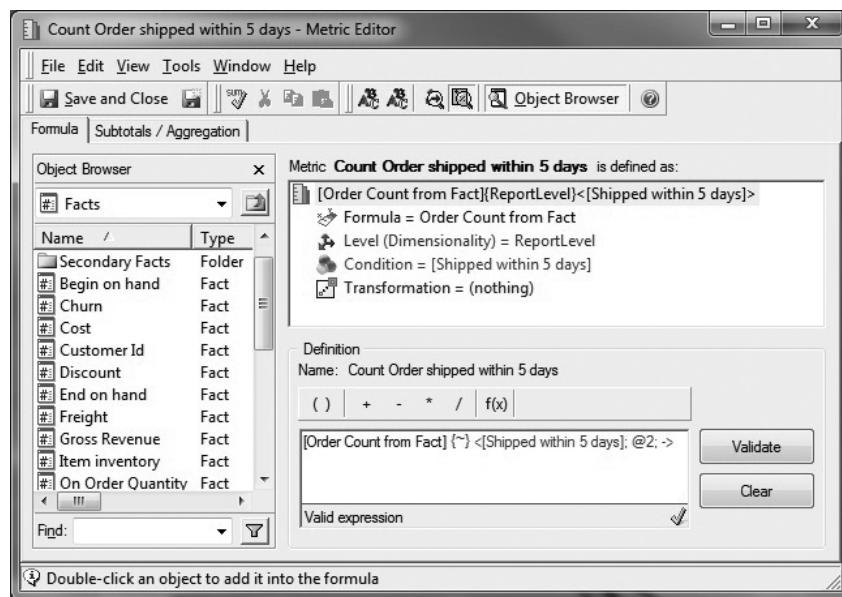


Figure 17-12 The metric editor in Desktop provides great flexibility to create any key performance indicator.

- Filters

A filter specifies the conditions the data must meet to be included in report results. A filter is equivalent to the WHERE clause in a SQL statement. Developers can use the following options to create different types of filters:

- Attribute qualification—allows you to filter on an attribute’s form, such as ID and description, or on the elements of the attribute.
- Set qualification—allows you to create a set based on a metric or on the relationship between two attributes
- Report qualification—uses an existing report as a filter
- Filter qualification—uses an existing filter as a base to add further conditions to a report

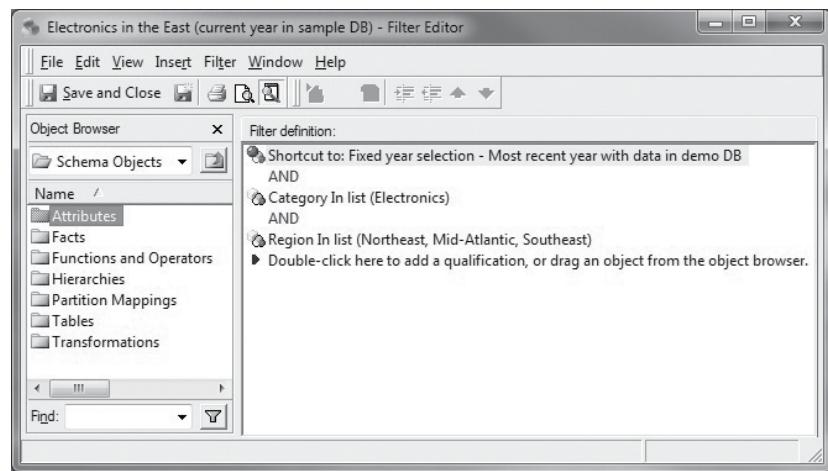


Figure 17-13 Filters control the amount of data that is included in reports.

MicroStrategy Desktop is the only product in the market that allows users to create advanced analyses in a fully integrated, object-oriented environment. Users can identify commonalities from the most detailed, transaction-level data to perform advanced set-based reporting. These sets are completely integrated into the metadata, allowing users to re-use existing analysis objects.

- **Prompts**

Prompts are used to dynamically modify the contents of a report at runtime. With prompts, users can determine, during report execution, the objects they want to retrieve for the report and the report filtering conditions. In addition, users can make different prompt selections each time they run the report. Users can save their prompt answers and use the answers in other reports.

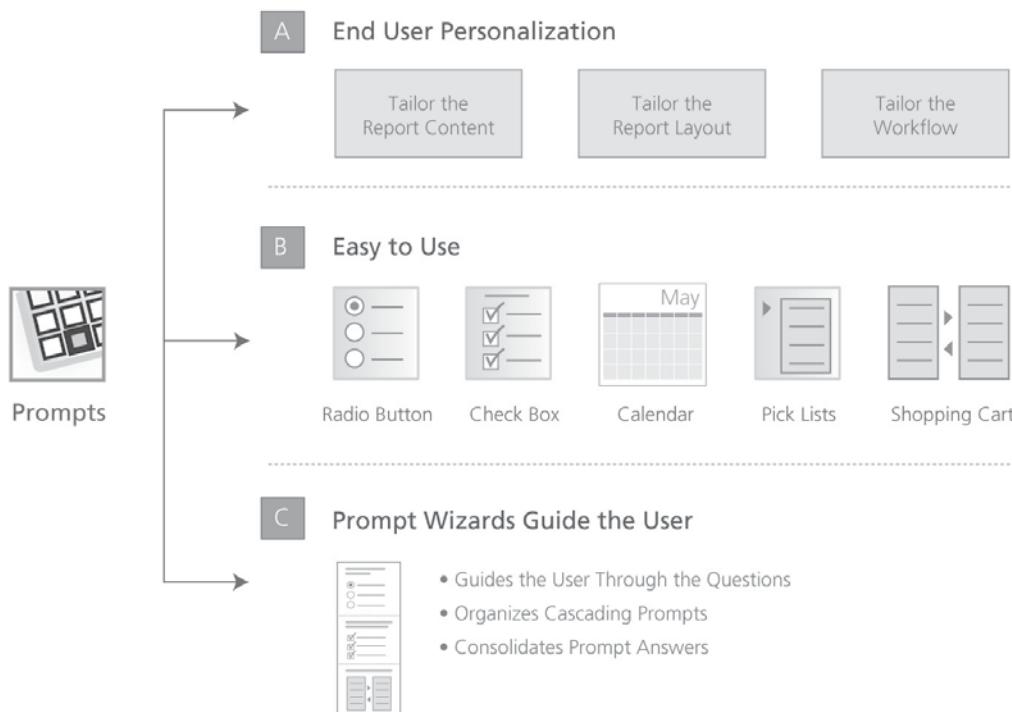


Figure 17-14 Prompts personalize the user experience at report runtime.

The MicroStrategy architecture allows for many types of prompts, including:

- Filter definition prompts — prompt types that allow users to define the filtering criteria, such as attributes in a hierarchy, attribute forms, lists of attribute elements, and metrics
- Object prompts — displays MicroStrategy objects for selection in a report, such as attributes, metrics, and custom groups
- Value prompts — captures a single input value, such as a date, a number, or a text string. The value given is compared to metrics or attributes to filter the data.
- Level prompts — allow users to specify the level of calculation for a metric
- Custom Groups

A custom group is an object that can be placed on a report layout template, and is made up of a collection of elements called custom group elements. Each element contains its own set of filtering or banding qualifications.

	Metrics	\$ Sales YTD	\$ Sales
Sales Customer Group			
Products on Sale this Week		\$ 19,606	\$ 8,954
Brass Cigar Cutter		\$ 6,181	\$ 2,923
Elegant Writing Set		\$ 5,560	\$ 2,256
Fisherman's Reel Clock		\$ 4,630	\$ 1,906
WWI Trench Lighter		\$ 3,235	\$ 1,869
Top 4 Product Groups by Sales Contribution		\$ 43,621	\$ 14,323
Top 50% Product Groups by Sales Contribution		\$ 187,881	\$ 67,345
Bottom 10% Product Groups by Sales Contribution		\$ 5,724	\$ 1,856
Top 3 Brands by Division		\$ 11,920	\$ 4,563
Books - Bowles		\$ 2,265	\$ 1,006
Books - Brater		\$ 1,775	\$ 980
Books - Young		\$ 1,075	\$ 676

Figure 17-15 Custom Groups define the content of rows in a MicroStrategy report.

For example, using the Custom Group Editor, a custom group for Store Inventory can be created as follows:

- Small stores with low inventory: Store Sales < 50 AND Store Inventory < 200
- Large stores with low inventory: Store Sales > 50 AND Store Inventory < 200

Custom groups provide drill down capabilities to their constituent attribute elements and can display both the group and constituent levels on a report.

- Consolidations

Consolidations group specific attributes together. They allow developers to place a group of attribute elements on a template as if it were an attribute. The elements of the consolidation appear as rows on the report and can include arithmetic calculations.

For example, suppose users want to see each season as a separate row on a report, but the seasons do not exist as attributes in the project. A consolidation allows you to group together the elements of the Month of Year attribute into various seasons and place them on the template. This consolidation will contain four consolidation elements, one for each season. Summer consists of June + July + August; Fall consists of September + October + November, and so on. The consolidation is placed in the rows of the report with the desired metrics in the columns. Therefore, when the user runs the report, the metric values for June, July, and August are added together to yield the value for Summer. This occurs for each season.

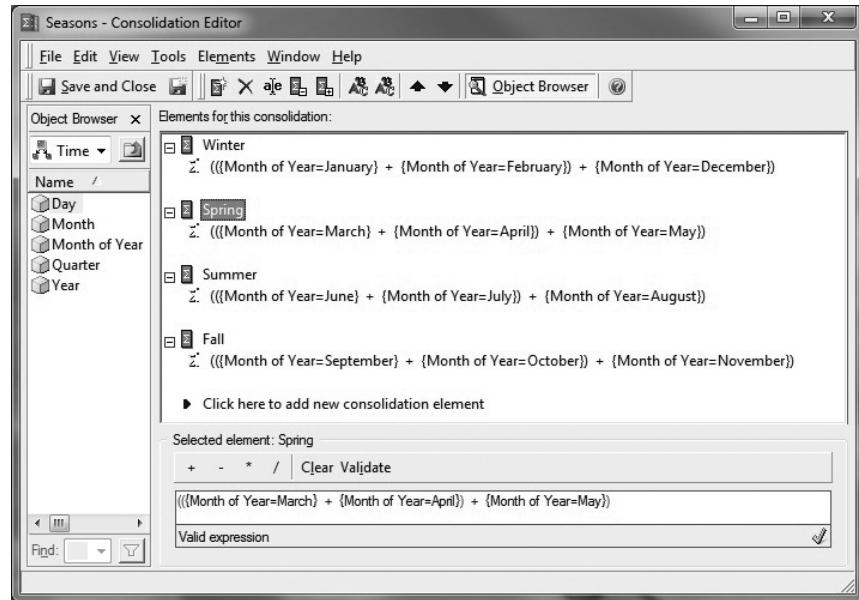


Figure 17-16 Consolidations enable arithmetic between individual rows or groups of rows in a report.

17.6 REPORT OBJECTS

The business abstraction objects described above are usually developed by a core group of developers who create re-usable objects with standard definitions that can be made available to all applications and users. For example, organizations need to maintain a standard definition of the metric “revenue,” which should be common across all users and reports. Report development in the MicroStrategy platform is usually carried out by a larger population comprised of IT developers and report users. There are two types of Report Objects in the MicroStrategy BI platform:

1. Grid and Graph Reports – These are single data set reports that are displayed to users either in a cross-tabulated grid format or a graph formats.
2. Report Services Documents – These are used to create Mobile apps, scorecards, dashboards, invoices, statements, and other operational and business reports which require custom formatting and print perfect layouts. These may be comprised of one or more data sets from different sources.

A report is a MicroStrategy object that represents a request for a specific set of formatted data from the data warehouse, operational database, or cube database. Reports are the focus and goal of business intelligence. They allow users to gather business insight through data analysis. Reports in MicroStrategy can come from the following data sources:

- Data warehouses using MicroStrategy’s dynamic SQL engine
- Operational databases, datamarts, and Web services using freeform SQL or XQuery
- Cube databases, such as SAP BW, Microsoft Analysis Services, Hyperion Essbase, and IBM Cognos TM1 using MicroStrategy’s dynamic MDX engine

17.7 REPORT CREATION USING THE DYNAMIC SQL ENGINE

Most MicroStrategy reports are built using fundamental building blocks such as data abstraction objects and business abstraction objects. When these are assembled onto a report, the MicroStrategy’s dynamic SQL engine generates optimized SQL at report runtime.

Reusable templates specify the layout and formatting of the data on a report. Attributes, hierarchies, metrics, custom groups, consolidations, and prompts can be placed in rows or columns on the template grid, or in pages to break reports into manageable pieces. Report developers can specify data presentation characteristics such as font, color, alignment, and number formats, as well as additional report characteristics such as subtotaling, metric thresholds, and graph properties. Filters specify the conditions that the data must meet to be included in a report or in a metric.

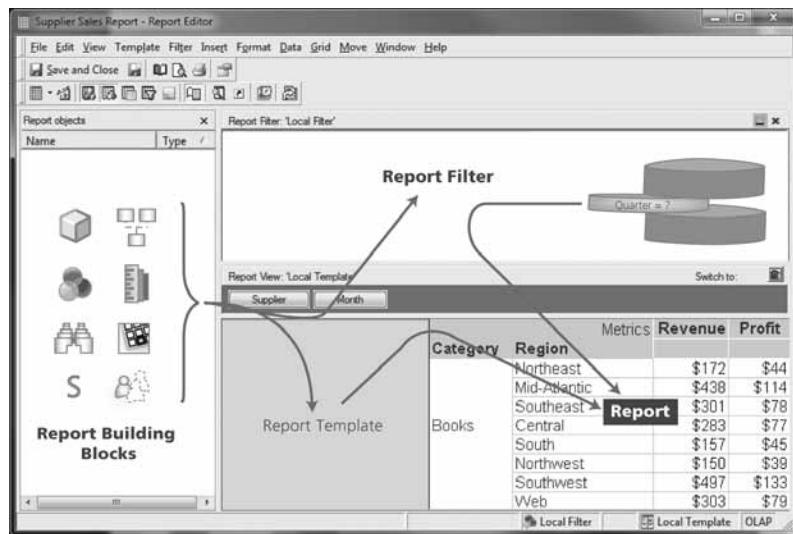


Figure 17-17 Report design involves defining the filter (data) and template (layout) from the report abstraction objects in the metadata.

Reports and documents present decision-makers with information gleaned from the organization's data stores. Report developers can assemble all analysis objects necessary to turn data into insight using the most intuitive presentation format – whether it be a classical cross-tabulated grid, a graph, or a highly visual dashboard or scorecard.

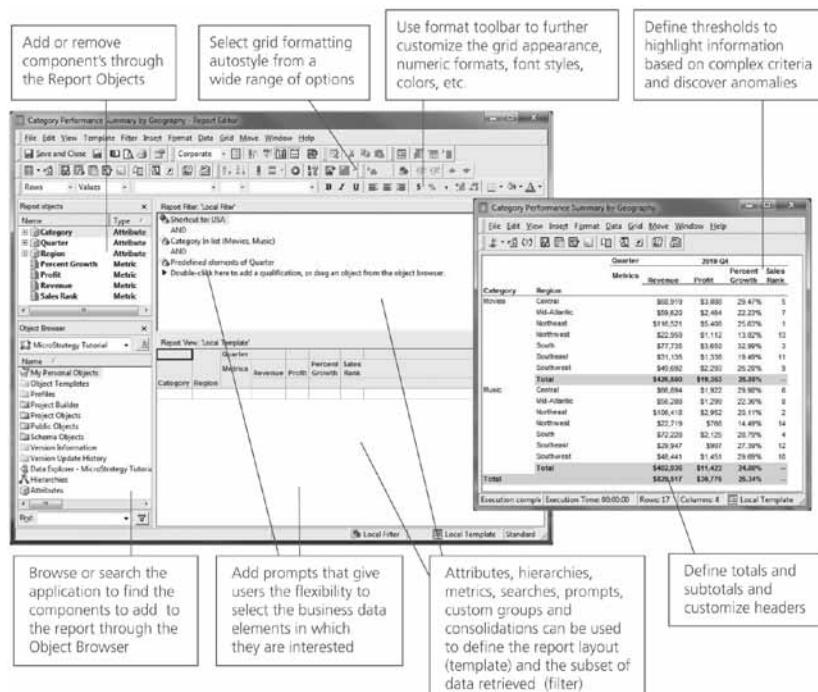


Figure 17-18 Report developers have a wide range of features to create any report or analysis.

Tuning Report SQL Using VLDB Properties

VLDB properties allow developers to customize the SQL that MicroStrategy generates. Developers can configure settings such as SQL join types, SQL inserts, table creation properties, and so on. Developers can use the VLDB Properties Editor to alter the syntax of a SQL statement and take advantage of unique, database-specific optimizations. Some of the qualities that make these properties valuable are:

- Optimization: Take advantage of database-specific settings to further enhance the performance of queries.
- Flexibility: VLDB properties are available at multiple levels so that the SQL generated for one report, for example, can be manipulated separately from the SQL generated from another, similar report.
- Complete database support: VLDB properties allow you to easily incorporate and take advantage of new database platforms and versions.

VLDB properties can be set at various levels in the MicroStrategy architecture. The following diagram shows how VLDB properties that are set for one object take precedence over those set for another.

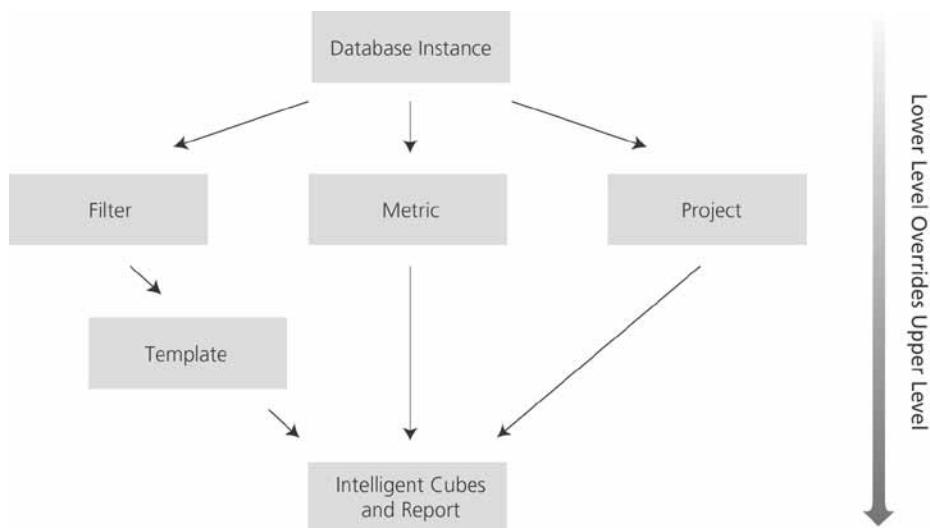


Figure 17-19 VLDB Order Priority with the lower levels overriding the higher levels.

17.8 CREATING REPORTS FROM INTELLIGENT CUBES

MicroStrategy OLAP Services allows for the creation of Intelligent Cubes. These cubes are in-memory multi-dimensional collections of attributes and metrics which constitute a sandbox of information within which users can perform OLAP operations to get speed of thought analysis. However, Intelligent Cubes are not bound by the limitations of traditional cubes. Users can drill outside of the cube at any point to retrieve more information from underlying data sources.

Reports using Intelligent Cubes are created in MicroStrategy Desktop in a very similar manner to regular reports. Developers add metrics, attributes, hierarchies, custom groups, and other objects to the layout template view and add any filter criteria.

In addition, for OLAP Services-enabled reports, developers have the option of adding objects to the “Report Objects” pane. Attributes and metrics placed in this window become part of the in-memory Intelligent Cube. The user can drag and drop or drill to these available objects at any time and retrieve the information from the OLAP Services cube.

Cube Design Recommendations with Cube Advisor

MicroStrategy Cube Advisor assists developers and administrators in the choice of which Intelligent Cubes to build in order to best support their end users. Cube Advisor makes recommendations based on how existing reports are defined, how frequently the reports are executed, and how much database execution time was required.

Dynamic Sourcing of Intelligent Cubes

In order to ensure end users get fast response times to their report queries, MicroStrategy enables dynamic sourcing of Intelligent Cubes. With dynamic sourcing, MicroStrategy OLAP Services will check the contents of a report data request and will compare it to all existing Intelligent Cubes. If one of the existing Intelligent Cubes can act as the data source, the report will automatically source data from that Intelligence Cube rather than retrieving the information from the data source.

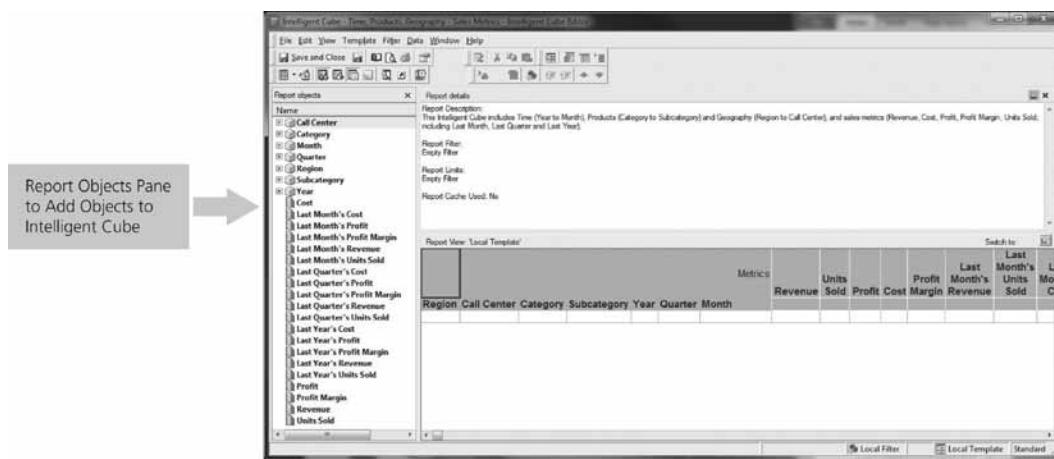


Figure 17-20 Report Designer for reports using Intelligent Cubes.

17.9 BUILD AND EXECUTE REPORTS WITH THE DYNAMIC MDX ENGINE

The MicroStrategy BI platform extends the scope of analyses provided by BI applications by incorporating data assets other than the traditional data warehouse. Organizations often have information stored in multiple locations, such as SAP BW, Microsoft Analysis Services, Hyperion Essbase, and IBM Cognos TM1 cubes. These cube databases can be queried using the MDX query language.

The Intelligence Server has a dynamic MDX engine that allows organizations to query the information stored in these databases, and also combine this information with other data sources using Report Services. The dynamic MDX engine creates MDX based on user query requests at runtime, similar to the SQL engine.

For example, the multi-dimensional models of SAP BW Infocubes and Querycubes are imported and represented in the data abstraction objects. This import process allows MicroStrategy to read all the SAP BW business content such as queries, Infocubes, variables, structures, hierarchies, key figures, and characteristics. These SAP metadata elements are translated into corresponding MicroStrategy metadata objects. SAP BW cube objects are mapped to business abstraction objects and used to create reports and documents.

During Report execution, the MicroStrategy metadata serves as a pointer to the underlying SAP metadata. MicroStrategy dynamically generates optimized MDX and executes the MDX against SAP BW to retrieve the data.

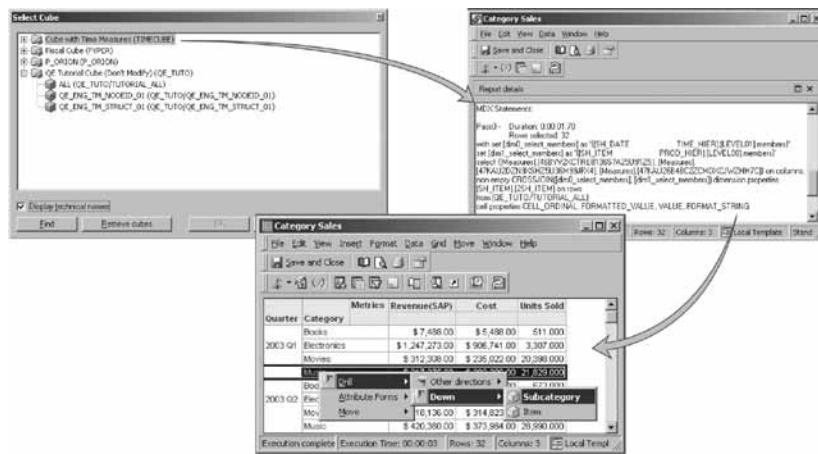


Figure 17-21 Reports can be built against cube databases such as SAP BW cubes.

17.10 OPERATIONAL REPORTS USING THE FREEFORM QUERY ENGINE

The Freeform Query functionality adds great flexibility to MicroStrategy's query and reporting capabilities. Traditionally, developers use the MicroStrategy Engine to generate SQL or MDX to run against a relational or multidimensional database to get a desired report. In addition to generating reports in the traditional way, developers can also use their own customized SQL or XQuery statements to generate reports from operational systems and Web services available in the organization. This capability saves organizations a tremendous amount of time since developers do not need to place the data into a data mart or data warehouse first. In addition, these reports may be used in cases where quick proof of concept reports need to be built or in cases where the table structures in the data sources are not ideally suited for dynamic SQL generation.

The MicroStrategy freeform query editor is flexible to allow developers to insert prompts and security filters into the SQL and XQuery statement so that the returned results are personalized for each user.

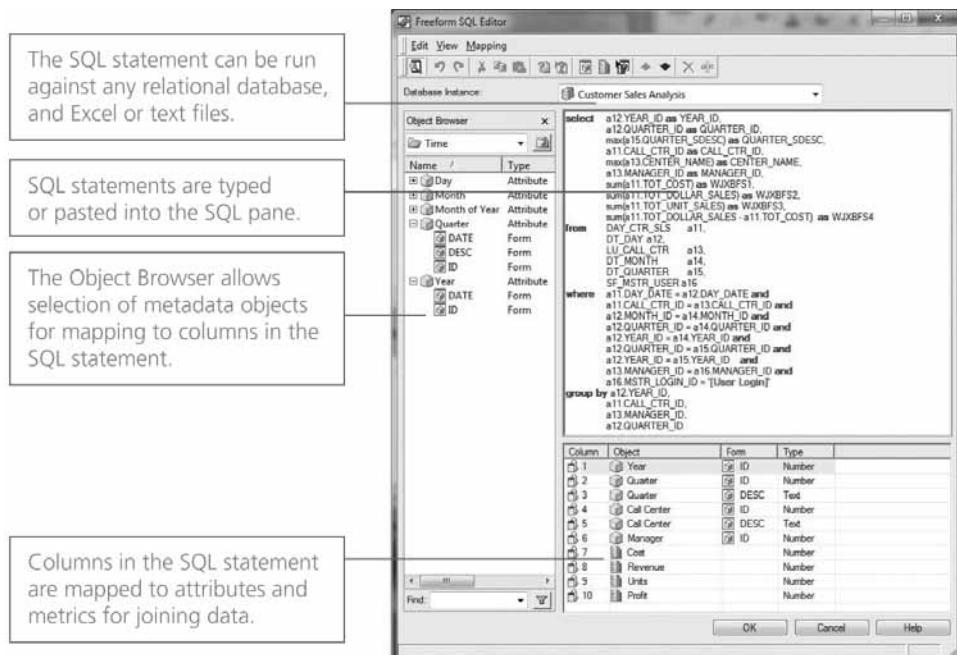


Figure 17-22 Freeform SQL editor in MicroStrategy Desktop.

17.11 TRANSACTION REPORTS

MicroStrategy offers developers the ability to create documents that request and capture input from end users. This transaction document is linked to a transaction report that writes the user input to a database or Web service. An example of a transaction report might be an Inventory Analysis report where the end user can make changes to product inventory values directly on a document.

Transaction reports embed a placeholder in the report SQL or XQuery to write input from an end user to a database or Web service. The placeholders are mapped to other MicroStrategy objects, such as attribute forms or metrics in the document, and to columns in the SQL or XQuery statement. When an end user inputs information in the document, the new information is substituted for the placeholders in the transaction report, which writes the information to the data source.

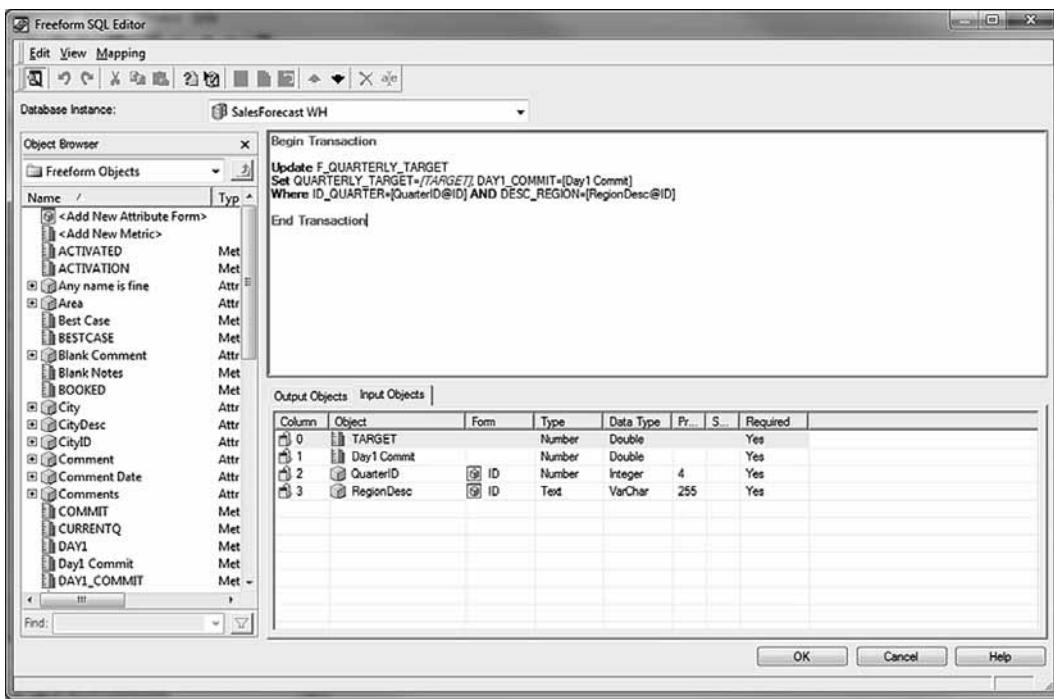


Figure 17-23 Transaction reports write back data to the data source using SQL or XQuery.

17.12 BUILDING GRAPHS IN MICROSTRATEGY DESKTOP

Once a report has been laid out using a template and filter combination, the report developer can also choose to expose the report as a graph. By dragging and dropping attributes and metrics into areas called drop zones, report designers can quickly create attractive graphs. MicroStrategy Desktop provides a comprehensive set of graph formatting options to allow users to graphically visualize reports in the best possible manner.

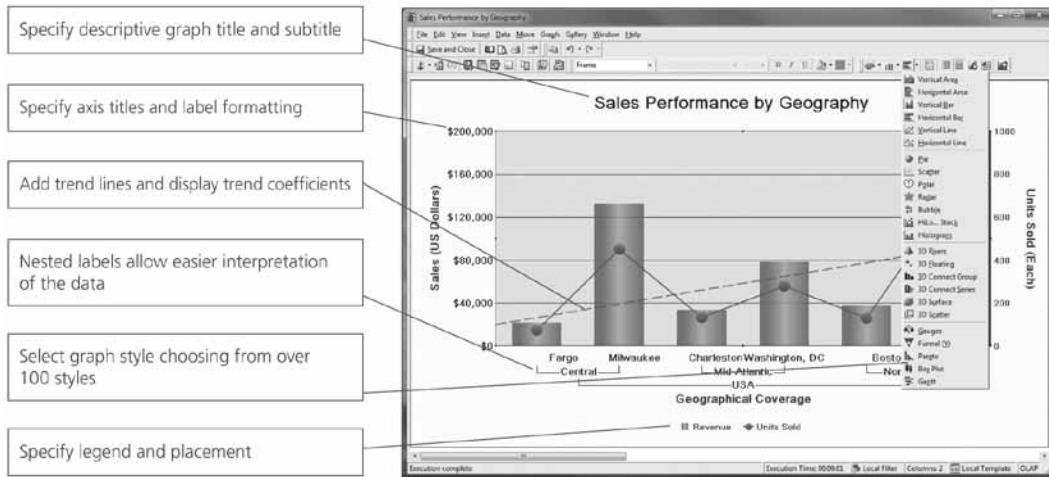


Figure 17-24 Graphs provide the means to show visually appealing and effective graphical representations of the data.

17.13 DATA MINING AND PREDICTIVE REPORTS IN MICROSTRATEGY DESKTOP

Using MicroStrategy Desktop, report developers can create data mining and predictive metrics, train the data mining models using live data, and deploy them to create real-time forecasts of future events.

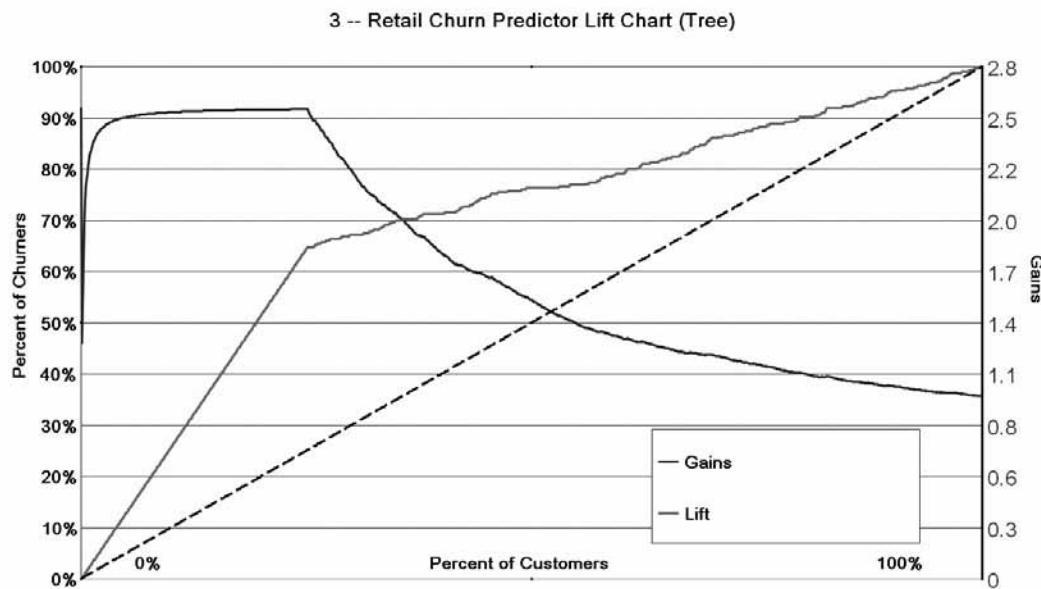


Figure 17-25 This Lift Chart shows the improved accuracy of a customer churn predictor over a random sample of customers.

Organizations can leverage their investments in third-party data mining tools by importing predictive models created in those tools in Predictive Modeling Markup Language (PMML) and use the predictive metrics in any report. Scripts created using R can be embedded in MicroStrategy's function library, providing new analytical functions and graphics capabilities.

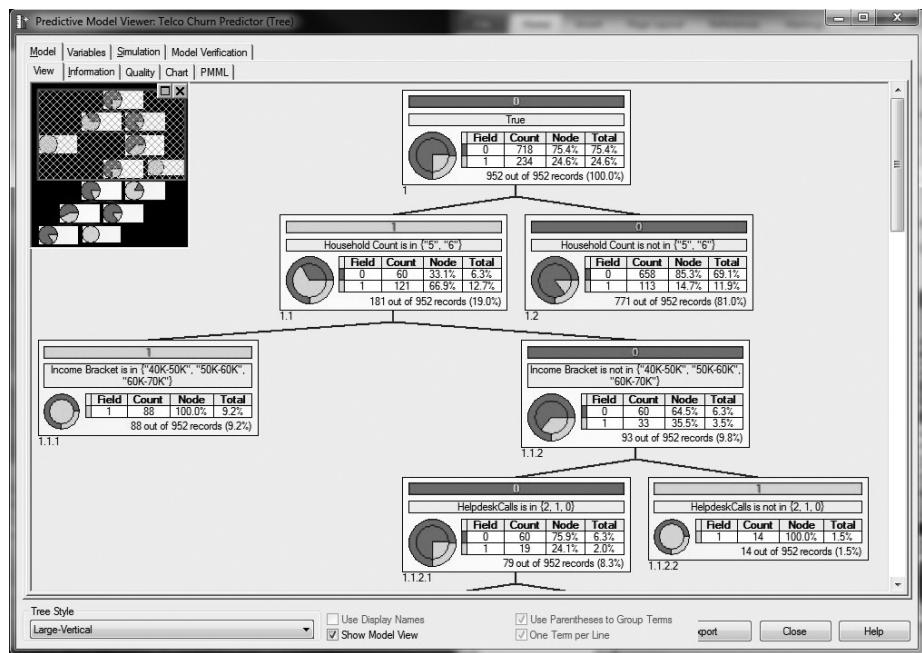


Figure 17-26 This model view of a decision tree algorithm can be used effectively for customer segmentation analysis.

17.14 REPORT SERVICES DOCUMENT CREATION

MicroStrategy Report Services raises the bar for enterprise reporting by combining the best OLAP analysis and predictive analysis capabilities with pixel-perfect Mobile apps, enterprise reports, scorecards, and dashboards on a unique industrial-strength business intelligence platform for Mobile and Web applications.

Report Services Documents in MicroStrategy are used to create print-perfect and pixel perfect document layouts. Types of documents that can be created with MicroStrategy Report Services are:

- Mobile apps
- Scorecards
- Dashboards
- Business performance management reports
- Invoices and Statements
- Operational reports
- Compliance and Auditing reports
- Data Input applications

Creating Report Services Documents in MicroStrategy Desktop

Report Services Documents are created in MicroStrategy Desktop using the Document Editor. The Document Editor allows developers to create, customize, and save documents to be used across the MicroStrategy platform. There are four main sections in the Editor:

1. Dataset Objects

This section displays all of the datasets used in the document. Datasets are any MicroStrategy Report and can come from any source such as data warehouses, operational databases, Web services, or

multidimensional databases. Users can combine data from multiple sources into a single document by adding multiple datasets. It lists all attributes, metrics, custom groups, and consolidations in the existing MicroStrategy report, regardless of whether or not they are displayed on the report.

2. Layout area

The Layout area provides the framework for precisely controlling the display of the fields when the document is viewed as a Mobile, PDF, Excel, HTML, or Flash view. To add data, developers drag objects from Dataset Objects and drop them into this area. Depending on the section they place objects in, they print or display differently.

- Page Header/Footer: prints at the top and bottom of each page
- Document Header/Footer: prints at the beginning/end of the document
- Grouping Header/Footer: for each field in the Grouping panel, prints before and after the Detail Header/Footer
- Detail Header/Footer: prints immediately before and after each group of Detail sections
- Detail: repeats for each row in the dataset

3. Grouping panel

The Grouping panel lets developers group information in the document in a hierarchical structure. To add a group, developers drag and drop any attribute, consolidation, or custom group from the Dataset Objects onto the Grouping panel. The sequence automatically determines the hierarchy.

4. Property List

The Property List displays the properties of the object selected on the Layout area. The properties that are listed vary depending on the type of object (text field, image, line, section, and others) selected.

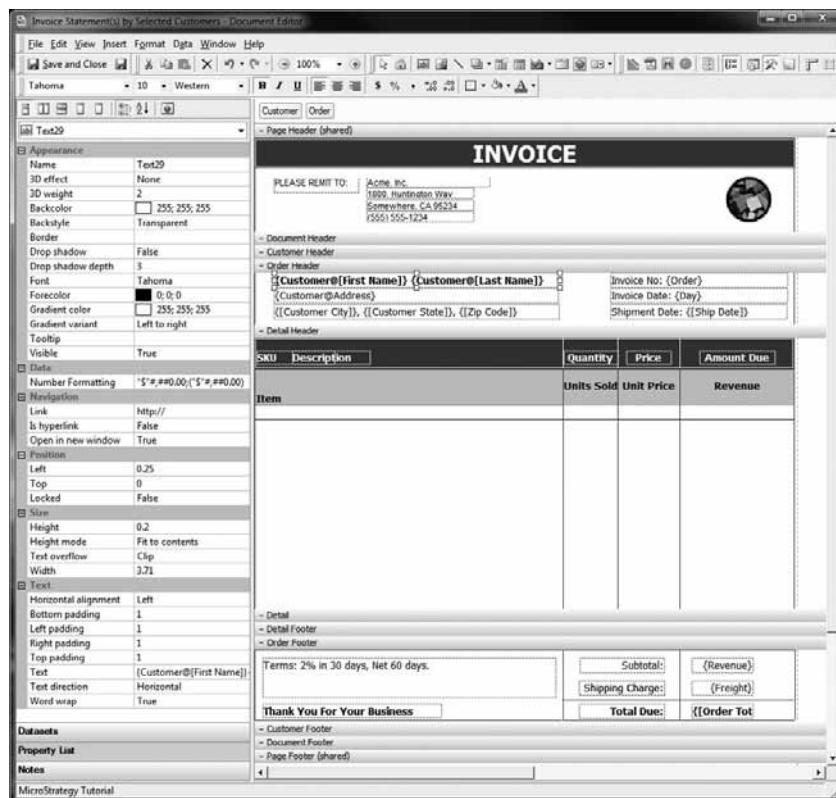


Figure 17-27 MicroStrategy Desktop Document Editor Layout.

Dashboard Creation Using Zone-Based Layouts

Zone-based Layout is optimized for the creation of Mobile apps, scorecards and dashboards. The term “zone” refers to the graphical building block technique that allows users to layout entire tables, graphs, visualizations, and selectors onto a page or screen, where they have a presentation and interactive behavior of their own. Zone-based layout is optimized for displaying graphical content rather than tabular data. Zone-based layout is further optimized for on-screen display, where users have scroll bars to move around the report and selectors that provide interactivity requiring little to no end user training.

MicroStrategy users can build scorecards and dashboards on the fly using simple drag-and-drop techniques to insert multiple reports, graphs, text, hyperlinks, and images onto the report layout screen. These objects may also be arranged anywhere in the layout and will automatically adapt to the size and shape of their content, moving other objects dynamically to fit.

Predefined templates can be used to begin the design process. Using MicroStrategy’s powerful formatting capabilities, developers can add many layers and depth to their dashboards. Report developers can create dashboards that contain multiple different layouts, or pages, with each page containing different reports, graphs, or data visualizations. Selector controls can be added to dashboards to make it possible for end users to interact with and manipulate what data is displayed. Report developers can even harness information on other websites by incorporating HTML containers within a dashboard allowing them to design any number of unique dashboards.

Enterprise Report Creation Using Banded Layouts

MicroStrategy Report Services Documents are ideal for designing enterprise reports by making it possible for report developers to use Banded Layouts. Banded Layout is the predominant vehicle for operational reporting. It has been the cornerstone of all leading report writers for years. The reason for its long-term popularity for enterprise reporting is that it allows the organization of large amounts of data into a natural hierarchy of information that users can easily navigate. The term “banded” refers to specific horizontal bands superimposed on the report layout screen that dictate where data should be automatically summarized for page headers and footers, report headers and footers, and within user-defined groupings or hierarchies that map to the business organization.

Banded report layout is truly optimized for traditional operational reporting, with its dense hierarchical information presentation, multi-page repeating sections that cover entire ranges of enterprise operations, and need for fine printing optimization. MicroStrategy offers pure banded layout following the same visual paradigms that are now well understood by any person who has created enterprise reports using any of the common report writing products available today. MicroStrategy offers easy drag and drop options for designing operational reports and flexible formatting options to meet all report specifications. In addition to flexible formatting, MicroStrategy also makes it easy for developers to add important additions to their reports, such as a watermark and table of contents.

17.15 ADVANCED VISUALIZATIONS

In certain instances, a chart or graph is not sufficient for analyzing data in a report. In these cases, advanced visualizations can be used to analyze the data. Advanced visualizations are intuitive, interactive displays of data that allow data consumers to quickly understand large quantities of information quickly and in a visual manner.

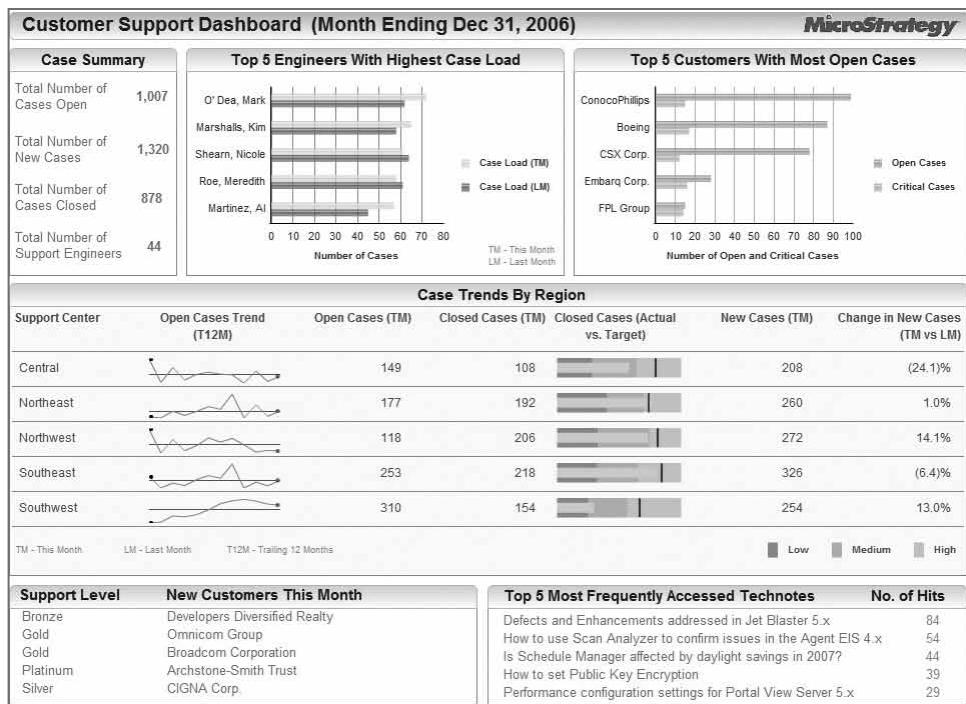


Figure 17-28 Report developers can include intuitive data displays called visualizations in reports.

MicroStrategy provides a wide selection of pre-made advanced visualizations, also called “widgets”. Visualization widgets are rendered in Flash and include:

- Bubble Grid
- Cylinder
- Data Cloud
- Date Selection
- Fisheye Selector
- Funnel
- Gauge
- Google Map
- Graph Matrix
- Heat Map
- Interactive Bubble Graph
- Interactive Stacked Graph
- Media Widget
- Microcharts
- RSS Reader
- Thermometer
- Time-Series Slider
- Waterfall chart
- Weighted List Viewer

Different visualizations have different degrees of interactivity. Some visualizations, such as the Graph Matrix, include several different data display options that the report consumer can adjust to change the way the data is displayed. These options include both formatting and data display manipulations, making it possible for the report consumer to get a whole new perspective of the data by making a few adjustments. Other visualizations, such as the Gauge, by its very nature do not require interactivity and are instead meant to help with rapid data consumption.

17.16 CREATING DASHBOARDS AND REPORTS IN WEB

As business intelligence applications grow in scope and usage, two important themes regarding the user population have become apparent:

1. Traditional lines between roles such as end user, power user, and developer are rapidly becoming blurred with end users requiring more functionality and ad hoc capabilities to create their own reports.
2. Organizations have distributed development teams spread out across different business units and geographies. Report development can be farmed out to a broader user population.

Installing a thick client desktop application on end user machines is too expensive to deploy and maintain for larger user populations. A true thin client Web interface that allows for the editing of reports and the creation of new reports from scratch is required. MicroStrategy Web Professional offers organizations and developers just that capability. The main features of MicroStrategy Web Professional from a developer perspective are:

- Create new Grid and Graph reports over the Web using a thin client browser interface.
- Create pixel-perfect formatted dashboards, scorecards, and operational reports over the Web.
- Edit existing reports and dashboards over the Web in a WYSISYG manner.

Create Grids and Graphs Over the Web

MicroStrategy Web Professional offers Web-based developers and power users the ability to create reports in the form of grids and graphs.

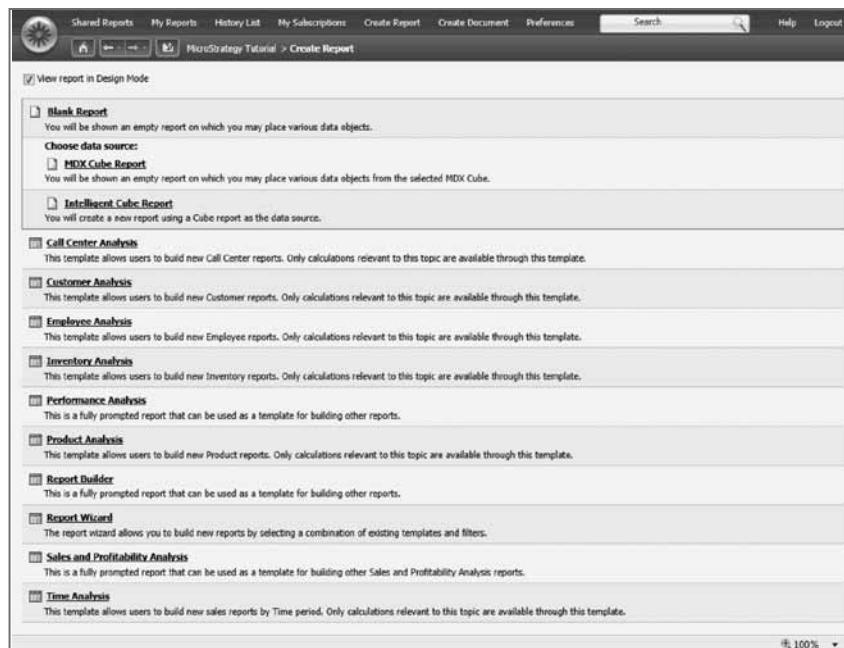


Figure 17-29 The report creation options in the MicroStrategy Web interface.

There are a number of options with which a user can create reports in MicroStrategy Web.

- Ad hoc report creation from a blank template.

This option allows users to pick and choose from any available key figures and attributes within the MicroStrategy metadata. Users drag and drop these objects on the design area to create the report. Complete ad hoc report creation is possible with the developers choosing from any available objects in the MicroStrategy metadata and placing them in the report template and filter.

- Object Browser Panel – Choose from all the attributes and metrics available in the MicroStrategy metadata to place on the report template.
- Report Filter Window – Developers and power users add conditions to restrict the data brought back to the user. Existing filters can be reused or new filters can be defined from the objects in the Browser Panel.
- Page-by Axis for drop down attributes – Users can also place attributes and custom groups into page by axis. When attributes are placed in this position, they generate a drop down list of elements that users can scroll through.
- Template definition pane built using drag and drop – Users drag and drop attributes and metrics from the Object browser pane on to the template area. This allows users to choose order, placement, and location of attributes and metrics on the report layout.

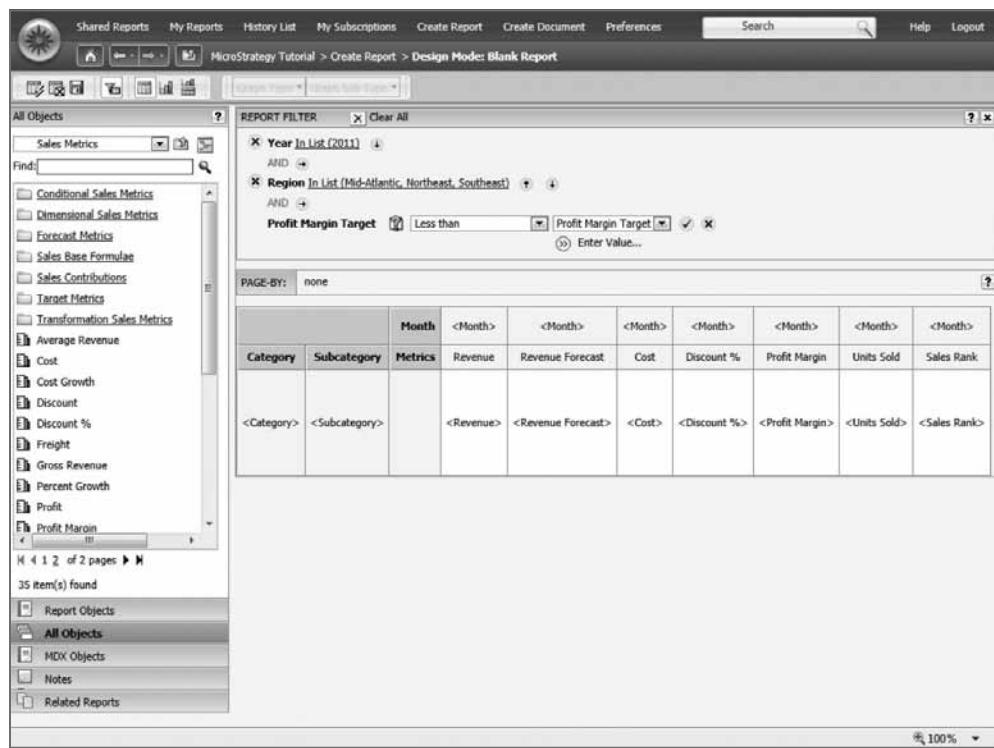


Figure 17-30 Ad hoc report creation in MicroStrategy Web using a blank template.

- Ad hoc report creation from an Intelligent Cube

A predefined template contains the set of associated metrics and attributes stored in an Intelligent Cube. This option narrows the range of objects presented to the user. For example, users can then choose from a subset of objects in a closely related area such as customer analysis or financial analysis. Once users choose a particular template, they can drag and drop the attributes and metrics contained in that template, and create reports in a manner similar to the ad hoc report creation process.

The screenshot shows the MicroStrategy Web interface in Design Mode. The top menu bar includes Shared Reports, My Reports, History List, My Subscriptions, Create Report, Create Document, Preferences, Search, Help, and Logout. The main content area is titled "MicroStrategy Tutorial > Create Report > Design Mode: Product Analysis". On the left, a "Report Objects" browser lists various dimensions and measures like Brand, Category, Quarter, Region, Subcategory, Supplier, Year, Cost, Profit, Revenue, and Units Sold. A message at the top of the main area says "REPORT FILTER The filter is empty. Use the object browser to add objects." Below this is a "PAGE-BY" section with "Year" selected. The main workspace displays a table structure with columns: Region, Category, Quarter, Units Sold, Revenue, and Profit. The "Region" column has a dropdown menu showing "Category" and "<Region>". The "Quarter" column has a dropdown menu showing "<Quarter>". The "Units Sold" column has a dropdown menu showing "<Units Sold>". The "Revenue" column has a dropdown menu showing "<Revenue>". The "Profit" column has a dropdown menu showing "<Profit>". At the bottom left, it says "11 item(s) found" and lists Report Objects, All Objects, MDX Objects, Notes, and Related Reports. A zoom control at the bottom right shows "100%".

Figure 17-31 Ad hoc report creation in MicroStrategy Web from a predefined template.

- Using the Report Builder

This option presents a set of prompts that enable the selection of attributes, metrics, hierarchies, and conditions needed for a particular report in an ad hoc fashion. Report Builder is useful for users who require a tightly controlled workflow for report creation.

The screenshot shows the Report Builder interface. The top menu bar includes Shared Reports, My Reports, History List, My Subscriptions, Create Report, Create Document, Preferences, Search, Help, and Logout. The main area is titled "MicroStrategy Tutorial > Create Report > Design Mode: Report Builder". It contains two main sections: "1. Choose the attributes of the report (Required)" and "2. Choose the metrics of the report". In the first section, under "Available", items like Customer State, Days to Ship, First Order Date, Income Bracket, Last Order Date, Order, Payment Method, and Phone Plan are listed. Under "Selected", items like Age Range, Customer Gender, Education Level, Household Count, Housing Type, and Marital Status are listed. In the second section, under "Available", items like Avg Revenue per Customer, Max Revenue per Customer, Min Revenue per Customer, and Remainder Revenue are listed. Under "Selected", items like Targeted Customers Revenue, Std Dev Revenue per Customer, and Transactions Per Customer are listed. At the bottom, there is a "Report Message Name:" field containing "Report Builder", and buttons for "Edit in Design Mode", "Save", and "Cancel". A zoom control at the bottom right shows "100%".

Figure 17-32 The Report Builder prompts for report objects and filtering criteria.

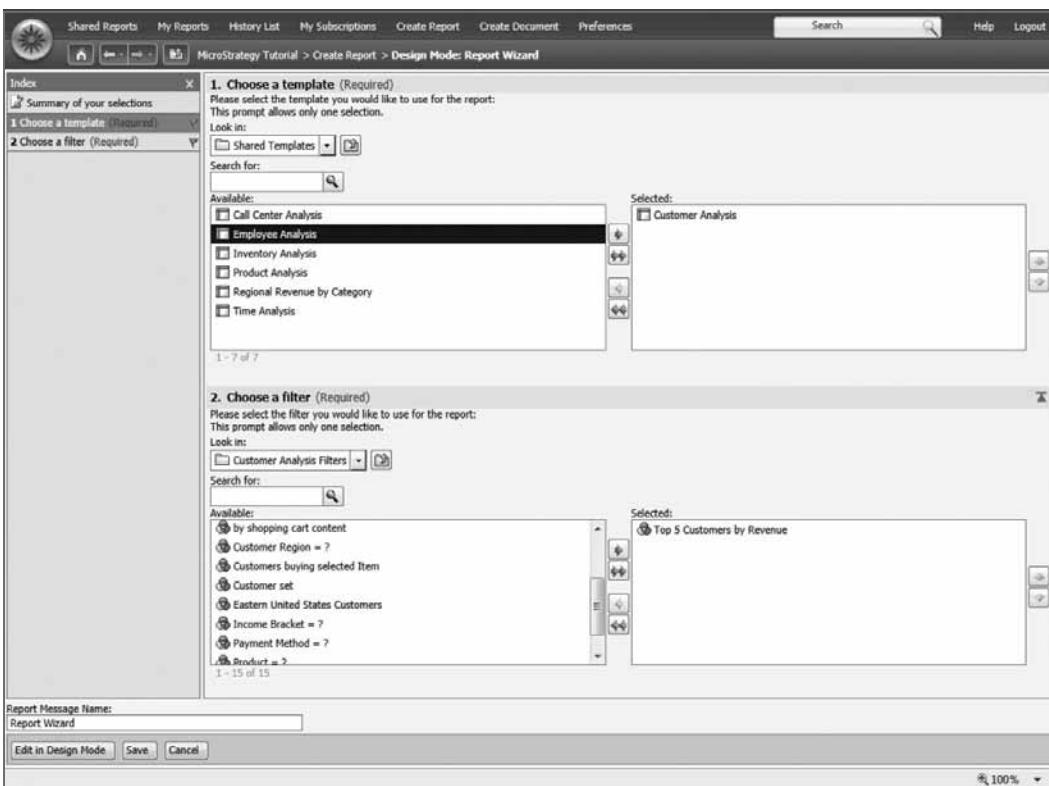


Figure 17-33 The Report Wizard prompts for templates and filters stored in the metadata.

Fine-Grained Control of Grid Formatting in MicroStrategy Web

Once the report has been created, MicroStrategy Web offers powerful capabilities for formatting the report in either grid or graph formats for print or presentation purposes. With formatting toolbars and functionality intuitively familiar to users of spreadsheet applications, the properties of various cells in a report can be changed to achieve the desired grid formatting. Colors, fonts, styles, numbers, alignment, borders, and sizes can be altered by users over the Web.

These properties can be accessed either from a readily available formatting toolbar or by right clicking on any attribute or metric to get to the formatting options for that particular set of values.

	Metrics	Profit	Profit Forecast	Revenue	Revenue Forecast
Books	Business	\$83,891	\$80,496	\$365,872	\$ 355,524
	Literature	\$67,512	\$64,932	\$302,789	\$ 294,026
	Books - Miscellaneous	\$43,764	\$42,670	\$223,600	\$ 214,875
	Science & Technology	\$39,654	\$38,640	\$236,491	\$ 226,177
	Sports & Health	\$138,649	\$126,684	\$610,623	\$ 594,194
	Total	\$55,855	\$53,233	\$250,526	\$ 228,743
	Total	\$429,326	\$406,655	\$1,989,903	\$ 1,913,538
Electronics	Audio Equipment	\$477,908	\$438,436	\$2,859,428	\$ 2,758,334
	Cameras	\$681,959	\$651,348	\$3,823,372	\$ 3,655,107
	Computers	\$253,011	\$248,508	\$1,441,638	\$ 1,395,727
	Electronics - Miscellaneous	\$610,457	\$607,888	\$3,521,771	\$ 3,285,087
	TV's	\$507,682	\$464,228	\$2,862,817	\$ 2,676,106
	Video Equipment	\$701,256	\$685,691	\$3,854,434	\$ 3,644,328
	Total	\$3,232,273	\$3,096,101\$18,363,460	\$ 17,414,690	
Movies	Action	\$28,344	\$26,321	\$464,309	\$ 462,286
	Comedy	\$24,937	\$23,045	\$502,380	\$ 488,354
	Drama	\$31,944	\$30,320	\$527,088	\$ 490,438
	Horror	\$29,557	\$28,599	\$476,950	\$ 439,849
	Kids / Family	\$30,195	\$28,596	\$501,034	\$ 460,424
	Special Interests	\$47,078	\$45,590	\$614,588	\$ 568,142
	Total	\$192,054	\$182,471	\$3,086,349	\$ 2,909,492
Music	Alternative	\$6,807	\$10,910	\$531,459	\$ 500,995
	Country	\$5,051	\$9,884	\$551,331	\$ 540,058
	Music - Miscellaneous	\$46,302	\$44,895	\$451,501	\$ 443,880
	Pop	\$14,763	\$15,787	\$516,461	\$ 497,482
	Rock	\$8,322	\$11,796	\$528,724	\$ 510,976
	Soul / R&B	\$54,586	\$52,568	\$357,282	\$ 336,523
	Total	\$135,829	\$145,839	\$2,936,758	\$ 2,829,914
	Total	\$3,989,483	\$3,831,066\$26,376,470	\$ 25,067,635	

Figure 17-34 Users can format report grids and graphs with MicroStrategy Web.

Display Report Information in Graphical Format in MicroStrategy Web

Users often need to visualize data in graphical formats. A well designed graph can quickly convey more information than a large grid of information by using visual aids to contrast areas of interest or concern. MicroStrategy Web allows users to quickly display grid reports in graph format using a simple one click menu action.

MicroStrategy Web offers developers and power users the ability to quickly change the formats of graphs that they create or to edit existing reports. Business users have the power to format the graphs exactly as they want, without having to rely on IT developers. Through the graph formatting toolbar and dialog boxes, many graph properties can be adjusted, including:

- Graph types
- Graph subtypes
- Legends
- Axes scales and labels
- Fonts
- Series and category options

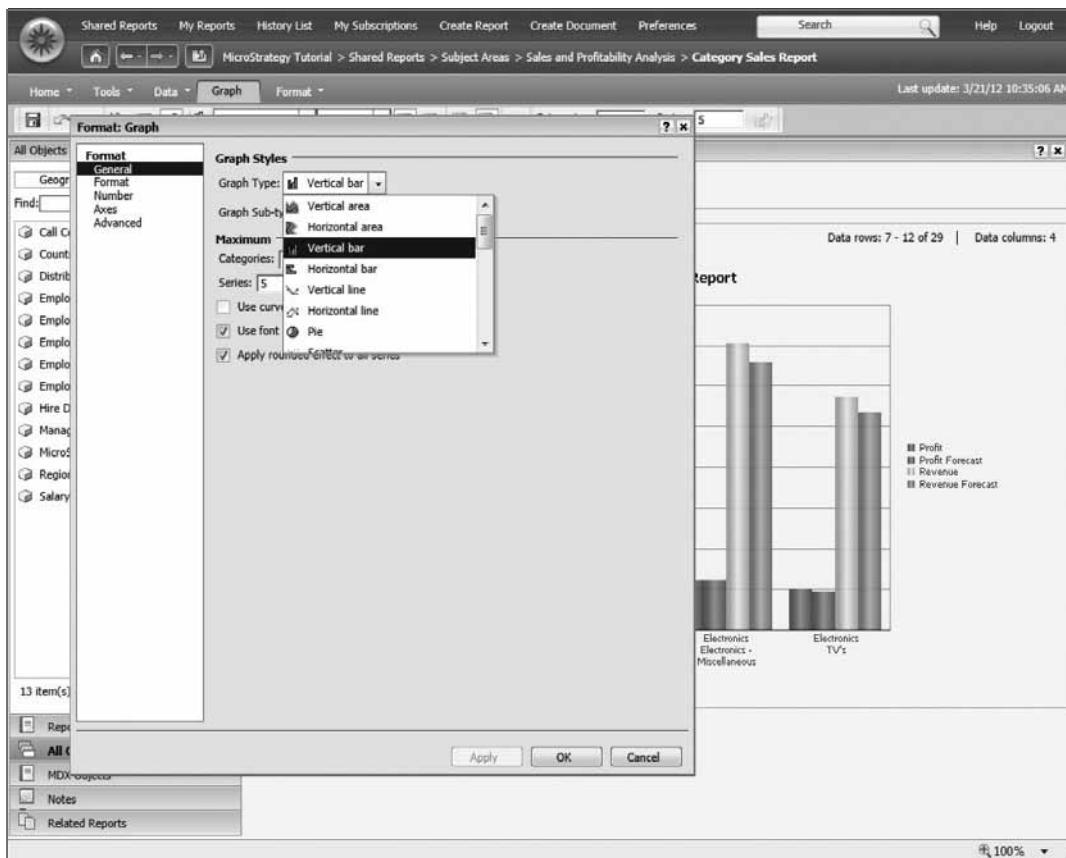


Figure 17-35 Powerful graph formatting over the Web in MicroStrategy Web.

17.17 CREATING PIXEL-PERFECT DASHBOARDS AND SCORECARDS OVER THE WEB

In addition to creating grid and graph OLAP reports, well formatted documents can be defined in MicroStrategy Web. Developers can add datasets, drag and drop fields, insert grids and graphs, and use powerful formatting to create print-perfect scorecards, dashboards, Mobile apps, and operational reports. The editors and techniques used to create these reports are similar to the document editor.

Users have the option of choosing either a blank template or any other existing corporate template as a base for the creation of these documents. By providing a set of standard corporate templates, developers control common fonts, headers, footers, and other corporate formatting standards.

Datasets are then added to the document. Any MicroStrategy report, regardless of where the data is retrieved, can be used in a Report Services document. If more than one dataset is used, the information is automatically merged and additional metric calculations across datasets are possible.

All attributes and metrics in the datasets are added to the document by dragging and dropping them onto the layout template. They can be placed in standalone fields in the detail section and also into various bands and zones as defined by the document grouping. Precise control of the object placement and size defines the appropriate formatting and layout required. The documents are saved into the standard folder structure for other users to access.

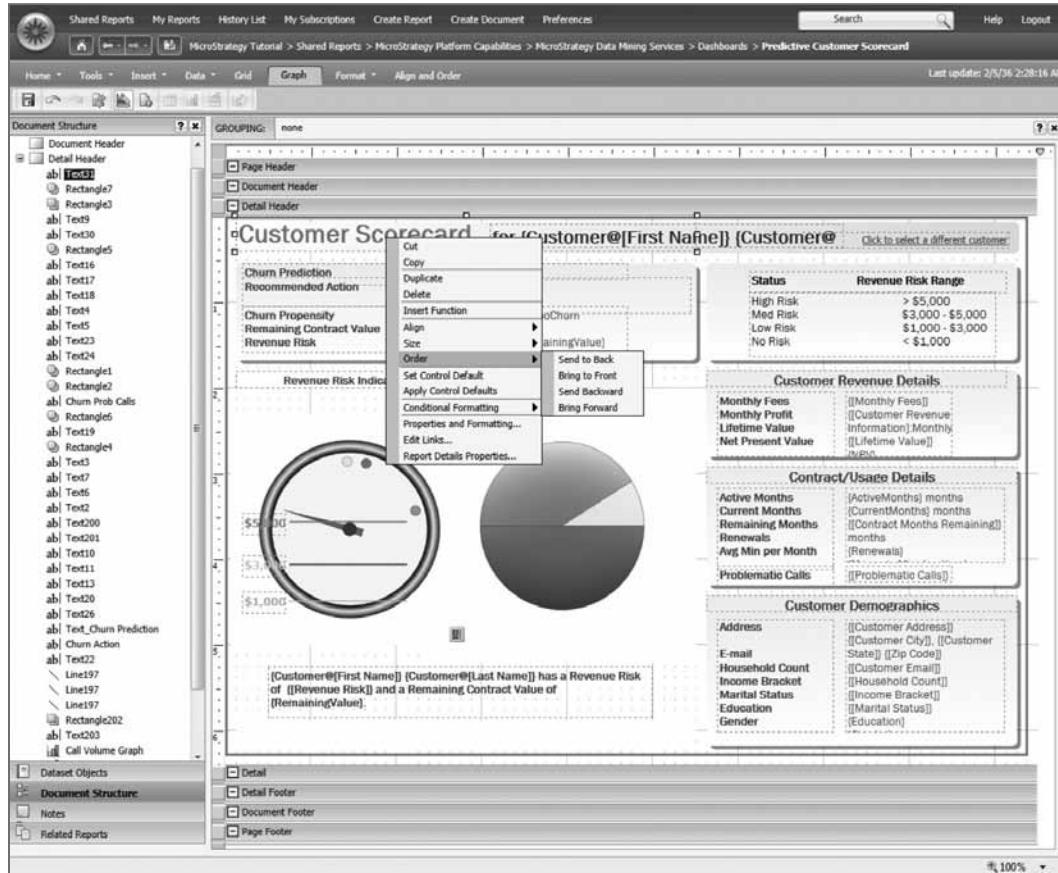


Figure 17-36 MicroStrategy Web offers an array of formatting and layout options for documents in the Document Editor.

WYSIWYG Editing of Reports and Dashboards Over the Web

In addition to the creation of new documents and scorecards, developers and users have the ability to edit these documents in a What-You-See-Is-What-You-Get (WYSIWYG) manner over the Web. Users can change formatting, layout and other parameters of a report to tailor the layout and contents to their needs. This drastically reduces report development times by eliminating endless iterations with report developers in IT departments.

MicroStrategy Web dramatically increases the number of users who have access to powerful ad hoc, as well as WYSIWYG, report editing and creation abilities. The interface is intuitive and is used with minimal training by any user. In conjunction with a secure architecture and a true thin client interface, this presents an opportunity to distribute development teams geographically and by business units, reducing the number of IT developers needed to maintain the application.

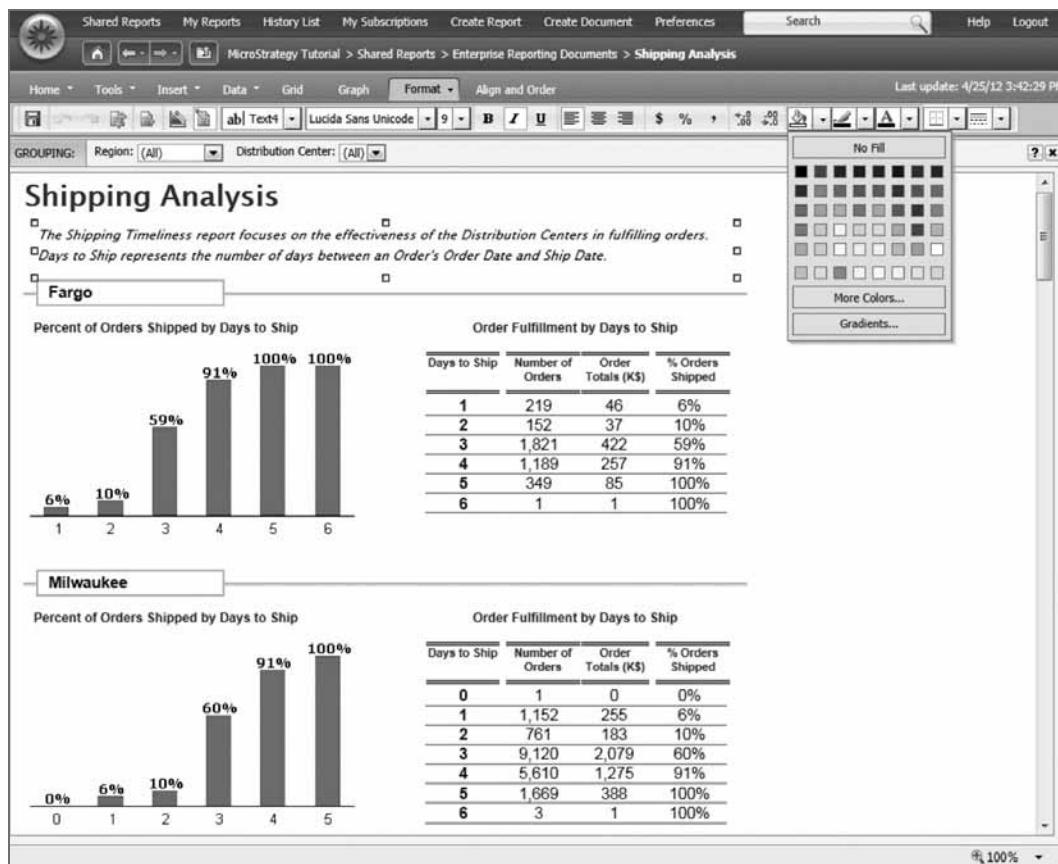


Figure 17-37 WYSIWYG Editing changes the layout and format of Report Services Documents directly.

17.18 TRANSACTION DOCUMENTS

By utilizing transaction reports, report developers can start building a Report Services document that can be accessed through MicroStrategy Web or through a Mobile device such as Apple's iPhone and iPad and Android phones and tablets. Depending on whether users want to manipulate (update or delete) existing data or insert new data to the transactional data source, they will employ different approaches for building the dashboard. Transaction documents are especially useful when they are accessible through a mobile device. By accessing transaction documents through a mobile device, users can input data whenever they need to wherever they are.



Figure 17-38 A Transaction document can display data and collect user input.

Manipulating Existing Data

If users want to manipulate existing data, they must first display it on a dashboard using a query report. It can be displayed as a grid or as an Interactive Grid widget (available only for mobile devices). Users also need to enable some or all of the data cells to be editable. These cells can be displayed as drop-down lists, text areas, toggle buttons, or any of the supported data input controls.

For example, business users might want to approve or deny time off requests. This could be done by displaying queued requests in an Interactive Grid widget. Then, to approve or deny the request, users would just toggle the approve/deny buttons, type any additional comments in a pop-up text area, and submit the transaction.

Inserting New Data

If users want to insert new data, they do not need to display existing data on the dashboard. Instead, users can define a set of input fields (text boxes), place them on a panel stack, and display them as editable.

Potentially, they may want to display some data from a query report that uniquely identifies the current transaction or provides additional insight to the user, but it is not necessary.

For example, users might want to enter inspection details for a store they are visiting. On the document, users could display the name and address of the store (retrieved from a data source with a query report) which can be presented as a set of sliders, drop-down lists, and a text area to enter their observations, and submit the inspection back to the transactional data source. To configure a transaction, right-click on the grid or text field -> click on 'Configure Transaction'. In this dialog, the transaction report is selected, input objects are linked to grid objects/text fields, and the Input control style is defined.



Figure 17-39 Transaction documents, accessed on a mobile device, make it very easy for users to input data on the go.

17.19 CUSTOMIZATION / SDK DEVELOPER

The MicroStrategy BI platform is an open platform built upon standards that allows developers to customize the look, feel, and functionality of the platform. Details of the customizability of the MicroStrategy Platform can be found in the chapter that details the Software Developer Kit (MicroStrategy SDK). The following section discusses some of the developer resources available to the Developer interested in customizing the look and feel of the MicroStrategy BI platform.

The MicroStrategy Developer Library

The MicroStrategy Developer Library contains over 6,000 pages of documentation and reference material, complete with API references, and how-to's for developers to quickly access information at the most detailed level.

Code Sample

The following steps help you understand the process of creating a Report Bean. The inline comments make the code self-explanatory.

```

//Step 1. Create Report Bean and Transform instances to be used later
WebBeansFactory factory = WebBeansFactory.getINSTANCE();
ReportBean rb = (ReportBean) factory.newInstance("ReportBean");
TransformInstance til = rb.addTransform(new
com.microstrategy.web.app.transforms.ReportGridTransformImpl());
til.setKey("grid");

//Step 2. Set I-Server Session, and set bean state through arbitrary name
rb.setSessionInfo(WebSessionInfo.myJesse);
rb.setName("MyReportBean");

//Step 3. Specify report to be used by name or ID, and call collectData()
rb.setObjectID("5A3434234290A8E2B721ABAC65818F47");
rb.collectData();

//Step 4. Render collected data
Map<String, Object> mo = rb.transform("grid");
StringWriter sw = new StringWriter();
mo.send(sw);
System.out.println(sw.getBuffer().toString());

```

Here are some important points to note in the code sample above:

- The `setName()` method in Step 2 is not explicitly necessary in this example. However, providing this line allows us to easily persist this bean by our specified name.
- `myJesse` is presumed to already exist either by creation from the `WebObjectsFactory` on this page, or by persisting a previous page's defined `WebSessionInfo`.
- The `setObjectID()` method is used in Step 3. Similarly, the `setName()` method can be used to set the ReportBean's state by the name of the report, rather than its ID.
- In Step 4, `ReportBean`'s `transform()` method passes the `TransformInstance` key simple to indicate that the associated transform should be used to render the bean's data. If a `TransformInstance` had called `setDefault()`, and `transform()` had been called without providing a key parameter, the default `TransformInstance`'s associated `Transform` object would be used to transform `rb`'s data.

The following code sample shows how to transform a bean using a layout definition file. The inline comments make the code self-explanatory:

Figure 17-40 The MicroStrategy Developer Library is in a handy electronic format.

The MicroStrategy Developer Library provides:

- Comprehensive documentation detailing the MicroStrategy BI platform architecture
- API reference, complete with input/output parameters
- Over 70 task-oriented examples with sample code
- Tools and tips to upgrade to the latest version of the MicroStrategy BI platform
- Case studies for integration with external applications
- Sample applications and utilities

APIs Available to the MicroStrategy Developer¹³

Type of API	Description
Web API	The Web API allows customization of the Web interface and integration with third-party applications. It enables modification of the look and feel of the Web interface, presentation of BI data and functionality available to end users.
Mobile API	The MicroStrategy Mobile API enables mobile app developers to customize MicroStrategy Mobile app on Android and iOS mobile devices by changing the look and feel of the app, creating custom visualizations, or developing new Mobile Apps that use MicroStrategy BI content.
Visualization API	The Visualization Framework lets application developers build new visualizations or extend visualizations to match specific business needs for dashboards rendered in DHTML or Flash.
Office API	The MicroStrategy Office API allows developers to build custom applications that use Microsoft Office products as a user interface to the MicroStrategy BI platform. These include connecting to a BI application, browsing folders and reports, specifying report parameters using prompts, and running and displaying reports and documents.

Figure 17-41 The MicroStrategy BI platform contains open APIs for all the major components.

17.20 SUMMARY

The MicroStrategy BI platform is designed to give developers of all skill levels, from system architects to Web-based power users, tools to build the most sophisticated BI applications. With tools for translating data models into business models, architects can map different enterprise data sources into terms that are readily understood by business users. Powerful statistical and custom grouping tools allow developers to build advanced predictive models as well as advanced OLAP metrics and groupings that can be used by other users. Web-based power users can then use these to create mobile apps, reports and formatted dashboards and scorecards. Developers can also build sophisticated transactional, alerting and batch reporting applications and customize MicroStrategy end user interfaces through a comprehensive, well-documented API set.

¹³See Chapter 19 Extensibility and the SDK for more information on customizing MicroStrategy

18

ADMINISTRATOR EXPERIENCE

As the scope of business intelligence systems expands with more data, more users and more applications, administration becomes a key driver in the lowering the total cost of ownership. There are different administrative roles in an enterprise. Sometimes all these roles are taken on by just one person, but often the responsibilities are spread over several departments or teams.

18.1 IN-DEPTH ADMINISTRATION ACROSS THE WHOLE PLATFORM

The administrators of business intelligence systems need to accomplish many tasks, including:

- Project Setup
 - Installing and configuring the BI system
 - Managing system performance
 - Controlling software licenses
- Access Control
 - Setting up users and their security profiles
 - Monitoring and enforcing security policies
- Operations Management
 - Ensuring data integrity after data loads
 - Maintaining a stable and reliable BI environment
 - Monitoring system usage to prevent performance bottlenecks
 - Troubleshooting user issues
 - Project Lifecycle Management
 - Managing BI applications in development, test, and production environments
 - Assessing the impact of changes to the system
 - Identifying and decommissioning unused BI objects

Experienced system administrators evaluate the administrative capabilities of BI systems using two key criteria:

1. Visibility — Administrators must access and track every key determinant of system performance
2. Controllability — Administrators must control every key parameter to tune the system to suit their own unique environment and to guarantee stability and performance, even under extreme conditions

Superior software architectures are set apart by explicitly incorporating visibility and controllability features into the system. Visibility is delivered through detailed and granular statistics, error logging, and diagnostic files. Controllability is achieved by providing a wide range of server and application tuning functions that let the administrator configure the system — both when first implementing the BI system and on an ongoing basis.

The architectural tenets for administering the MicroStrategy BI platform are:

- Allow completely centralized management and configuration of the business intelligence environment
- Allow fine-grained monitoring, logging, and control over the system's performance and use
- Provide fine-grained control over the types of operations various classes of users are allowed to perform
- Define the level of detailed diagnostic statistics to capture for troubleshooting purposes
- Provide real-time and historical analysis of system performance to identify trends and potential bottlenecks
- Provide impact assessment of proposed configuration changes and potential changes to the application definition
- Automate all repetitive tasks, and allow them to be initiated on time-based or event-based triggers
- Support bulk administration operations to simplify system changes, and reduce manpower investment
- Compare before and after scenarios when changes are made to the BI environment

The MicroStrategy BI platform meets all of these requirements. MicroStrategy Intelligence Server, MicroStrategy Mobile, and MicroStrategy Web all provide detailed administrative functionality and bring it all together for a BI system administrator.

Administrators overseeing a MicroStrategy BI system have complete visibility into and control over system performance and tuning. This detailed visibility and control is provided in an easy-to-use fashion and designed to be extremely powerful and efficient.

This section covers the MicroStrategy platform's capabilities in each of the following areas:

- Real-time monitoring
- Error logging and diagnostic files
- Enterprise Manager - Historical performance analysis
- Centralized administration
- Change Journaling
- Command Manager
- Object Manager
- Integrity Manager
- Web Administration
- System management software integration

18.2 REAL-TIME MONITORING AND CONTROL

At any point in time, the administrator can monitor system activity. MicroStrategy provides real-time information about important activities in the BI environment via the following administrator functions:

- Project Monitor — View existing BI projects and their status

- Cluster Nodes Monitor — View clustered machines and their availability
- Schedule Monitor — View a list of all scheduled reports and who scheduled them
- Jobs Monitor — View currently executing jobs and jobs containing errors
- User Connection Monitor — View active users and their session information
- Database Connection Monitor — View connections to the metadata repository and data sources
- Cache Monitor — View Intelligent Cube and report cache details with their cache expiration dates
- Change Journal Transactions Monitor — View the changes that have been made to objects in a project source.
- Scheduled Services — View the next time any report distribution service will be triggered
- Distribution Managers — View the currently executing report distribution services
- Debug Monitor — Receive real-time component-level statistics and debug information of the report distribution system

The administrator can also perform specific, related tasks directly from within these monitors, including:

- Discontinuing jobs — Stops errant runaway jobs that take up system resources
- Maintaining projects — Loads new projects, takes projects off-line, and adds projects to cluster nodes
- Disconnecting users — Removes inactive users
- Disconnecting database connections — Changes the number of concurrent database connections
- Expiring caches — Invalidates old reports, and frees up memory
- Cancel Service executions — Controls Distribution Services operations

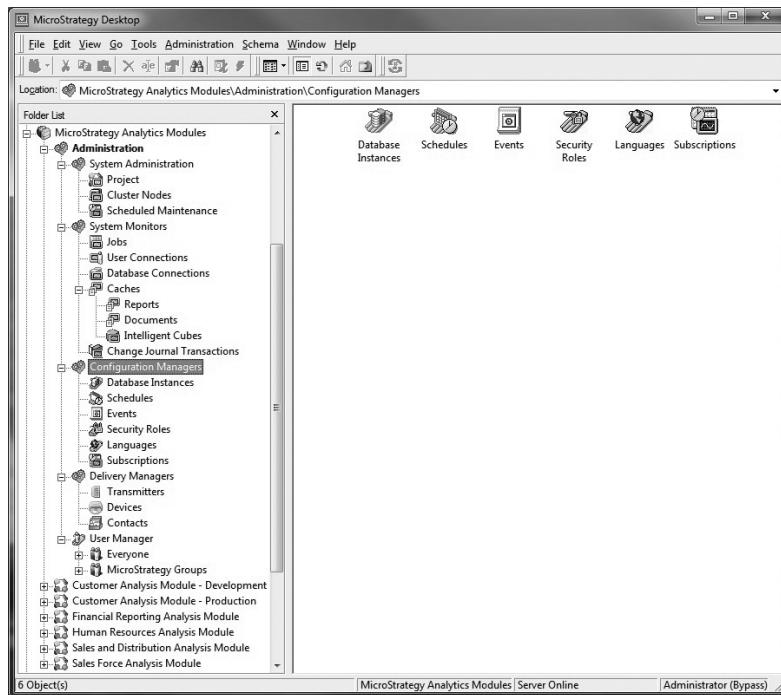


Figure 18-1 Intelligence Server administration and real-time monitoring from the Desktop interface.

Real-time monitoring of a range of operating system and Intelligence Server performance counters is also available using the Windows Performance Monitor and Event Viewer or the UNIX syslog facility for additional information about system performance. This integration is out-of-the-box and no additional work is required. Over 50 performance counters are traced in the following functional areas:

- Current jobs executing in Intelligence Server
- Current users connected to Intelligence Server
- Process resource allocated to Intelligence Server

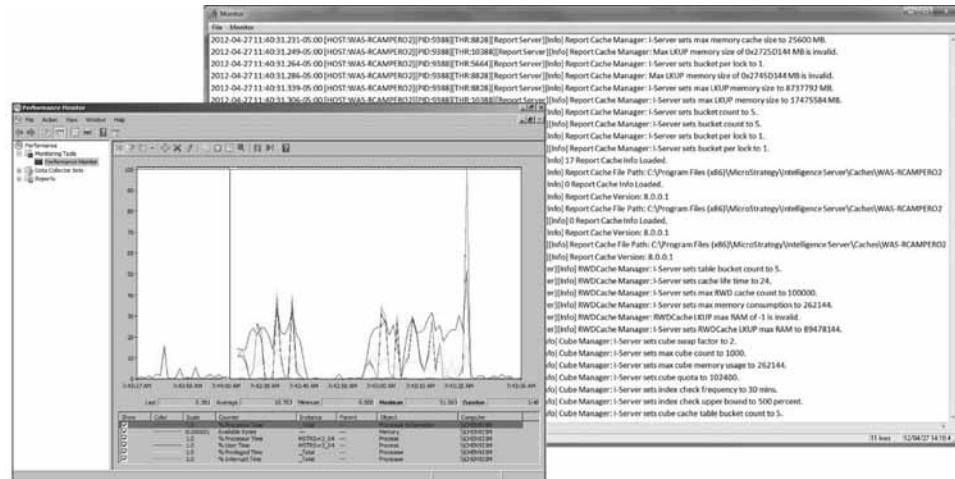


Figure 18-2 Monitoring the MicroStrategy BI environment.

18.3 ERROR LOGGING AND DIAGNOSTIC FILES

Log files and operational statistics enable historical data to be accumulated so that cause and effect analysis as well as chronological analysis can occur. The MicroStrategy BI platform enables various types and levels of logging, with complete control over the granularity of data being accumulated. Multiple log files are available to let the administrator analyze system performance. The administrator can set the level of logging to record statistical data, errors, warnings or informational messages, depending on the amount of information needed.¹⁴

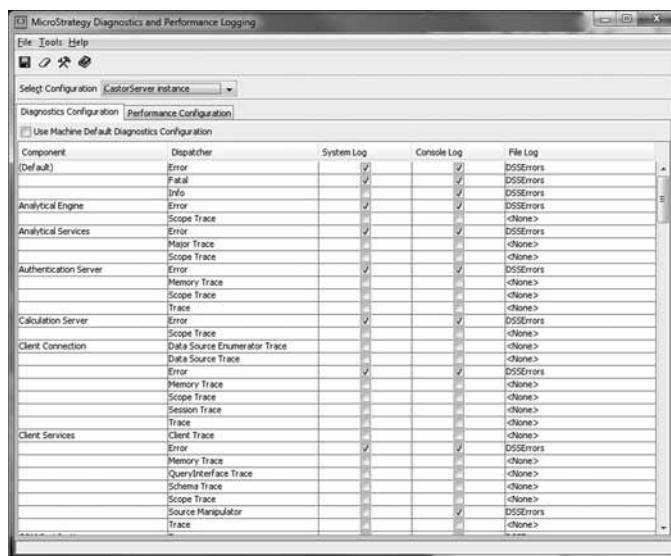


Figure 18-3 An administrator can control component level-logging from a centralized interface.

¹⁴See Appendix E Performance Counters and Key Performance indicators for details on performance counters in MicroStrategy

The administrator can view log file information from both the Windows Event Viewer and directly from flat files. Administrators can use these logs for the following purposes:

- An observation tool to monitor current activity
- An error display console to identify and correct system faults
- A diagnostic tool to tune the reporting environment
- An aid that allows MicroStrategy Technical Support to help customers resolve issues in a timely and accurate manner

In extreme situations, log files can grow to a point where they impact operations. To avoid these situations in a production environment, a maximum size for log files can be specified. For easier management of log files, MicroStrategy uses the implementation of time or size-limited rotation strategies. A maximum number of log files can be specified to control the overall required disk space used for logging and diagnostics.

18.4 MICROSTRATEGY ENTERPRISE MANAGER

In addition to viewing current system status using monitors and error logs, MicroStrategy can store historical information on system usage. This information is stored in a statistics database. Leveraging this rich historical view of the system, temporary anomalies in usage and performance which are of minimal importance, can quickly be distinguished from trends that may have a wider impact on the system. This type of long-term assessment is enabled through Enterprise Manager.

Enterprise Manager lets system administrators analyze, and act upon detailed and granular logs produced by the MicroStrategy BI platform. It incorporates a complete data warehouse, analytical application, a data loading process, and dozens of insightful system performance and performance tuning dashboards and reports.

Enterprise Manager uses MicroStrategy's own analytics to monitor and analyze usage statistics recorded by one or more Intelligence Servers. The Intelligence Servers capture user session and report usage statistics for each server, user, and reporting job. This information is written to a central database. The detailed information logged includes the servers, the users, the reports, and metadata objects, the time of submission, the execution time for various steps in report processing, and tables and columns that were accessed in the data warehouse.

Enterprise Manager provides over 150 reports categorized in four areas of analysis:

- Project analysis — Analyzes growth trends and displays summary and detailed information of the BI applications.
- Operational analysis — Analyzes user concurrency and project usage, and helps determine how system resources are being used by the MicroStrategy implementation.
- User analysis — Reports user connection activity and report execution history.
- Performance analysis — Reports detailed system performance trends, and provides insights into data warehouse tuning.

¹⁴See Appendix E Performance Counters and Key Performance indicators for details on performance counters in MicroStrategy

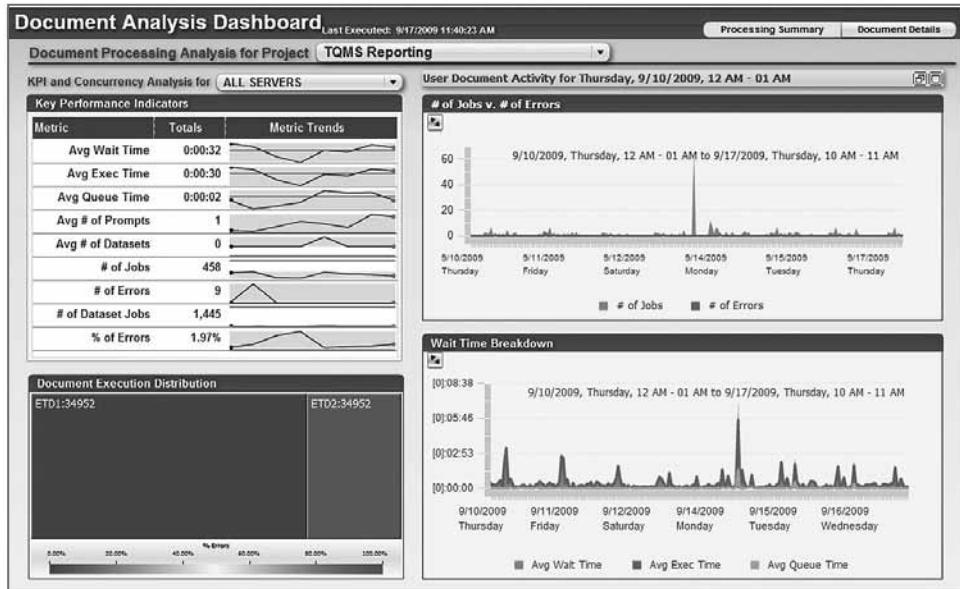


Figure 18-4 Project dashboard shows system usage at a glance in Enterprise Manager.

Many of the reports are parameterized to heighten flexibility, and speed up the analytical process. For example, some reports prompt for the time period to analyze (e.g., “today,” “last week,” and “last month”). Administrators are also able to track unread History List subscriptions to determine which History List subscriptions are not in use.

Administrative reports and documents are accessible from all MicroStrategy interfaces, and are contained in a standard MicroStrategy project in a metadata repository. The Enterprise Manager project can be extended to incorporate additional data sources, and additional reports can be created to reflect unique characteristics of the environment. The reports help the administrator perform the following tasks:

- Analyze usage trends and make decisions such as adding more hardware to the system
- Identify processing bottlenecks such as number of database threads
- Assess the effects of configuration changes and tuning strategies
- Identify the most active users and their time spent using the system
- Determine the best time to run scheduled reports in off-peak hours
- Identify the most popular reports, and determine which ones might need to be cached for optimum performance
- Track unread History List subscriptions to determine which History List subscriptions are not in use

With MicroStrategy Distribution Services, Enterprise Manager reports and documents can be sent to administrators and project managers via E-mail on a regular basis, providing an efficient mechanism for monitoring the MicroStrategy BI implementation.

18.5 CENTRALIZED ADMINISTRATION

Identifying bottlenecks or errors using real-time monitoring, error logs, and trend analysis is the first step in enhancing a BI system’s operational performance. Simple corrective action and granular control over the platform’s operations is vital to ensure service level agreements. The MicroStrategy BI platform contains

features, tools, and utilities that optimize, and manage all servers in the MicroStrategy environment from a single remote console.

Granular Parameters Provide In-Depth Server Configuration

Intelligence Server configuration capabilities can be used to manage system resources and to control user privileges.

1. System management and configuration – Each server can be configured individually to optimize the use of hardware resources by the server's components. With MicroStrategy, an administrator can:
 - Create connections from Intelligence Server to the databases
 - Add or delete servers from a server cluster
 - Expire or delete caches
 - Manage job requests
 - Set backup frequency
 - Configure memory settings
 - Create and trigger report schedules
 - Configure query governing settings
 - Define job thread prioritization
 - Set the level of diagnostic reporting and logging
2. User and security profile management – User application privileges can be set for individual product features. Data access for any user can be customized down to the row and column level. Setting these controls is simple and involves selecting check boxes or running pre-built scripts. No custom coding is required. Additionally, administrators can use graphical interfaces and text-based commands to:
 - Create and manage users
 - Define the user authentication mechanism
 - Organize users into groups and roles
 - Assign functionality privileges to users and groups
 - Assign data permissions to users and groups
 - Assign object access to users and groups
 - Manage user connections to the business intelligence projects

Information Governing Allows Real-Time Control of the Processing Environments

An important consideration in implementing and deploying business intelligence applications is managing the number of users, number of job requests, and amount of data that a server in the environment can process. In a MicroStrategy BI implementation, administrators can set limits to prevent servers from becoming overloaded by requests when extraordinary conditions exist. For instance, administrators can control the maximum number of users logged in at any time. This capability ensures that the servers in the MicroStrategy environment run efficiently at all times.

To ensure that important reports and users are processed ahead of others, priority levels can be set according to the report, the user group, the application being accessed, or the interface being used. Items with higher priority are processed before lower priority items. Governors can be applied at the server or application level.

The list below shows many of the governor settings available in Intelligence Server.

- Server Level Governing
 - Maximum amount of memory used by Intelligence Server
 - Maximum amount of memory for in-memory report caches
 - Maximum number of job requests
 - Maximum number of user sessions
 - Maximum user session idle time
 - Maximum number of XML cells
 - Maximum number of XML drill paths
 - Maximum memory consumption for XML
 - Maximum memory consumption for PDF
 - Maximum memory consumption for Excel
 - Priority of reports based on job cost
 - Priority of reports based on user group
 - Priority of reports based on project
 - Priority of reports based on request type
 - Priority of reports based on requesting application
 - SQL time-out
 - Maximum number of History list messages per user
 - Number of days History list messages are active
- Application Level Governing
 - Maximum report execution time
 - Maximum number of report result rows retrieved
 - Maximum number of prompt elements retrieved
 - Maximum number of intermediate result rows retrieved
 - Maximum job requests per user
 - Maximum job requests per session
 - Maximum job requests per application
 - Maximum user sessions per application
 - SQL time-out per report

Effective Caching is Vital for Good System Performance

Caching speeds up operations by storing data and report results in faster, more accessible locations. A sound caching policy should be established that will maintain the integrity of all cached data – documents, reports, prompt lists, and metadata objects. The MicroStrategy BI platform supports this in multiple ways.

- Monitoring – Detailed information about each cache file is available in real-time including creation time, status, type, size, the number of times accessed, the last time accessed, and expiration date. Enterprise Manager provides more analysis into the use of report caches, the cache hit ratio by report or user, and it identifies underutilized caches.
- Managing Data Integrity – After data has been loaded into the data warehouse, all relevant cache files should be invalidated to ensure data integrity. This task can be automated by triggering time- or event-based schedules. For more granular control, cache invalidation can be tied to changes to data warehouse tables.
- Tuning – Cache creation for the most frequently accessed and most important reports can be scheduled to run in off-peak hours. By monitoring the cache hit ratio, administrators can evaluate the effectiveness of their currently implemented caching strategy and find the optimal balance between end-user performance and cost of added batch window time.
- Governing – Intelligence Server maximizes caching efficiency and performance by storing the caches directly in memory. In order to ensure optimal use of server resources for all job processing, the amount of memory Intelligence Server dedicates for caching and the total number of cache files it can handle can be specified.

Using any or all of these techniques, business intelligence applications will provide optimal performance to users while ensuring data integrity. Any project can be managed with several data load schedules providing near real-time business intelligence.

Tracking Software License Deployment

The MicroStrategy BI platform helps to keep track of all deployed software licenses. An automatic daily scan aggregates the user privileges and reports if software licenses are being exceeded. The scan can also be run manually using the License Manager utility, which also displays details about the software installed on the machine, and the licenses allocated for users. Using the License Manager, one can:

- View system information and the version, edition, expiration, installation history, and other details of the products installed on the machine
- Determine the product licenses in use by a specified user group or user
- Print the license usage information or export it in HTML format.
- Upgrade licenses by providing the new license key, without reinstalling the products
- Activate or deactivate a MicroStrategy server installation.

18.6 CHANGE JOURNALING

Keeping track of the changes made to objects in a large MicroStrategy system can be a difficult task. Change journaling tracks the changes to each object in the system. This makes it easier for administrators to quickly determine when and by whom certain changes were made. For example, reports using a certain metric executed correctly in a test two weeks ago, but no longer execute correctly in this morning's test. The administrator can search the change journal log to determine which developers made changes to that metric within the last two weeks.

Certain governmental regulations, such as Sarbanes-Oxley in the United States, require detailed records of changes made to a BI system. Enabling change journaling on all projects in a production environment can aid in compliance with these regulations.

Tracking Object Changes in Business Intelligence Systems

Change journaling is the process of logging information about changes to objects in a project. The logged information includes items such as the user who made the change, the date and time of the change, and the type of change (such as saving, copying, or deleting an object). With change journaling, enterprises can keep track of all object changes, from simple user actions such as saving or moving objects to project-wide changes such as project duplication or merging projects.

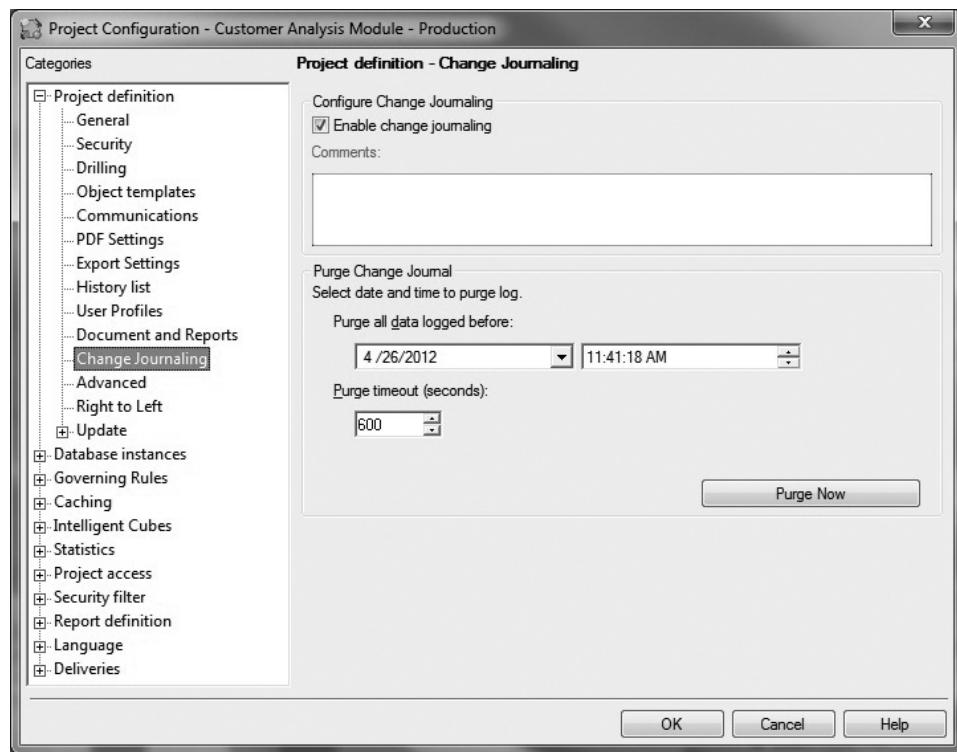


Figure 18-5 An administrator can easily enable or disable Change Journaling, as well as configure the Purge Change Journal settings.

When change journaling is enabled for a project, Intelligence Server records information in the change journal about any change made to any object in the project. This includes changes made in Desktop or Web as well as through other MicroStrategy tools such as Command Manager or Object Manager. Change journaling can also be enabled at the project source level. In this case, information about changes to the project configuration objects, such as users or schedules, is recorded in the change journal.

Users can enable change journaling for some or all projects in a project source. If change journaling is enabled for a project source, changes to the configuration objects in that project source are logged. If change journaling is enabled for a project, changes to all objects in that project are logged.

Given the fast rate at which the change journaling records grow, users can keep the size of the change journal at a manageable size by periodically purging older entries. When change journal is purged, a date and time needs to be specified. All entries in the change journal from prior to that date and time are deleted. This can be done for an individual project or for all projects in a project source at once.

Each Change Journal entry contains the following information:

- Object Name: The name of the object that is changed
- Object Type: The type of object changed. For example, Metric, User, or Server Definition

- User Name: The name of the MicroStrategy user that made the change
- Transaction Timestamp: The date and time of the change, based on the time on the Intelligence Server
- Transaction Source: The application that made the change. For example, Desktop, Command Manager, or Scheduler
- Transaction Type: Information on the transaction. For example Enable Logging, Save Object, Create Object
- Change Type: The type of change that was made. For example, Create, Change, or Delete
- Project Name: The name of the project that contains the object that was changed
- Machine Name: The name of the machine that the object was changed on
- Transaction ID: A unique 32-digit hexadecimal number that identifies this change
- Session ID: A unique 32-digit hexadecimal number that identifies the user session in which the change was made
- Link ID and Comments

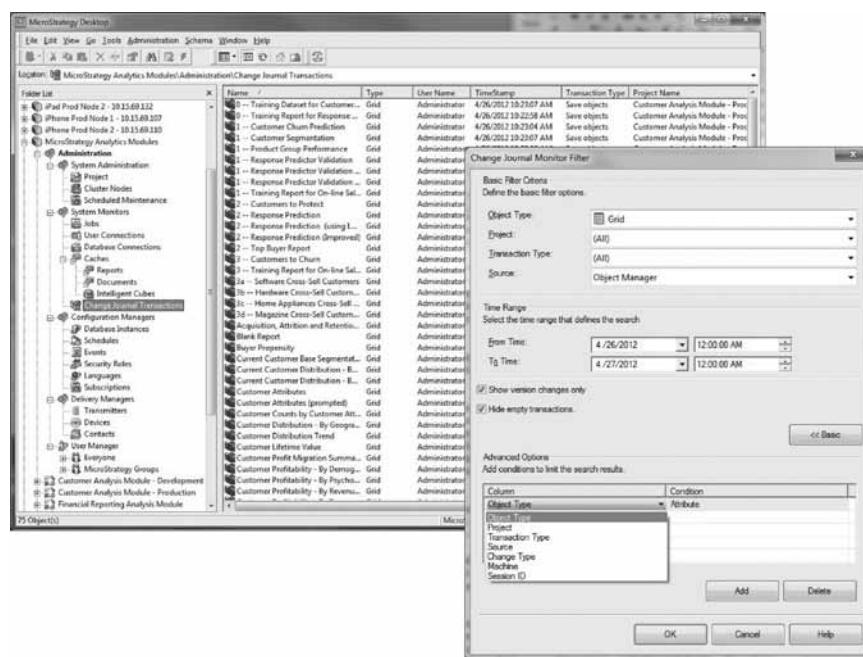


Figure 18-6 Information displayed for every Change Journal entry can be customized.

Changes can be viewed for each metadata object by going into the Properties section to access the change Journal. Additionally, a Change Journal Transactions monitor is provided under System Monitors that gives a view of all the changes that have occurred in all the projects and the entire configuration.

Interacting with Existing Change Journal Entries

Since the change journal records every transaction, finding the relevant records can sometimes be daunting. To make searching the change journal easier, users can filter it so that only relevant entries are seen. For example, users can quickly filter the entries so that they see only the entries for a specific object, or only the changes made by a specific user.

Contents of the change journal can be exported to a tab-delimited text file. This file can be saved to an archival location or E-mailed to MicroStrategy technical support for assistance with a problem. The name of

this file is AuditLog_MMDDYYhhmmss.txt, where MMDDYY is the month, date, and last two digits of the year, and hhmmss is the timestamp, in 24-hour format.

When exporting the change journal, any filters used to view the results of the change journal are also applied to the export. If the entire audit log is exported, make sure that no filters are currently in use.

18.7 MICROSTRATEGY COMMAND MANAGER

MicroStrategy eases the administrative workload through its Command Manager component. Command Manager provides scripting capabilities using text commands, bringing repeatability and automation to the administrative processes and featuring six core strengths:

1. Scalability – Recurring administrative tasks that contain large volumes of changing data, such as user, security, and report cache management, is much quicker with command scripts than in an interactive graphical environment.
2. Consistency – Often, commands in a script are essentially a repeat of the same pattern applied to a set of variables. Managing large user populations with changing security roles and profiles is simplified using third-party tools, such as SQL, that can easily be used to generate a large number of similar commands. Scripts also flawlessly execute large numbers of commands in sequence.
3. Efficiency – While commands duplicate the actions taken in a graphical user interface, running commands in a script removes the time between clicks. Repetitive tasks or generally large numbers of steps are significantly expedited. Asynchronous execution of scripts is fully supported.
4. Reliability – Script files can be fully tested before being used in a production environment. Archiving script files have the added benefit of documenting the history of setup and maintenance activity in the BI system.
5. Automation – Script files can be run automatically by a third-party utility or the operating system without human oversight. Critical and routine tasks can be scheduled to run during off-peak hours, freeing up resources during busy periods.
6. Cost savings – Administrators can focus on higher level tasks instead of being tied up with routine tasks. Both small and large deployments of the MicroStrategy BI platform can be managed with minimal administrative overhead.

Key administrative functions are accessible and controllable through commands that can be run from a command line, command line editor, or a graphical user interface. These commands can be saved in script files for reuse; script development in the graphical user interface is simplified though the highlighting of the words in the command syntax. Furthermore, a wide variety of searchable script outlines relieves the burden of memorizing the command syntax.

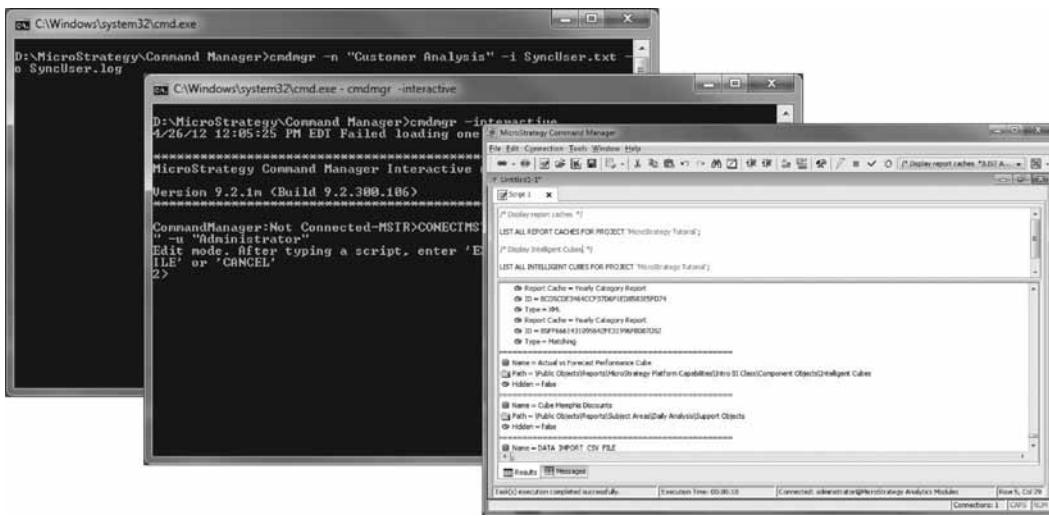


Figure 18-7 Command Manager runs commands from the command line, command line editor, or a graphical user interface.

Command Manager contains over 200 commands that fall into the following categories:

- Security Management
 - Create and manage users and groups
 - Maintain security filters and roles
 - Grant and deny object access and privileges
 - Set up user profiles and connection maps
- Operations
 - Monitor jobs, users, and database connections
 - Set governing properties on projects and servers
 - Manage server clusters
 - Manage report and table-level caches
 - Trigger events, schedules, subscriptions and report delivery services
 - Run operating system commands
 - Send messages to Windows Event Viewer
- Project Maintenance
 - Configure server and project parameters
 - Create and manage folders
 - Create and manage hierarchies
 - Create and manage filters
 - Create and manage facts and metrics

- Create and manage attributes
- Create and manage custom groups
- Create and manage Intelligent Cubes
- Create and manage reports
- Create and manage documents
- Create and manage subscriptions
- Import packages

Command Manager Procedures

Command Manager introduces a new way of developing automated tasks by using Procedures. Tasks commonly used together can be combined via Java programming to create procedures. Procedures save time by automating processes that require multiple complicated steps for completion. Procedures encapsulate one or more Command Manager Commands and business rules, programming logic, best practices and/or recommended actions. At the same time, the syntax for execution of Procedures would be similar to regular Command Manager Commands.



Figure 18-8 Procedures encapsulate Command Manager Commands and java programming code.

As a very simple example, a Procedure could be that of creating a user, which could involve the following actions - creating the actual user, adding the user to a user group and creating a security profile for the user in all the projects. From the point of view of a typical Command Manager administrator, the execution of procedures would be similar to execution of individual commands. By using Procedures, all the tasks above can be encapsulated into a single command as shown below:

```
EXECUTE PROCEDURE CREATE_USER ("user options", "group names", "project names");
```

Command Manager Procedures could be thought of as extensions to the available Command Manager Commands in that many commands can be combined and integrated with other business processes to meet end-user needs. Command Manager Procedures allow the user to better use Commands to suit their needs and essentially builds on the Commands and provides a level of abstraction for end users. Command Manager Procedures, just like scripts, can be made once and reused. They can be updated and versioned.

Command Manager supports User Procedures i.e. Procedures that are created by the administrator. To help the user in building User Procedures, Command Manager provides sample Procedures which the administrator can take and develop their own User Procedures.

18.8 MICROSTRATEGY OBJECT MANAGER

Object Manager is a graphical interface used to interactively manage metadata objects in related projects. Through a single, searchable console with drag-and-drop functionality, metadata objects such as facts, attributes, hierarchies, metrics, filters, reports, and other reporting objects can be migrated, copied, or deleted quickly and accurately. Object access permissions can also be set in Object Manager.

Understanding the effect of changes to metadata objects is vital to managing BI applications effectively. Object Manager determines the impact of modifications made to any MicroStrategy object by displaying parent and child objects. For example, if the definition of a metric has changed, an administrator can easily view the parent objects (reports, documents, templates, filters, and other metrics) and child objects (facts, filters, tables, and columns) of that metric with a click of a button, showing all objects affected by changing the metric. With this information, all affected users can be notified before the change is made. Impact analysis is also critical for documenting changes to BI applications, and accurately estimating the resources required to implement changes and upgrades.

The following tasks are available in the Object Manager interface:

- Manage report objects
- Share report objects
- Copy projects and make backups
- Compare projects in different environments
- Promote objects from development to test to production
- Search for objects based on name, type, owner, and time of last update
- Search for unused objects
- List parent and child object dependencies
- Create and Import Packages

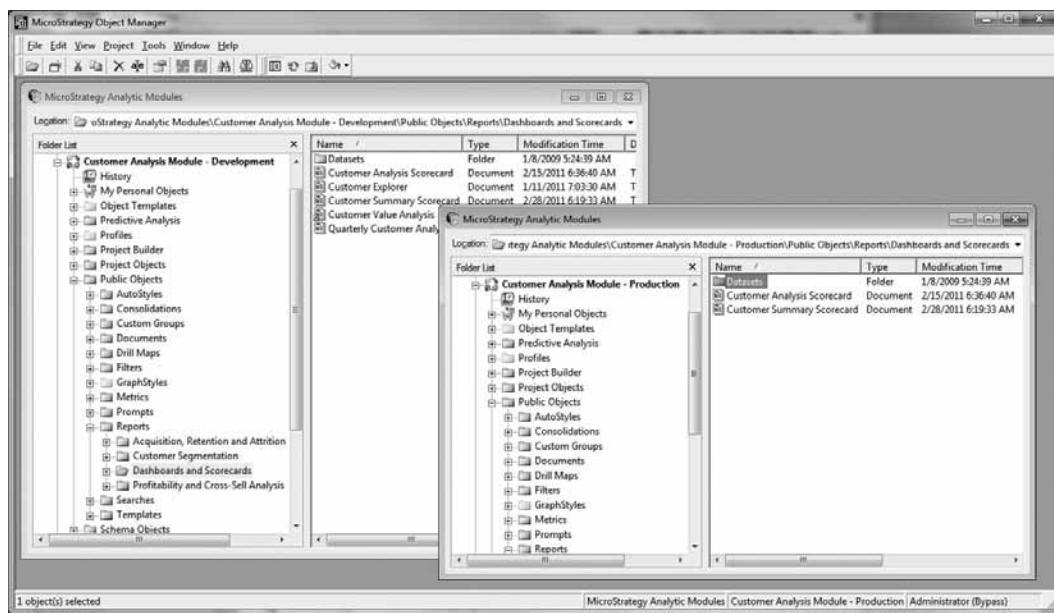


Figure 18-9 Object Manager simplifies moving objects between projects.

Upgrade BI Applications with the Project Merge Utility

When upgrading a production project with application changes from a development environment, project managers do not want to rely on manual drag and drop operations; instead, they prefer an efficient automated process.

MicroStrategy provides this ability with the Project Merge utility, a command-line utility that automates the simultaneous migration of all project changes. Conflicts can be resolved through the configuration of rules that are stored in an XML file and reused with each merge. The set of rules are defined through the Project Merge Wizard in four different levels:

- Object Category
- Object Type
- Folder
- Object

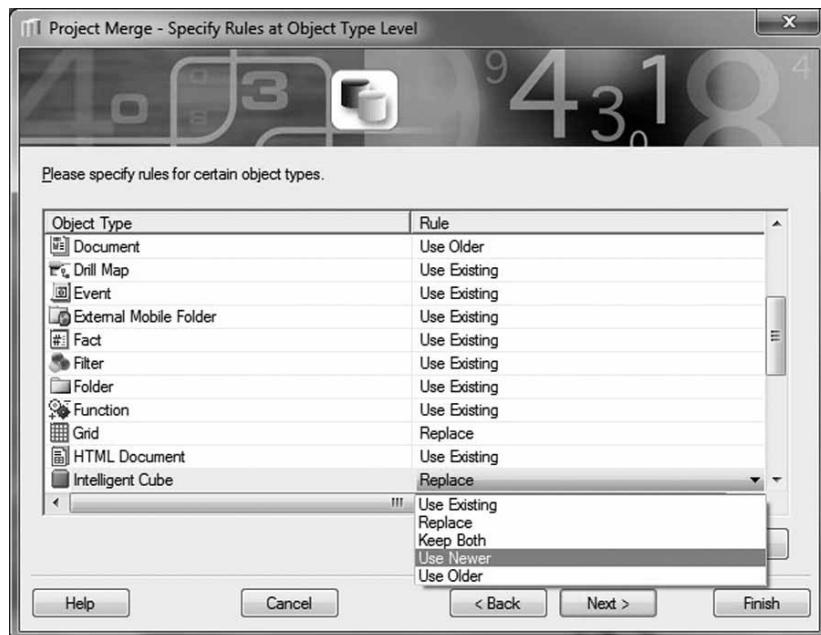


Figure 18-10 Specifying upgrade rules by object type in the Project Merge utility.

Migrate Changes with the Update Package Utility

Using this feature, report developers, architects, or administrators can extract object definitions from the metadata into a file with an extension .mmp. These package files can then be shared via E-mail or over a network like any other regular files. Object changes can be migrated without a simultaneous live connection to the source and target projects. Packages can contain many different types of objects such as: facts, metrics, attributes, reports, and documents just to name a few.

Update Packages provide the following benefits:

- Eliminates the need to have a simultaneous live connection to the source and target projects

- A Package exists as an exported standalone flat file with an .mmp extension
- Conflict resolution rules can be bundled into the package along with the object definitions
- Packages can be simply shared (via E-mail or through a network drive) between users
- Changes can be migrated into the destination project by “importing” (or applying) a package
- Packages can be applied using Command Manager scripts

To create a package, a connection to the source project has to be established, and the package creation editor has to be launched. At this point, all objects to be migrated can be added to the package. Dependent objects can be bundled into this package. This is particularly useful if a change made to an object involves changes to its dependent object(s). For example, if a report definition is changed which in turn led to changes to filters, metrics, prompts, etc. used in the report, there’s no need to track all of the changed dependents. All the dependents of that report can be simply added into the package. Once all the dependents are added, an user can set conflict Resolution rules (Replace, Use Existing, Use Newer, Use Older) on an individual line item thereby bundling the resolution rules into the package itself.

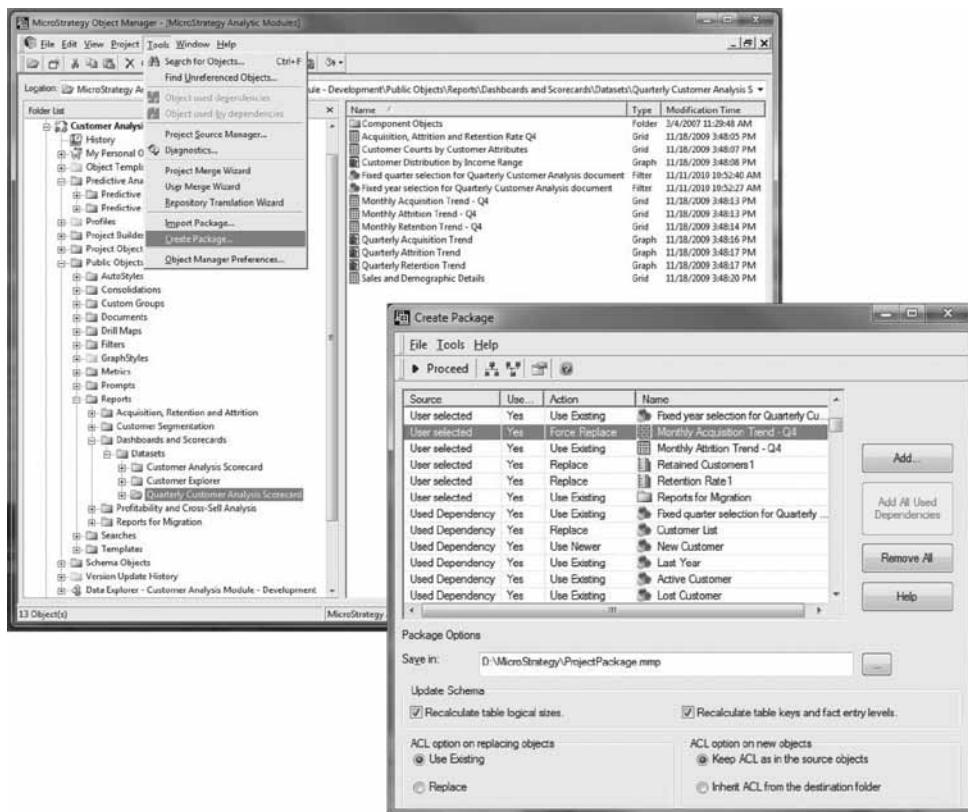


Figure 18-11 An Update Package is a simple way to migrate changes to a different metadata repository.

Repository Translation Wizard

The Repository Translation Wizard is a tool used to extract metadata objects into an external database for translation of metadata object names. This Wizard allows administrators to translate metadata objects in bulk in an external database. It is important to note that this tool does not actually translate the objects but provides a gateway for administrators/architects to extract objects from the metadata and then, translate them with a third party tool into the desired locales.

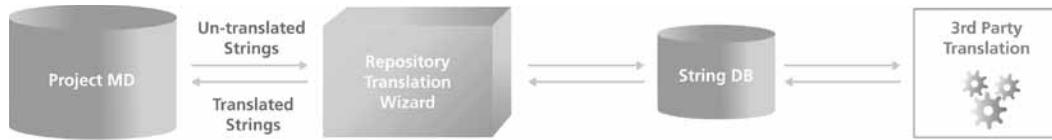


Figure 18-12 Integration of the Repository Translation Wizard with a third-party translation tool.

Main features of the Repository Translation Wizard are:

- Objects can be translated from one project at a time
- Support for command line execution
- Each extraction can be stored in a separate table
- While importing, administrators have the ability to filter out unapproved translations

Life-Cycle Management is Crucial to Smooth Application Deployment

Best practices in application deployment suggest an environment where development, testing, and production environments are clearly defined and separated from each other. An application is allowed to progress to the next stage only when it has clearly met the objectives and requirements defined for a particular stage. To enable a smooth and quick transition from one stage to another, an enterprise-class business intelligence platform must contain a set of application development, and management tools that allow clean and easy migration of business abstraction objects from one stage to another.

MicroStrategy's administration tools can manage many reporting applications and information delivery systems. The reports and other metadata objects are first developed in a development environment and tested under controlled conditions in a separate environment, before being deployed to the end users in the production environment.

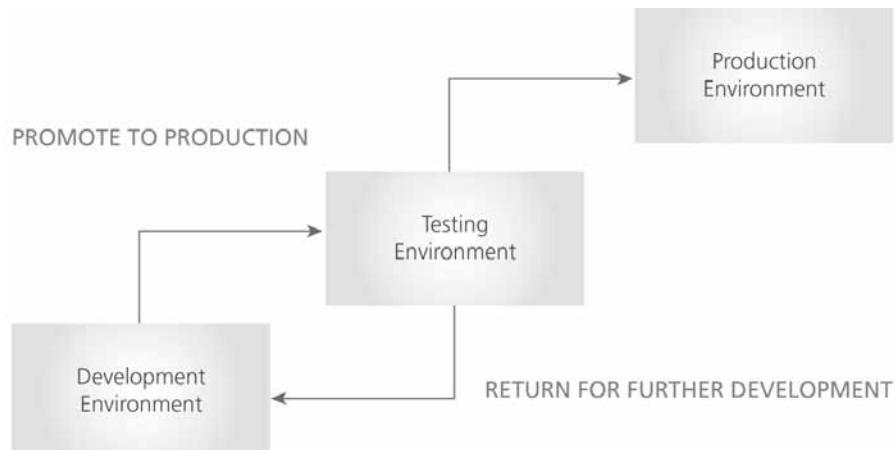


Figure 18-13 Migrating objects in the MicroStrategy BI platform.

The object management capabilities of the MicroStrategy BI platform were designed to support the migration of objects in various environments. The user interfaces make this process as easy as copying a file from one folder to another using a right-click menu or by dragging and dropping the objects. This simple task masks the complexity inherent in maintaining the integrity of every object in MicroStrategy's dynamic object-oriented architecture. Objects depend on other objects to complete their own definitions. Reports depend on metrics, attributes, templates, and filters. Metrics depend on facts, functions, and other filters. Because of these dependencies, copying an object, such as a report, without also copying the objects on which it depends, would produce an invalid object in the destination or target system, and the report would be unusable.

MicroStrategy automatically locates and migrates all of the dependent objects used by the object being migrated, ensuring the integrity of objects in all environments. When a report is migrated from one environment to another, any templates, filters, attributes, and metrics used by that report are also migrated. Since each of these objects can also have further object dependencies, MicroStrategy recursively ensures that objects used by those objects are migrated as well, until all required objects are copied.

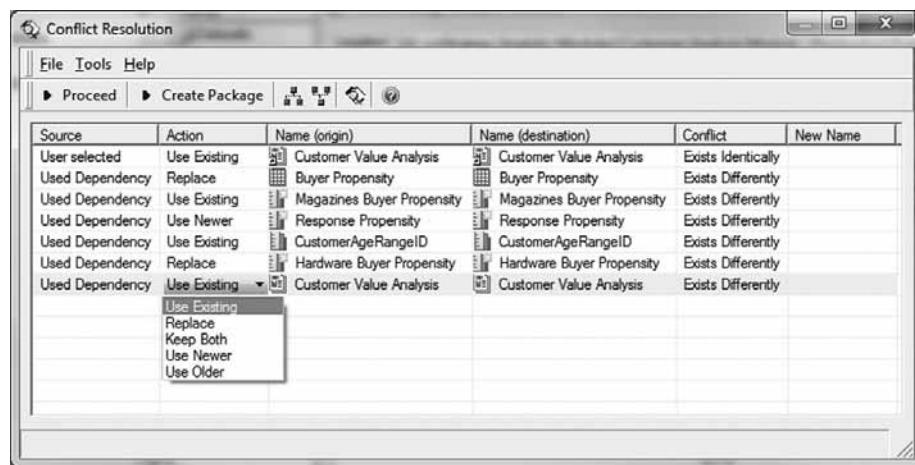


Figure 18-14 Overriding the default conflict resolution rules for specific objects.

18.9 INTEGRITY MANAGER

MicroStrategy Integrity Manager automates the detection of inconsistencies and errors so that business users can rely 100% on the accuracy of their information. MicroStrategy Integrity Manager enables automated validation testing for data and report integrity across the Business Intelligence system. MicroStrategy Integrity Manager is a powerful tool for BI developers, administrators, and power users that can automatically test large sets of reports to uncover inconsistencies or errors before business users see them. MicroStrategy Integrity Manager compares:

- The data presented in each report with its counterpart
- The SQL generated for each report with its counterpart
- The graphing output produced by each report with its counterpart
- The Excel output produced by each report, document, or dashboard with its counterpart
- The PDF output produced by each report, document, or dashboard with its counterpart
- The Execution Times (min, max and average) for each report with its counterpart

Automated Validation Testing for Data and Report Integrity

At its simplest level, Integrity Manager executes a report (e.g. Regional Sales) in one version of Application A, generating the Data, SQL, Graph Output, Excel Export Output, PDF export output, and execution times for this report. It then executes the same report in a newer version of Application A after a change to the BI system. Finally, it compares and matches the data, the SQL output, graph output, Excel export output, PDF export output, and execution times. Any difference in these outputs is not only highlighted at the overall report level, but also in the actual results themselves. The Integrity Manager interface compares the report output results from the different versions of BI applications side by side. Differences are highlighted in red to bring them directly to the administrator's attention.

Integrity Manager automates the report comparison for each report, comparing its data, SQL, graph, Excel, and PDF output including execution times. This saves time manually generating and comparing the output. This also eliminates potential human errors when manually comparing hundreds of thousands of cells of data, or hundreds of lines of detailed complex SQL, or pages of pixel-perfect formatted enterprise documents, between two reports, documents, or dashboards.

Integrity Manager is designed to handle thousands of reports thus adding value exponentially. Data and report integrity testing that previously took weeks if not months' worth of IT effort can now be accomplished overnight.

MicroStrategy Integrity Manager Replaces Manual Report Testing

Prior to MicroStrategy Integrity Manager, BI administrators had to manually test reports if they wanted to uncover any errors in the data or the report. Manually checking 1,000s of reports can take many days and hence is rarely done. In internal tests, MicroStrategy Integrity Manager compared 1,000 reports an hour.

There are 2 dominant use cases for MicroStrategy Integrity Manager:

1. Use MicroStrategy Integrity Manager as a daily auto-detection and confirmation system to proactively verify all key management reports before users ever see them. This way, errors can be fixed before a call comes from any end user complaining about inaccuracies from their reports.
2. Use MicroStrategy Integrity Manager for regression testing every time there are explicit changes to the BI environment. Verify the impact and extent of these changes.

Integrity Manager can detect the impact of any change to the BI environment. If a certain change has occurred, with all else being equal, Integrity Manager can test the output between systems to analyze the impact of that change. Using Integrity Manager, administrators can verify success after routine and essential changes, such as:

- Data warehouse data load, ETL, processes, and triggers
- Metadata object migrations from development to test to production
- MicroStrategy software versions
- Operating systems, including moving to 64-bit operating systems
- Database platform version
- The database platform itself

Integrity Manager can compare reports across multiple environments by connecting directly to two projects. The results of the reports run from each project can be stored as historical snapshots and used as a baseline for subsequent report comparisons. This lets Integrity Manager automate report output generation from a single BI application and compare the results to a stored baseline of report output. Baseline generation and

comparison enables administrators to set up daily comparisons that ensure that no unexplained changes occur that results in unexpected report output.

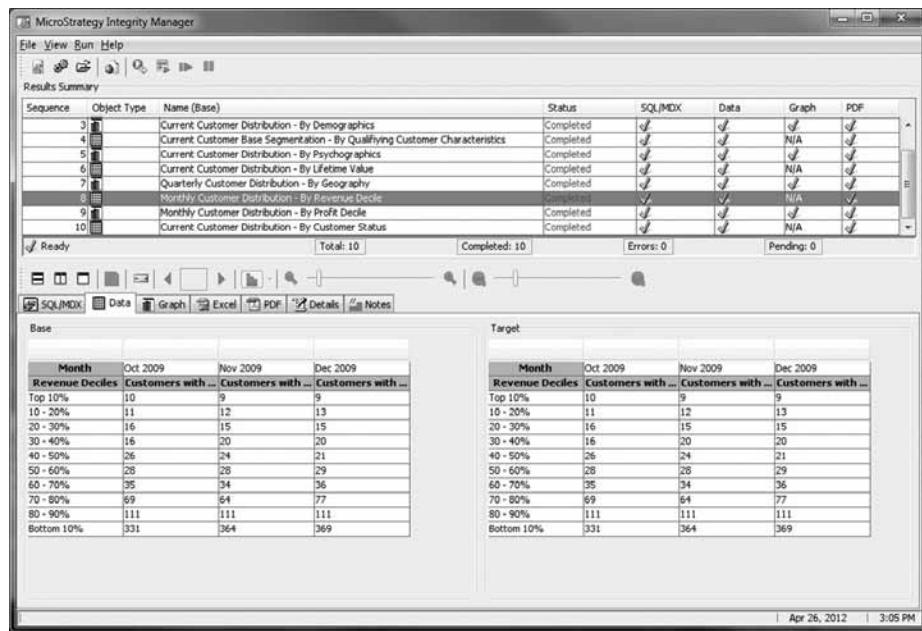


Figure 18-15 MicroStrategy Integrity Manager ensures the data integrity in the Business Intelligence System.

Every MicroStrategy Integrity Manager test generates a series of interactive HTML documents containing all test configuration selections, execution statistics, summary results, and report output details. The detailed execution results within the HTML output include links to each individual SQL statement, report output data, graph images, Excel export and PDF export letting not only administrators, but also users and developers, access test results for further analysis.

Resolving Prompts in Integrity Manager

Integrity Manager Tests can be configured to automatically resolve prompts using the prompt resolution strategies below. Users can select which of the prompt resolution strategies to use (multiple prompt resolution strategies can be used), and which order they should be applied. In Integrity Manager, the different prompt resolution strategies are:

- Personal answers
- Default object answers
- Integrity Manager user-defined prompt answers
- Integrity Manager internal answers using Integrity Manager advanced prompt resolution logic

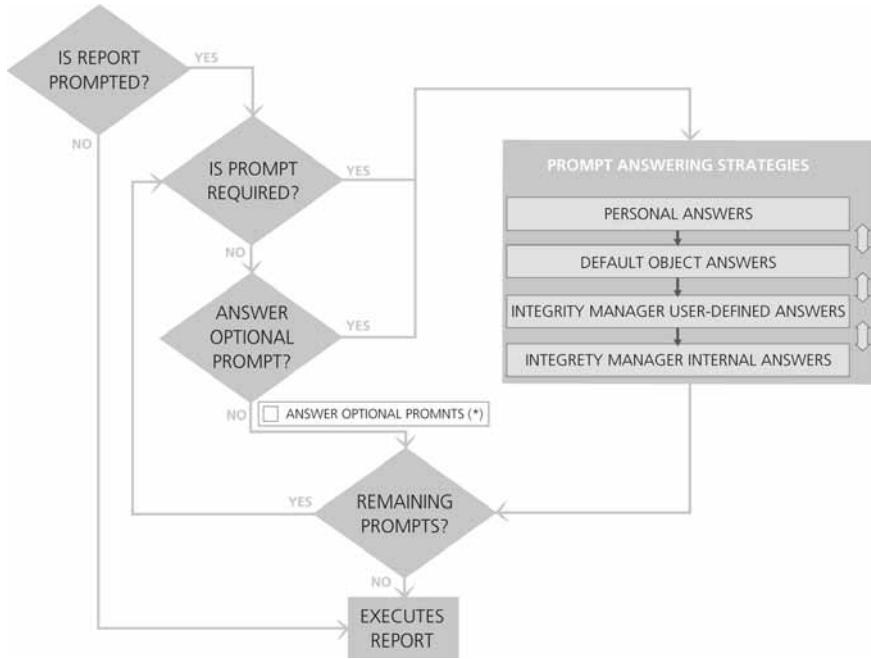


Figure 18-16 MicroStrategy Integrity Manager prompt resolution logic.

Intelligence Server Performance

Integrity Manager can determine how long it takes an Intelligence Server to execute a given set of reports or documents. Reports and documents can be executed in the integrity test multiple times, to get a better idea of the average time it takes to execute each report or document.

In a performance test, Integrity Manager records the time it takes to execute each report/document. If the reports/documents are being executed more than once, Integrity Manager records each execution time. Administrators can view the minimum, maximum, and average execution time for each report/document. In a comparative integrity test, Integrity Manager also displays the difference in time between the baseline and target reports/documents.

Report and document execution is optimized for performance. Integrity Manager uses multiple execution threads so many reports can be run simultaneously; the number of execution threads can be tuned for maximum performance. MicroStrategy Integrity Manager can be configured to run one or many tests. The tests can be initiated on-demand by the test administrator or run automatically using the command line interface. Automatic tests can run unattended and be triggered by different time-based schedules or when specific events occur, such as after a database load has completed.

18.10 MICROSTRATEGY WEB ADMINISTRATION

All BI projects require a degree of constant administration effort, which increase with added numbers of users and applications. These tasks may include setting up and maintaining user privileges, application, and data security. With MicroStrategy Web, administrators can create, modify, and manage users, groups, and security roles directly from a Web browser, allowing them to manage the BI system from anywhere, at any time. Thanks to MicroStrategy's centralized metadata, administrators need only create users and security settings once, and these settings apply to the entire MicroStrategy platform.

Intelligence Server Connection Properties

One of the tasks that administrators can perform from the Web-based Administration page is adding or removing Intelligence Servers to or from the Web Server, along with its connection properties. Such properties include: Connect Mode (Intelligence Server is manually or automatically connected when the Web Server or Intelligence Server is restarted), Port, and Load Balance Factor.

Administrators are also able to configure how users will login to MicroStrategy Web from the following options:

- Standard (user name & password)
- LDAP Authentication: (lightweight directory access protocol)
- Database Authentication
- Guest Account
- Windows Authentication
- Integrated Authentication
- Trusted Authentication Request

Web Server Security

MicroStrategy Web interface also offers a comprehensive set of security properties and settings that will allow system administrators to control the way session information is encrypted, cached, and stored on a MicroStrategy Web user's machine. System administrators can determine the following:

- Encryption of the data transferred between Web and Intelligence Server. By default, this data is not encrypted because encrypting the data increases the load on the Intelligence Server and this may result in reduced performance in MicroStrategy Web.
- Caching of any MicroStrategy Web content on the users' machines.
- Storing users' browsers cookies to hold information about MicroStrategy Web. If cookies are enabled, MicroStrategy Web stores some of MicroStrategy Web settings using browser cookies.
- Saving Intelligence Server session information in cookies. It may be necessary to store this information in cookies when a cluster is set up, but does not automatically handle session replication. Session replication is the distribution of the session information on the client instead of on the Web server so that the user can connect seamlessly to any of the Intelligence Server machines. When enabled, temporary information such as Session ID of the Intelligence Server sessions is saved.
- Enabling users to bookmark a page in MicroStrategy Web from their browsers, and whether the session information is included in the bookmark's URL. By default, a user can bookmark a MicroStrategy Web page by adding it to her browser's Favorites list.

MicroStrategy's BI platform can be set to encrypt data as it is being transmitted across the network or while it is being stored. MicroStrategy uses the following encryption protocols:

- Secure Socket Layers (SSL)
- Advanced Encryption Standard (AES)
- Tiny Encryption Algorithm (TEA)

User Management

MicroStrategy BI platform maintains a profile for each user of the BI system. Different privileges can be assigned to grant or deny to certain functionalities in any given project. MicroStrategy Web provides a graphical interface through which system administrators can perform this pivotal task.

Such fine-grained control ensures that all users access the MicroStrategy platform according to their level of proficiency. With administrators setting various levels of application functionality security, users can start using the BI applications with minimal training. Over time, administrators may grant users more privileges as they become more experienced.

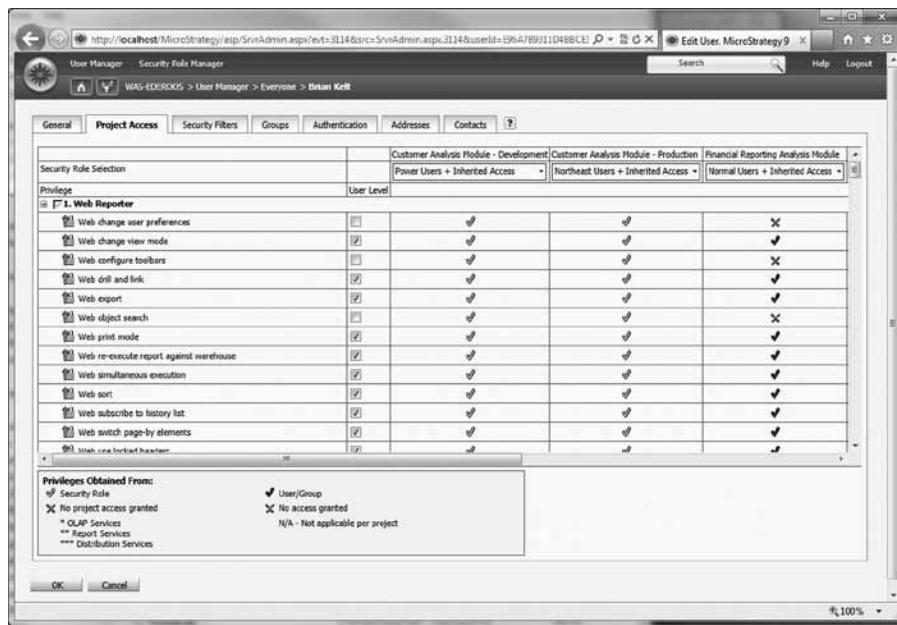


Figure 18-17 MicroStrategy employs over 220 privileges to assign application functionality to user groups, user roles, and individual users.

Security Role Manager

Another way of controlling users' access is the use of Security Roles. With this approach, all elemental building blocks of User Authorization – Application Functionality Privileges, Object Access Permissions, and Data Access Security – are first assigned to a series of user groups and security roles. Users are then assigned to any number of groups and roles, and the MicroStrategy architecture dynamically calculates privileges, permissions, and security filters to create an aggregate user profile for each user upon login. Same as with user management, there is a way for system administrators to manage security roles from MicroStrategy Web.

MicroStrategy Web Preferences

MicroStrategy Web enables its users to perform a great number of tasks, from creating grid reports, graphs reports, and dashboards to exporting and distributing them to a broader audience. Along with all these capabilities comes a wide array of settings and preferences to enhance the end-user experience. Administrators can access and modify these preferences to customize the way users work with reports, documents, grids, graphs, prompts, and more. The Project defaults in MicroStrategy Web are divided into the following categories:

- General
- Folder browsing

- Grid display
- Graph display
- History List
- Print Reports
- Export
- Print Reports (PDF)
- PDF Reports
- Drill mode
- Prompt
- Report Services
- Security
- Office

In addition to these Project defaults, individual users can customize preferences specific to them such as E-mail addresses, file locations and printer location.

18.11 MICROSTRATEGY MOBILE ADMINISTRATION

Similar to MicroStrategy Web, MicroStrategy Mobile also offers a web-based interface for system administrators to manage connection, and security settings. Additionally, system administrators will also be able to create, configure, and manage Mobile Configurations which determine how the mobile devices, MicroStrategy Mobile Server, and the Intelligence Server communicate. Thanks to MicroStrategy's centralized metadata, administrators only need to define these settings once, and they will be applied to the entire MicroStrategy platform.

Intelligence Server Connection Properties

One of the tasks that administrators can perform from the web-based Mobile Administration page is adding or removing Intelligence Servers to or from the Mobile Server, along with its connection properties. Such properties include: Connect Mode (Intelligence Server is manually or automatically connected when the Mobile Server or Intelligence Server is restarted), Port, and Load Balance Factor.

Administrators are also able to configure the trust relationship between the Mobile Server and the Intelligence Server. They are presented with 3 options for the Trusted Authentication Providers: SiteMinder, Tivoli, or create their Custom SSO setup.

Mobile Server Security

The MicroStrategy Mobile administration interface offers system administrators the option to encrypt the data transferred between the Mobile Server and the Intelligence Server. By default, this data is not encrypted because encrypting the data increases the load on the Intelligence Server and this may result in reduced performance in MicroStrategy Mobile.

MicroStrategy's BI platform can be set to encrypt data as it is being transmitted across the network or while it is being stored. MicroStrategy uses the following encryption protocols:

- Secure Socket Layers (SSL)
- Advanced Encryption Standard (AES)
- Tiny Encryption Algorithm (TEA)

Mobile Configuration

MicroStrategy Mobile delivers MicroStrategy reports and documents to Mobile devices such as iPhone, iPad, and Android phones and tablets. Users receive the same reports and documents as they would with MicroStrategy Web or any other MicroStrategy client, and can analyze them using many of the same techniques, including drilling, grouping data by page, and more.

Before a user can view MicroStrategy reports and documents on a mobile device, that device must be configured to communicate with the Intelligence Server. Manually configuring each device can be time-consuming and difficult. Instead, for MicroStrategy Mobile system administrators can create a configuration, that is, an XML file containing mobile device configuration instructions, in the Mobile Server.

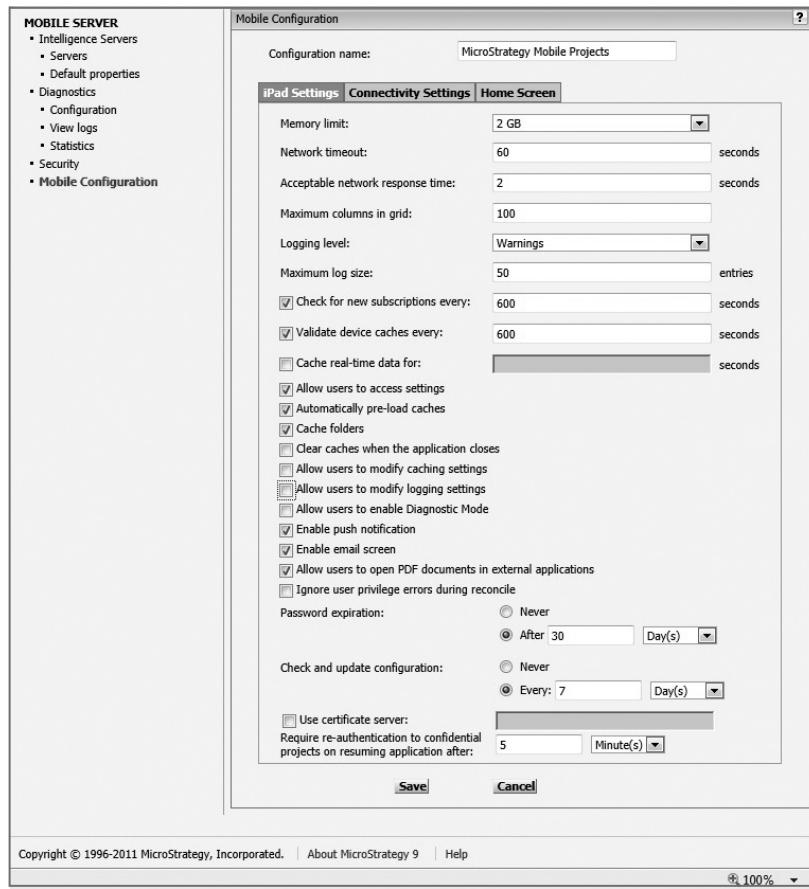


Figure 18-18 System administrators can configure how the Mobile Server and the Mobile clients communicate with Intelligence Server by creating a Mobile configuration.

For iPhone and iPad, administrators can then create a URL for a configuration and E-mail that URL to the mobile users. When a user opens the URL on her mobile device, the MicroStrategy Mobile app is automatically configured using the settings in the configuration.

For Android, administrators can create a URL for a configuration and post that URL to an HTML page. Mobile users can then use their mobile devices to browse to the web page and click on the URL. MicroStrategy Mobile app is automatically configured using the settings in the configuration.

Multiple configurations can be created and saved. Each configuration can have its own login credentials, list of projects, and home screen configuration. For example, a specific configuration can be defined for regional

sales managers, providing access to only those projects that report on sales data. A different configuration can be defined for executives, giving them access to all projects currently in production and also provides a home screen with buttons for several high-level dashboards

18.12 HEALTH CENTER

MicroStrategy Health Center is a free tool that helps System Administrators prevent, diagnose and fix problems in a MicroStrategy environment. It detects known issues and provides immediate solutions with a single click. For more complex issues, MicroStrategy Health Center collects important diagnostic information and system configuration details to help administrators resolve problems and streamline their interactions with Technical Support specialists. MicroStrategy Health Center has three main components:

1. Master Health Agent: The Master Health Agent machine serves as the hub of the Health Center system. It keeps track of the scheduled system checks for itself and the connected Health Agents, and triggers those system checks at the appropriate times. It also stores the network topology for the system, and transmits diagnostic packages to MicroStrategy Technical Support.
2. Health Agent: These machines perform system checks on themselves when scheduled to do so by the Master Health Agent, and report the results of those checks.
3. Health Center Console: It is the interface from which the system will be monitored and accessed and consists of a navigation pane on the left and a main pane on the right

Health Center uses a distributed system architecture. It consists of a central machine (the Master Health Agent) that connects to multiple additional machines (Health Agents).

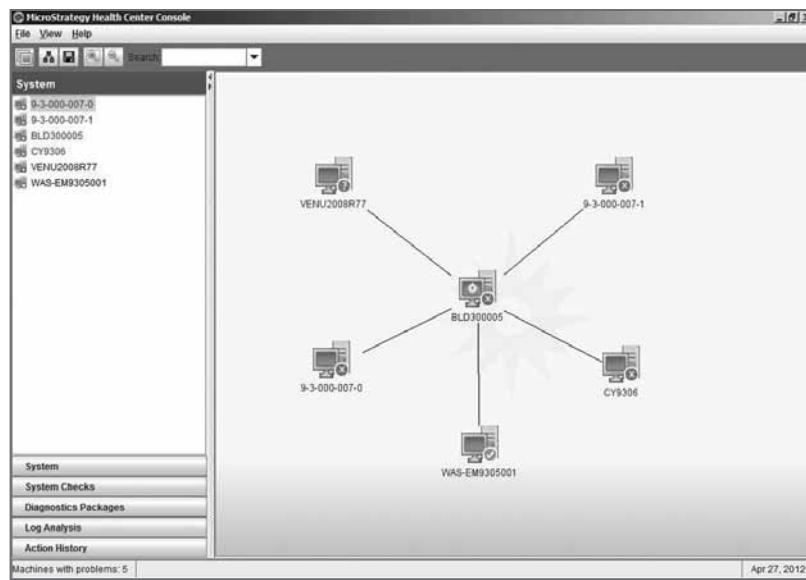


Figure 18-19 MicroStrategy Health Center provides a single interface to manage all operations across all machines.

A Health Center system represents a network of machines that are all administered by the same person or set of people. There can be multiple Health Center systems in a MicroStrategy system. For example, if the machines in the development environment all have one set of administrators, and the machines in the production environment are administered by a different group of people, the development environment and production environment should be configured as separate Health Center systems, each with their own Master Health Agent.

Diagnosing and Fixing Problems

One of the main purposes of Health Center is to diagnose any problems within the MicroStrategy system. Health Center does this by performing system checks on the Master Health Agent and on all Health Agents in the Health Center system to identify any problems. In many cases, Health Center also provides the ability to fix the problems immediately.

These system checks can be manually executed from the Health Center Console, or can be scheduled to automatically run daily or weekly, or at a specified time. The Master Health Agent handles the scheduling, and at the appropriate time, it triggers the scheduled system checks on all Health Agents in the system.



Figure 18-20 MicroStrategy Health Center helps administrator to prevent, diagnose, and fix problems.

When a system check encounters an issue raised by a Health Agent, that system check is displayed in red in the navigation pane in the System Checks panel. A list of all actions taken through the Health Center Console is displayed in the Action History panel. Other problems may not be able to be fixed from within Health Center. In these cases the Current Status tab indicates what the problem is and possible actions that can be taken to resolve the problem.

One of these possible resolutions is to contact MicroStrategy Technical Support. Typically Tech Support would ask for logs and configuration files to figure out what the problem is. This process can take several iterations until Support receives all the required information to understand the problem and recommend a solution. Health Center allows Tech Support to log into the system with the necessary permission from the system owner, and download the files they require.

18.13 SYSTEM MANAGEMENT SOFTWARE INTEGRATION

Many IT organizations implement software management applications that monitor the complete hardware and software environment. These products allow system exception handling to occur automatically, and in accordance with predetermined rules, guaranteeing predictable behavior even when a human administrator is unavailable.

Using logging databases, diagnostic files, and Windows Event Viewer, third-party system management software such as IBM® Tivoli®, CA UniCenter®, HP Openview®, and BMC® Patrol® can seamlessly integrate with the MicroStrategy BI platform. Automated, real-time tuning of the MicroStrategy's servers ensures the best possible performance and throughput. Two key MicroStrategy platform features make this possible.

1. Obtaining real-time component level information and statistics about MicroStrategy operations programmatically

- Controlling each function of the system in a programmatic fashion through Command Manager commands or directly through MicroStrategy APIs

For example, if the system management software detects that the load on the data warehouse is reaching a critical level, the number of connections opened by the Intelligence Server can be reduced automatically to alleviate the load. System management software can control Intelligence Server at a granular level, and perform the following tasks:

- Start, stop or idle Intelligence Server
- Change query governing parameters
- Change the number of open database connections
- Switch to another Intelligence Server if the original server stops responding
- Load a project that experiences peak traffic on additional nodes in a cluster
- Cancel jobs that are causing bottlenecks in the data warehouse

These and other administrative capabilities provide the ability to automate Intelligence Server configuration changes, thus increasing system availability, improving performance and report throughput, and reducing system administrator workload.

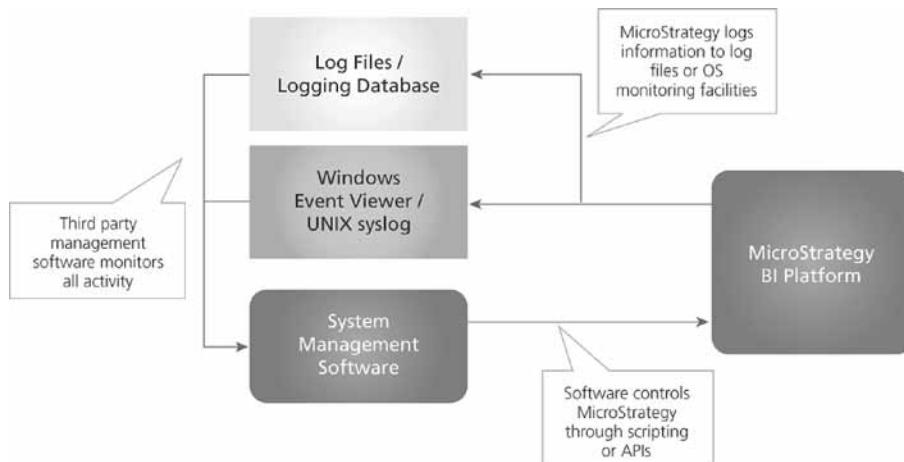


Figure 18-21 Integration with third-party system management software.

The diagram above illustrates how the MicroStrategy BI platform integrates with system management software. Intelligence Server periodically logs state information to the logging database, Windows Event Monitor, or to diagnostic files. The system management software monitors activities in these log files, identifying situations that require action. When the system management software detects one of these situations, it adjusts the configuration of the Intelligence Server accordingly using Command Manager commands or open API calls for those products. System management software can greatly automate the system administrators' activities, and is one more way that MicroStrategy reduces the total cost of ownership.

18.14 SUMMARY

The administrative capabilities of the MicroStrategy BI platform provide the most comprehensive system management environment in the business intelligence industry. MicroStrategy is the only BI platform in the industry that ensures 99.99% uptime of business intelligence applications providing access to all corporate data to all corporate users. It contains a powerful set of tools that offer a comprehensive environment for developing, deploying, monitoring, and maintaining small-, medium-, and enterprise-scale systems.



IMPLEMENTING BI APPLICATIONS

19

DEVELOPING BUSINESS INTELLIGENCE APPLICATIONS

Over the years, MicroStrategy's thousands of customers have implemented tens of thousands of BI applications. Consequently, MicroStrategy has designed the MicroStrategy BI platform to make implementation easier, and speed up deployment. Customer methodologies have ranged from loosely defined preferences to the most detailed and strictly enforced implementation policies.

Regardless of the nature of the implementation, there are common themes in the application development and implementation lifecycle. This chapter begins with a brief discussion of the common BI application lifecycle based upon the tens of thousands of BI applications built using the MicroStrategy BI platform.

Irrespective of an organization's starting point with BI – whether they have departmental solutions that expose a single style of BI or highly functional enterprise-wide applications that enable all 5 styles of BI – there is a natural evolution of successful BI applications. Evolution is driven by:

- Increasing number of users
- Increasing amounts of data
- Addition of new data sources
- Addition of new user interfaces
- Introduction of new application requirements

Only MicroStrategy provides an industrial-strength BI technology that is suited for all levels of BI deployments – from departmental, to divisional, to enterprise-wide and extranet deployments. These deployments are covered following the discussion of the application lifecycle. This chapter ends with a brief discussion of the various factors that influence capacity estimation for BI deployments.

19.1 Ad-Hoc DATA DISCOVERY

This approach to Business Intelligence allows business users to analyze data in a fast and simple way without IT intervention. MicroStrategy Visual Insight, in conjunction with Data Import, enables quick exploration of large sets of business data through an interactive and user friendly interface over the Web and mobile devices. This state-of-the-art interface empowers users to interact and manipulate data with a speed-of-thought filtering and quickly spot outliers and anomalies.

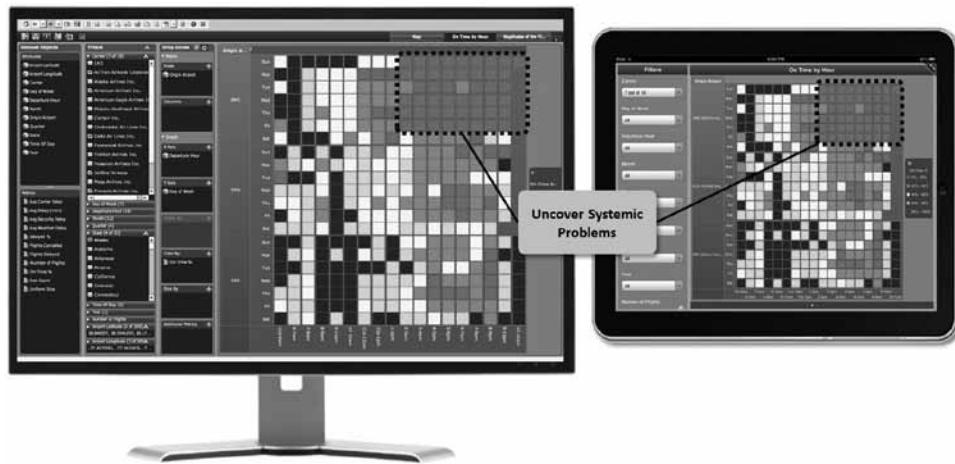


Figure 19-1 MicroStrategy Visual analyses help business users quickly uncover problem areas through web browsers and mobile devices.

Platform Components Installation

In order to begin this easy and convenient data discovery process, a set of components from the MicroStrategy platform must be installed. Once the following components are installed, business users can upload any data set they intend to interact with and begin their ad-hoc data exploration. The following products are needed to enable this functionality:

- MicroStrategy Intelligence Server
- MicroStrategy Web
- MicroStrategy Mobile¹⁵
- MicroStrategy Report Services¹⁶
- MicroStrategy OLAP Services¹⁷

After these components are installed, end-users can be added to the data discovery implementation and given access to Data Import and Visual Insight functionality. This is done through the MicroStrategy Web administration interfaces.

Ad-hoc Exploration

End users begin their ad-hoc data exploration process by importing any dataset of interest through MicroStrategy Web. The MicroStrategy data import feature supports importing data directly from files like Microsoft Office Excel spreadsheets and CSV files, relational databases, or other sources such as Salesforce.com. Users then map the imported data to MicroStrategy attributes and metrics that will be later on used in the Visual Insight analysis. Finally, the mapped data is published and saved as a shareable In-Memory cube which can be updated when new data is available.

¹⁵Optional for Mobile user

¹⁶Required for Visual Insight

¹⁷Required for Data Import

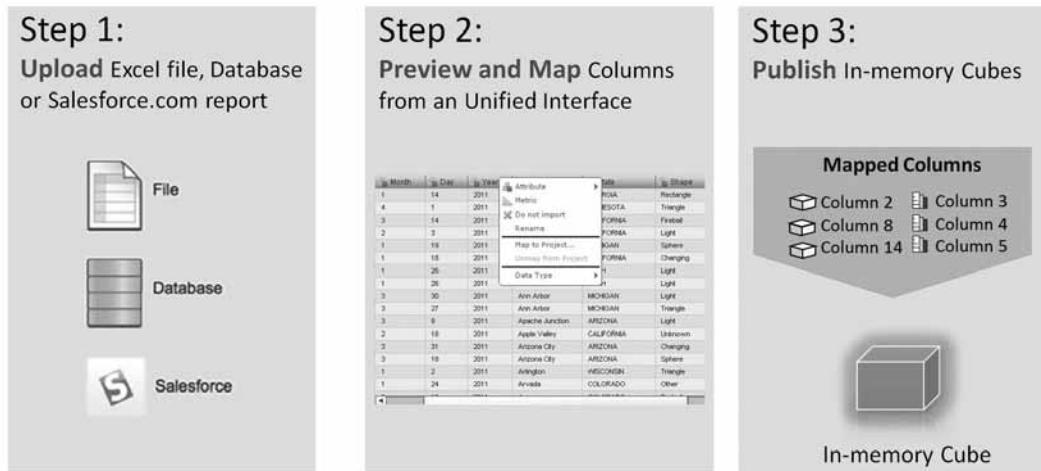


Figure 19-2 MicroStrategy Data Import provides a streamlined workflow in which business users can upload their data, map the columns to MicroStrategy reporting objects, and publish these mappings as an In-Memory Cube.

MicroStrategy Web users create Visual Insight analyses using the imported In-Memory cubes and display data in a variety of visualization types such as grid, graph, graph matrix, map, and heat map visualizations. An intuitive filter panel lets business users manipulate, rearrange, and filter data based on any business dimension and metric to get to the desired view and better understand the data. Additional calculations can quickly be added to the analysis. Once completed, Visual Insight analyses can be exported to several formats such as Excel spreadsheets, PDF files, and images. Furthermore, analyses can be saved so other users can access them through web browsers and mobile devices.

19.2 APPLICATION PROJECT LIFECYCLE

The typical BI implementation involves four key stages in the application development and implementation lifecycle:

- Plan
- Develop
- Deploy
- Maintain

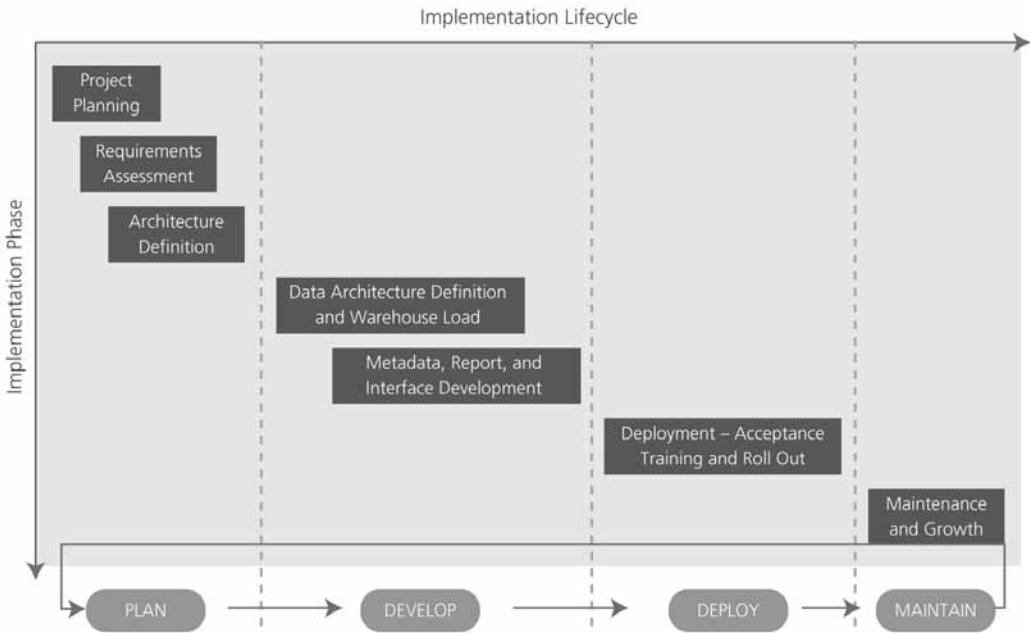


Figure 19-3 The high-level project plan of a typical BI implementation lifecycle.

Plan Stage

The Plan stage produces a comprehensive project plan and application architecture for the implementation of the BI application that serves the business needs. In general, this stage involves four types of activities:

- **Project Definition:**
Project definition begins with a conceptual vision for a solution, and defines the steps, costs, staffing, and risk mitigation strategies required to implement the conceptual vision. During this stage, project planners define the functional scope, phases of implementation, and acceptance criteria for each phase of the implementation. Project definition also includes education and training plans for developers as well as end-users of the application.
- **Requirements Gathering:**
Requirements-gathering involves identification of specific business needs served by the BI application. It identifies functionality that the future application needs to offer in the following categories:
 - Reporting, analysis, and monitoring
 - User interface(s)
 - Users and user training needs
 - Data structures, data movement, and transformation
 - Infrastructure: hardware, software, networks, and security
 - Integration with other applications
- **Architecture Definition:**
Architecture definition defines the plan for augmenting the existing technical architecture to support the new application. This requires the assessment of architectural constraints, infrastructure capacity,

system software standards, hardware standards, and development standards. Architecture definition also results in initial capacity plans for the development, testing, user acceptance, training, and production environments. Planners develop the technical architecture based upon the system integration requirements, initial capacity plans, and functional requirements of the application. This includes factors such as number of end-users, their application requirements, planned usage, and service level requirements. The typical deliverables of the Architecture definition include:

- Hardware and software architecture for development, test, user acceptance, training, production, and maintenance steps of the implementation cycle
- Security architecture
- Backup and recovery strategy
- User Interface Style guide

Develop Stage

During the Develop stage, project teams build the physical application solution scoped in the planning stage. It is composed of the following primary activity groups:

- Design:
The design step translates the highly specific business requirements obtained from the requirements gathering step into actionable specifications. The output of the design step includes detailed specifications on each category of application requirements. In some cases, organizations develop a prototype application before conducting full-scale development.
- Construct:
In this step, the development team builds all components of the BI application. This involves building the data warehouse, extract-transform-and-load routines, schema layer, application layer, and reports, scorecards and dashboards. All activities around developing custom user interfaces or extending MicroStrategy functionality occur in this step.
- Test:
The test phase involves testing the application at both the unit level as well as the system level. Unit tests ensure that individual components of the application match the specifications, while system tests ensure that all components of the BI application work as designed.

In addition, organizations conduct stress or volume testing to ensure that the system can withstand the application load expected in the production environment. The stress testing allows organizations to identify performance bottlenecks, which can be addressed using numerous tuning settings available in the MicroStrategy BI platform, and through modifications to database, ETL tools, application servers, Web servers, network, and system management tools.

Deploy Stage

The Deploy stage integrates the newly developed application into the organization. The typical steps in the Deploy stage include:

- System Acceptance:
In system acceptance testing, a set of users trained on the application goals, system usage, and success criteria approve the application for rollout to the rest of the target audience. Typically, acceptance testing occurs on a mirror of the production environment.

- Training:
One of the most crucial, but often overlooked steps in application deployments is the requirement to train the user base, including users and application administrators.
- Rollout:
During the rollout step, organizations deploy the new application to users, and shut down the legacy system.

Maintain Stage

The Maintain stage includes activities that ensure that the BI application and underlying infrastructure continue to meet business expectations. These activities include:

- Providing a responsive Help Desk to address business user questions
- Conducting regular system upgrades
- Performing periodic system and database monitoring, tuning, and administration

The Maintain stage closes the loop on application development by feeding vital information for improvements back to project sponsors, planners, and application developers.

19.3 EVOLVE FROM DEPARTMENTAL TO ENTERPRISE BI APPLICATIONS

Organizations are in various phases of adoption of Business Intelligence. Companies that are embarking on BI initiatives for the first time typically begin with departmental BI solutions. The MicroStrategy architecture's plug-and-play components make it a perfect choice for small departmental applications. A company can only use the mobile dashboarding capabilities of MicroStrategy, if all they need from the application are dashboards for executives on their smartphones and tablets. Customers who only need E-mail alerting can deploy MicroStrategy's report distribution functionality. Invariably, companies that start with narrowly focused BI applications expand those applications with new styles of BI over time. MicroStrategy's integrated architecture makes it easy to "plug in" new components while enhancing or maintaining existing ones.

This ability to future proof BI applications to scale to large user populations, or to increase analytic scope, allows application designers to roll out new functionality incrementally. No matter how organizations begin their MicroStrategy implementations, they can be confident about their ability to increase the number or functional richness of their BI applications, the number of users who access those applications or the amount of data these applications access.

Departmental Deployments

Companies often start using MicroStrategy for a single, departmental BI application. MicroStrategy's ease of use and low administrative overhead make it a perfect choice for these initial departmental applications.

The characteristics of departmental deployments include:

- A short development cycle that leverages existing data models and schema
- The provision of any style of BI, with the option of adding functionality upon user request. For example, organizations start with Enterprise Reporting and effortlessly add OLAP slice-and-dice capabilities as needed.
- Quick deployment to 10s through 100s of users through an interface of their choice –Mobile devices, Web browser, Windows Desktop, Microsoft Office suite, E-mail, printers, and file servers
- Seamless scalability to account for user and data growth

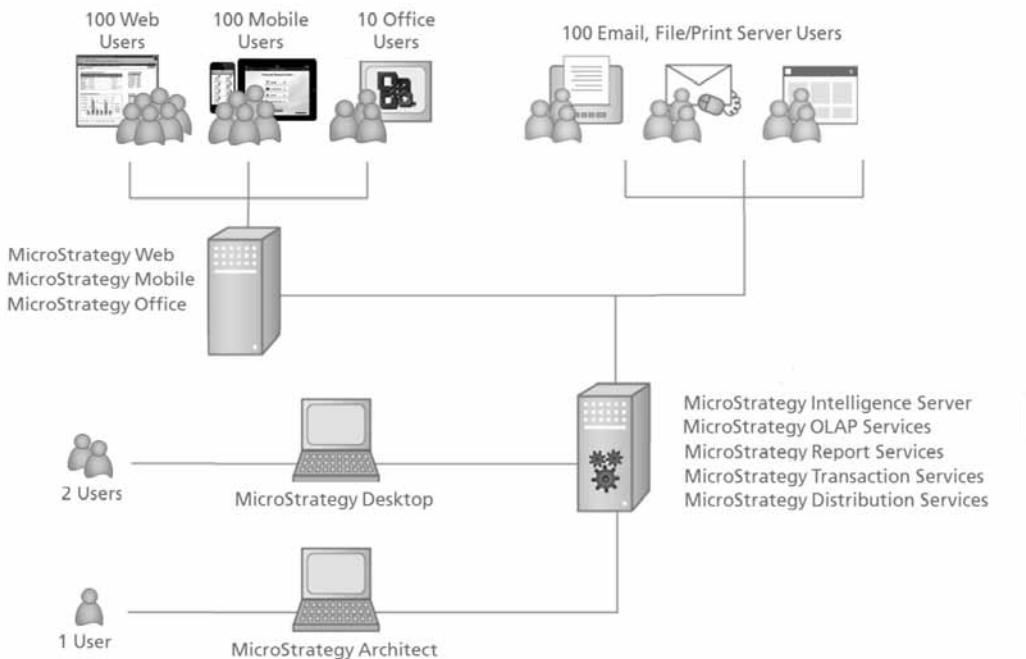


Figure 19-4 An example of a departmental deployment of the MicroStrategy BI platform.

With departmental deployments, it is common to install MicroStrategy Intelligence Server, with its plug-in components Report Services, OLAP Services, Transaction services, Distribution Services, and Multisource Option, on a single computer server. Modularity of the MicroStrategy BI platform ensures that Report Services and OLAP Services are only available if pixel-perfect reporting, dashboards, and OLAP Analysis are required by users. Separate servers host MicroStrategy Mobile and MicroStrategy Web. A handful of users manage the complete application development and deployment cycle.

Divisional Deployments

As a company's BI appetite grows, it typically demands additional styles of BI. For example, a company may have started with an operational reporting and dashboarding application, and now needs to implement advanced and predictive analytics to gain a competitive edge. Companies need support for larger user groups and data volumes, and more BI applications with the ever-increasing requirements for security.

The characteristics of a division-level BI deployment include:

- 1 to 10 Applications hosted on the same BI infrastructure
- Applications use all 5 styles of BI as needed
- Rapid deployment to thousands of business users across the whole spectrum of user interfaces supported by the MicroStrategy BI platform
- End user self-service and powerful report delivery to E-mail, printers, and file servers
- Single, robust security model for all applications and interfaces
- Comprehensive administration and change management infrastructure
- Clustered Intelligence Servers, Web Servers, and Mobile Servers for performance, fault tolerance, and reliability

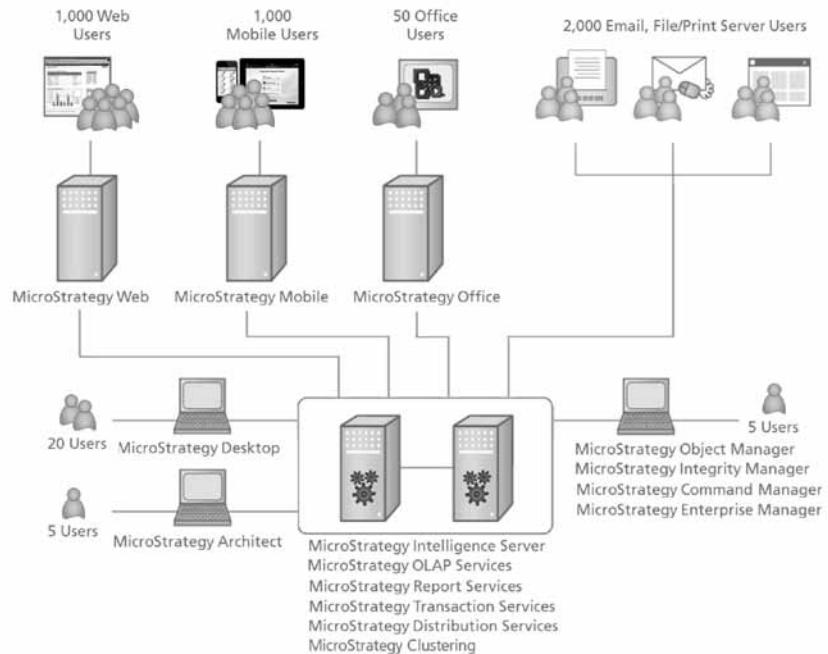


Figure 19-5 An example of a divisional deployment of the MicroStrategy BI platform.

With divisional deployments, it is common to separate the development environment from the test and production environments to improve control over the development process. The development environment includes some BI architects, and a small set of report developers and interface designers. In many cases, the BI architects and report developers are the same person. In addition, a handful of administrators manage application migration from development, to test, to production.

Enterprise-Wide Deployments

As BI becomes an enterprise utility, usable by all business users, companies can easily scale their MicroStrategy implementation by adding additional styles of BI, by increasing capacity for tens of thousands of users, and by adding infrastructure to support many applications. Since MicroStrategy's dynamic metadata objects are reusable, objects such as metrics, reports, and security definitions can be reused in new applications.

In addition to the deployment characteristics of divisional deployments, the hallmarks of enterprise-wide deployments include:

- Over 20 applications hosted on the same BI infrastructure
- Very large user communities of over tens of thousands of users
- Distributed development teams
- Enterprise caliber security for all applications and interfaces, including single sign-on authentication
- Comprehensive change management, monitoring, and administration of the BI application
- Deployment through Enterprise portals and integration with external applications using Web Services
- Effortless capacity additions through the addition of more servers that host Intelligence Server, Web, and Mobile

The MicroStrategy architecture minimizes administrative costs by automating critical but often mundane administrative tasks such as managing user privileges, purging caches, and expiring jobs. In addition to a graphical user interface or GUI-based administration, MicroStrategy gives administrators over 330 predefined scripts and procedures that automate tasks.

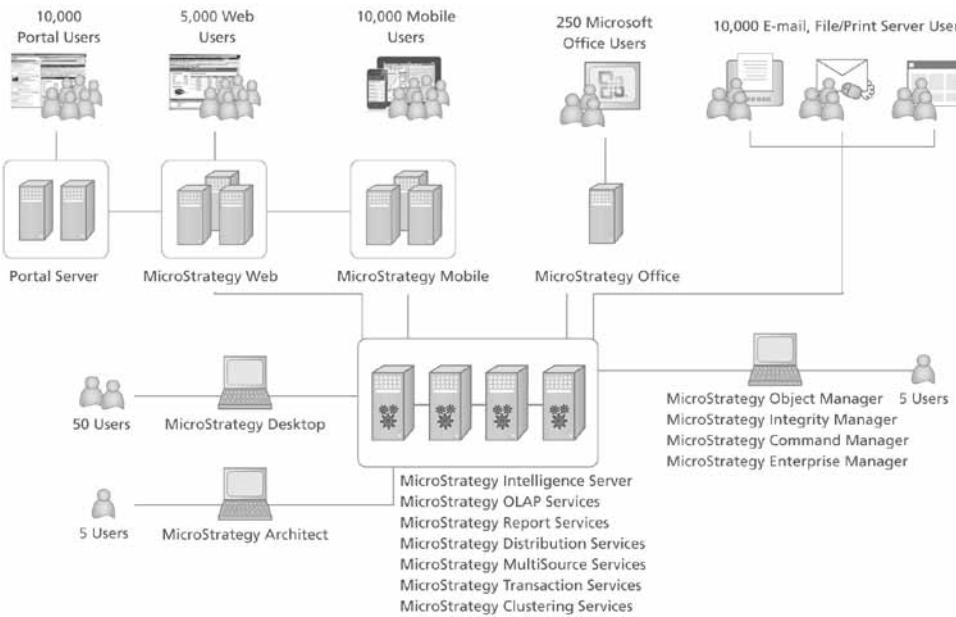


Figure 19-6 A typical configuration of MicroStrategy products for enterprise-wide deployment.

Extranet Deployments

With the success of BI applications in improving corporate performance, organizations look to implement BI applications to analyze and monitor external operations. Extending an existing application or building new applications to reach suppliers and customers places additional challenges on applications due to:

- Large distances between the business users and application infrastructure
- Increased security requirements
- Large numbers of external users
- Greater administrative overhead due to user management and enhanced system management
- Variety of technologies used by external parties

The geographic spread of users and the resulting placement of the various MicroStrategy products for optimal performance are of particular interest during extranet deployments. The remainder of this section discusses key considerations with respect to the placement of the various MicroStrategy components – data warehouse, metadata repository, Intelligence Server, Mobile, Web, application server, and client.

- Placement of Report Servers and Metadata Repository

Frequent interaction of Intelligence Server and the metadata repository mandate close proximity of the servers to the metadata. A dedicated high speed, high bandwidth network greatly increases performance due to reduced network latency.

- Placement of Intelligence Server and Data Warehouse

The same reasons cited for the relative placement of Intelligence Server and the metadata repository apply to the placement of the data warehouse. While the Intelligence Server and data warehouse can be spread out geographically, a dedicated high speed, high bandwidth network greatly increases performance.

- Placement of Intelligence Server and Web and Mobile Servers

While a high bandwidth network is desirable between Intelligence Server and the Web Servers, many performance-enhancing strategies implemented in the MicroStrategy platform, such as multi-level caching, incremental fetch, and client-side processing of user actions, reduce traffic between the Intelligence Server and Web and Mobile Servers.

- Placement of Intelligence Server and the development and administrative clients

There is no restriction for the placement of development and administrative clients. A dedicated network, however, with high bandwidth is recommended between the development/administration interfaces and Intelligence Server.

- Placement of Web Servers relative to Web users, and Mobile Servers relative to Mobile users

In general, of the four tiers of an extranet BI deployment, the network bandwidth between the Web Server and Web browser is the least controllable by the organization. For this reason, when users are highly distributed and high bandwidth networks do not exist, it is common to place Web Servers closer to remote users.

The MicroStrategy architecture provides numerous features that address BI application deployment and administration, which have been discussed in great depth throughout this book:

- Scalability for large user deployments
- Scalability for data growth
- Superior monitoring and administration of the BI system

Support for Physical and Virtual Environments

MicroStrategy can be installed on both physical and virtual environments. In a physical environment, the MicroStrategy products are installed directly on the operating system running on the machine. Generally, only one installation of a MicroStrategy product is performed, and all the hardware resources of the server are dedicated to MicroStrategy. In a virtual environment, the server's hardware resources are shared between 2 or more virtual machines that each contains a distinct "guest" operating system. MicroStrategy products can be installed in each of these virtual machines. Supported physical and virtual environments are listed in Appendix G: Operating Platforms and in the latest platform release notes.

Choosing whether to use a physical or a virtual environment is purely an implementation issue. A virtual machine is less efficient than a physical server particularly if a few resource-hungry applications are implemented in different virtual machines on the same physical server. Virtualized environments, however, have the advantage of application provisioning, maintenance, and disaster recovery over physical environments, and they also use more of the available hardware resources by sharing them across many applications.

Cloud-based environments are implementations of virtual environments. Private clouds are hosted internally by an organization, whereas public clouds are hosted by 3rd-party vendors. The MicroStrategy Cloud offers a platform-as-a-service (PaaS) to host a MicroStrategy implementation as well as data warehouse storage and extract-transform-load (ETL) services. Deploy applications to thousands of users in weeks instead of months without sacrificing anything in application power, performance, and customization on the MicroStrategy Cloud, dramatically reducing project risks and cutting operating costs by eliminating capital expenditures

19.4 ESTIMATING THE CAPACITY REQUIREMENTS OF THE BI APPLICATION

A critical step in the planning and development of a BI implementation is estimating the capacity requirements. This task approximates the initial computing resources needed to meet an application's objectives and service level agreements. Capacity estimation is achieved through either an informal or a structured approach, which fall into the following categories:

- Following generic guidelines
- Using rules of thumb
- Comparing information from similar systems and workloads
- Extrapolating from unit tests
- Extrapolating from capacity models based upon a series of unit and system tests
- Performing benchmarks that simulate the actual load

Although a more structured methodology produces more accurate results, it also uses more resources and time, and therefore comes at a much higher cost.

Managing user requests, retrieving data from data sources, adding analytical richness, formatting and distributing report results, and performing report interactions all use computing resources. Increases in the number of users, the number of BI applications, the volume of data, and the amount of analytical reporting handled by the BI servers also increases the amount of computing resources needed to maintain similar levels of performance and report throughput. Capacity estimation gives a reasonable idea of the amount of computing resources needed to power a BI implementation. The main system requirements that should be estimated are:

- Processing power

Processing power is generally the most critical resource as it is the most expensive and hard to change investment. This involves determining the number of physical computer servers, the speed of system, and number of CPU cores.

- System memory

The memory requirements of the BI application can influence the choice of the operating system and, in turn, the processing power decision. For example, any application process running on a 32-bit Windows platform can use only 2–3 GB of system memory. In contrast, 64-bit Windows, UNIX, and Linux operating systems support a theoretical 18 exabyte address space for application use.

- Hard disk space

In terms of computing resources, hard disk resources represent the cheapest of the three types of system resources and are also the least critical.

Capacity estimation should be an ongoing exercise, rather than a one-time endeavor, to ensure that the system infrastructure continues to support a successful BI application deployment. Assumptions and factors must be regularly refined with data from ongoing monitoring and tuning activities. While there are numerous factors that affect the capacity requirements of a BI application, the most important factors include:

- Number of users
- Design of reports
- Interactivity of reports
- Monitoring and administration

Capacity Estimation Factors	Affect Processors?	Affect Memory?	Affect Disk Space?
Number of Users			
Total Users			X
Active Users		X	
Concurrent Users	X	X	
Reports Design			
Intelligent Cube Size		X	X
Report Size		X	X
Analytical Complexity	X	X	
Report Layout	X	X	
Report Interactions			
Interacting with Intelligent Cubes	X	X	
Interactive Web Reporting	X	X	
Batch Reporting & Delivery	X	X	X
Exporting Report Data	X	X	
Implementation and Administration			
Metadata		X	X
Caching		X	X
Monitoring	X	X	X

Figure 19-7 Capacity estimation factors and their influence on processor, memory, and hard disk resources for Intelligence Server.

Estimating the Number of Users

The total number of users of an application is the easiest factor to estimate. Since user information is stored in the metadata repository, increasing the number of users increases the hard disk requirements. Depending on how they use the application, users have a secondary impact on the CPU and memory requirements.

At any given time, only 10-15% of all users are actively using a typical BI application. Intelligence Server allocates memory and address space for each active user from a shared pool of system resources. The larger the number of active users connected to a BI application, the more memory resources are consumed by Intelligence Server.

Typically, only 5-15% of the users logged into an application run jobs on Intelligence Server simultaneously. Thus, BI applications have a user concurrency ratio of 1-5% of the total named users of the application. Concurrent users directly influence the processing and memory resources of the computer servers.

Understanding Report Design Characteristics

The number of reports in a BI application is an important factor in determining the capacity of BI applications. However, the number of reports alone does not provide an accurate estimation; the size, analytical complexity, and report layouts must also be considered.

- Intelligent Cube Size

Intelligent cubes are stored in memory by Intelligence Server. The amount of memory required depends on the amount of data and the data types used. The size can be estimated ahead of time, and once created, the actual size of each cube is displayed.

- Report Size

Report instances are stored in a shared pool of memory called the working set. The amount of memory used depends on the amount of data and data types used, and this can be estimated relatively accurately, and the size is displayed for each cache file once they have been created.

- Analytical Complexity

MicroStrategy adds analytical richness to reports by cross-tabbing data, adding calculations, and subtotals, and managing prompts. Reports that require a large amount of analytical processing require more processing power and memory, but the exact amount can be difficult to quantify. Existing implementations with similar characteristics provide a better estimate.

- Report Layout

MicroStrategy formats reports in many ways, such as cross-tab grids, graphs, HTML, Flash, Mobile, PDF, CSV, and Excel. Applying formats to report results uses both processing power and memory. The more datasets, formatting properties, controls, and report objects contained on a report, the more processing and memory resources are consumed.

Incorporating the Effects of Report Interactions

The MicroStrategy BI platform supports all 5 styles of BI, and provides the full spectrum of user interactivity within BI applications. These interactions affect different parts of the BI implementation, and can be categorized as follows:

- Interacting with Intelligent Cubes

Intelligence Cubes accelerate BI queries by acting as the source of data for report queries bypassing the data warehouse. Report requests are processed directly in memory further enhancing performance. Intelligent Cubes support many of the features offered by MicroStrategy, such as drilling, aggregating

data, adding calculations, grouping rows of data, applying filtering criteria from static and prompted filters, and slicing report results based on user selections. They are also dynamically interrogated by ad-hoc queries and used if they have the requisite information. These interactions use both processing and memory resources on Intelligence Server.

- Interactive Web Reporting

Most MicroStrategy BI applications are deployed on a tiered Web infrastructure. This provides a very rich reporting, analysis, and monitoring environment that is centrally maintained. MicroStrategy contains a broad range of functionality, from basic activities such as accessing the system, finding, running, and printing reports, to powerful analytical features such as drilling, sorting, pivoting, adding calculations, and exporting. Sophisticated report development capabilities are also available in the platform.

The quantity and variety of report interactions consume both processing power and memory on all tiers on the Web infrastructure – Intelligence Server, MicroStrategy Web, the data sources, and the Web browser. Capacity should be estimated for all of the tiers.

- Batch Reporting and Delivery

Many organizations process commonly used reports and scheduled reports in batches. Ideally this occurs during off-peak hours and periods of low system usage. The key constraints that govern system resources for batch reporting are the quantity of reports executed and delivered, and the size of the batch window in which the processing occurs. Shorter batch windows and more reports require more processing and memory resources on Intelligence Server to complete report creation and delivery in the allotted time.

- Exporting Report Data

Report data can be exported to many different output types, such as Microsoft Excel, HTML, Flash, PDF, and CSV. The main items that govern resource consumption when exporting are the output type, amount of data, formatting of the data, frequency of export, and user concurrency.

The processing power and memory consumption on MicroStrategy Web, Mobile, and Intelligence Server increase with an increase in any of these items. Large data volumes also affect resource consumption on end user machines running Web browsers and on mobile devices.

Administering and Monitoring BI Implementations

The MicroStrategy BI platform provides the richest application metadata, with numerous monitoring and performance enhancing features to ensure successful implementations. Increasing functional richness and continued performance-monitoring take up system resources that must be included during capacity estimation.

- Metadata

The central metadata repository in the MicroStrategy platform can house many BI applications. The size of the metadata has an influence on the memory consumption of Intelligence Server. Upon startup, Intelligence Server loads the server and metadata configuration information, and the data abstraction objects for each application, into memory. Intelligence Server's memory footprint increases when loading more applications or when loading large applications.

- Caches and Intelligent Cubes

The MicroStrategy BI platform uses Intelligent Cubes and a comprehensive, multi-level caching architecture to improve throughput, optimize query performance, and reduce end user response times. Cubes and caches are stored both in memory and on disk. They affect not only the memory footprint of the Intelligence Server, but also the allotted hard disk space required to store cache files off-loaded to disk. The cache monitor in Desktop provides accurate information about the number and size of Intelligent Cubes and caches in existing implementations.

In addition to shared caches, personal reports can be saved to individual users' history lists. These reports increase the amount of hard disk capacity and memory needed on Intelligence Server. The History List storage can also be loaded in a relational database.

- Monitoring

MicroStrategy logs statistics, errors, and counters that monitor processor resources, memory usage, system throughput, and user activities. Storing the usage and performance statistics consumes a negligible amount of processor and memory resources, but additional hard disk space is required to store the log files. The capacity estimation exercise must account for performance monitoring and usage statistics logging. It is important to note that these monitoring resources should be used to improve future capacity estimation and tuning activities.

19.5 SUMMARY

The MicroStrategy BI platform provides all the ingredients of an industrial-strength BI technology ideally suited for all levels of BI deployments – from ad hoc, to departmental, to divisional, to enterprise-wide and extranet deployments. Irrespective of where organizations begin with their BI implementations – whether they have departmental solutions that expose a single style of BI or highly functional enterprise-wide applications that enable all 5 styles of BI – MicroStrategy deployments grow with the organization to meet the evolving requirements of:

- Increasing number of users
- Increasing amounts of data
- Addition of new data sources
- Introduction of new application requirements
- Deploying on a physical server, a virtual environment, or in the cloud

With thousands of customers and tens of thousands of successful BI applications under its belt, MicroStrategy has designed the MicroStrategy BI platform to make implementation easier, and speed up deployment. Customers worldwide ensure their BI success and mitigate deployment risk by relying on a team of experienced MicroStrategy Professional Services experts. The Services team leverages decades of cumulative industry, process, and technology-specific knowledge, and provide best practice recommendations to ensure the success of BI applications.

20

PORTABLE ANALYTIC MODULES

Every day, companies around the world in all major industries rely on MicroStrategy to give their entire enterprises a leading edge in monitoring all aspects of their business. Executives, managers, and business analysts yield tremendous benefits from the ability to report, analyze, and monitor business processes with BI applications that measure the performance of the corporation as a whole, departments within the corporation, individual business processes deployed, and customer and supplier interactions.

With over 20 years of experience in business intelligence, MicroStrategy has developed a suite of starter kits to help businesses get started with developing analytical applications. Like packaged applications, the starter kits enable rapid deployment of BI applications. Like custom applications, the starter kits allow companies to reuse their data warehouse, and adapt easily to changing business requirements.

20.1 RAPID APPLICATION DEVELOPMENT FRAMEWORK

The starter kits, or analytic modules, are bundled with MicroStrategy Architect and contain the following components:

1. Analytic Modules
 - i. Prepackaged metadata – complete metadata repository of best-practices reports, dashboards, key performance indicators, attributes, metrics, filters, and custom groups
 - ii. Default physical and logical data model – analytics that work off of a logical data model that can easily be mapped to an existing data warehouse or packaged data warehouse schema
 - iii. Reference Guides – documentation on the data model, analysis areas, metadata object definitions, data dictionary, and report usage scenarios
 - iv. The analytic modules span the following key business areas:
 - Customer Analysis
 - Financial Reporting Analysis
 - Human Resources Analysis
 - Sales And Distribution Analysis
 - Sales Force Analysis
2. Implementation Methodology
 - i. Implementation and Portability Guide – step-by-step guide for implementing analytic modules against existing data warehouses
 - ii. Best Practices for Building Analytic Modules – design rules for designing and developing analytical applications

Each analytic module contains a set of reports, key performance indicators (KPIs) and business metrics, and sample scorecards. Developers and users use these reports and key performance indicators as building blocks to create and deploy additional reports and performance measurements that are specific to their organizational needs.

MicroStrategy provides report-building wizards to design and build reports. Developers also have at their disposal over 270 functions and operators (arithmetic, aggregate, statistical, financial, mathematical, OLAP, and data mining) with which to build any new KPIs and business metrics.

For example, a developer can use the revenue metric included in the customer analytics module to build additional metrics such as the maximum, minimum, running sum, moving average, standard deviation, future value, and seasonal forecasts of the revenue. Other areas of analysis and navigation paths can be customized, and new workflow wizards let end users build their own reports to analyze information. Customization that traditionally took months can now be achieved in days.

20.2 PORTABLE ANALYTIC APPLICATIONS

Packaged analytic applications have had limited success in the marketplace because they are difficult to implement and customize, and they require large investments of time and resources. A better approach, known as portability, is to build packaged reporting modules that are independent of the physical data warehouse schema using an abstraction layer that maps the logical business structures to the physical database.

MicroStrategy analytic applications offer the benefits of packaged applications without the constricting requirement of conforming to a vendor-provided data-model and schema, or the need to modify existing warehouse structures to work with vendor-provided reports and analyses. Businesses can now leverage their existing data warehouses without the need for additional data extraction, transformation, and loading processes.

Developing a portable analytic application is different from developing a typical business intelligence application. The portability paradigm is governed by a set of rules and premises that should be adhered to when designing and building these applications. These rules span the three main development phases:

1. Analytic Domain Definition

Adherence to standard analytic domains ensures the application applies to a wide number of warehouse schemas, and that it leverages standard processes and best practices for the analytic domain. Portable applications for analytic domains should represent one of the following:

- Standard business process – e.g. sales force automation, human resource management
- Business problem area – e.g. customer retention, product affinity
- Functional area – e.g. marketing, finance
- Standard data source – e.g. Web logs, sales and distribution data

2. Multi-dimensional Model Design

The multi-dimensional model is a key component of the portable applications architecture. MicroStrategy portable analytic modules use an abstraction layer between packaged reports and the underlying data warehouse. It is imperative that the multi-dimensional data model complies with the following rules:

- Modular Architecture – minimizes the impact of missing data attributes from the data warehouse, and makes implementation methodical
- Standard Analytic Modules – focuses on horizontal functional areas, and makes the applications applicable in many vertical industries
- Simple Hierarchical Structures – reduces the dependence on the underlying physical schema and complex parent-child relationships

- Physical Schema Independence – ensures applications are not restricted by underlying physical structures, and works with all data warehouse schemas

3. Report organization and development

This important step involves creating the final package, including the documentation of the reporting components. Report design and development in portable analytic applications should be conducted according to the following rules:

- Organized Report Library – reports grouped by distinct analysis areas, dictated by key reports and metrics
- Non-Database Specific Analysis – defines the business logic in the abstraction layer, and allows portability across database platforms
- Schema-Independent Abstraction – uses underlying definitions to build key performance indicators and reports that ensure changes to definitions permeate seamlessly throughout the application
- Package Standard Reports – focuses on standard best practices reports to provide a starter kit for analytic applications that can easily be implemented and extended

The portable analytic applications are built on a complete platform architecture that makes customizations, extensions, and deployment an easy and manageable task.

20.3 MICROSTRATEGY ANALYTIC MODULES

The Analytic Modules are a set of packaged analytic components, including reports, scorecards, KPIs, and business metrics, which are intended as starter kits to begin developing analytic applications specific to each organization. The modules are designed on top of a pre-defined data warehouse schema but can be mapped directly to existing data warehouses. The packaged best practices in terms of reports and metrics reduce application development time significantly, making this a low-risk, high return investment.

Customer Analysis

The Customer Analysis Module includes over 40 reports and 65 key performance indicators and business metrics for marketing and customer relationship management. The components help segment customers, monitor lifetime customer value, improve customer satisfaction, and drive profitability. The Customer Analysis Module's best practice analytics are fully documented in a 160-plus page reference guide.

1. Key Areas of Analysis

- Acquisition, Retention and Attrition
- Customer Segmentation
- Profitability and Cross-Sell Analysis

2. Key Questions Answered

- Has the rate of attrition, retention, and acquisition changed over time?
- What is the impact of tenure on customer retention and attrition?
- Are valuable customers more likely to leave?
- What is the revenue loss attributable to customer attrition?
- What is the distribution of customers by demographic and psychographic characteristics, and by lifetime value and revenue contribution?
- How have demographic and psychographic characteristics changed over time?

- What percent of revenues are being contributed by a given segment of customers?
- What are the revenue, profit and margin contributions of customers by customer profile?
- What are the characteristics of the most profitable customers, and what products do they buy?
- Which products have increasing sales momentum?
- Which products sell well together?

3. Key Attributes

- Customer Geography, Customer Demographics, Lifetime Value Score, Customer Psychographics, Time, Revenue Deciles, Profit Deciles, Product, Customer, Affinity Product, Customer Status

4. Key Performance Indicators and Business Metrics

- Active Customers, Lost Customers, New Customers, Retained Customers, Attrition Rate, Retention Rate, Attachment Rate, Revenue, Revenue Per Customer, Profit, Margin, Profit Variance, Revenue Variance

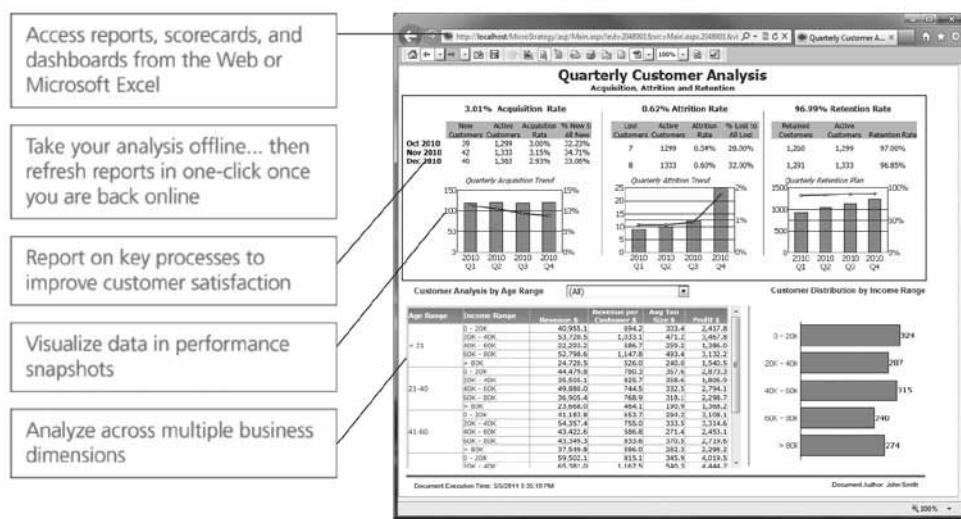


Figure 20-1 Customer-centric organizations benefit from reporting, analysis, and monitoring with MicroStrategy, yielding higher customer loyalty and satisfaction. As shown above, business users may opt to run MicroStrategy reports from Microsoft Office Excel, PowerPoint, Word, or Outlook.

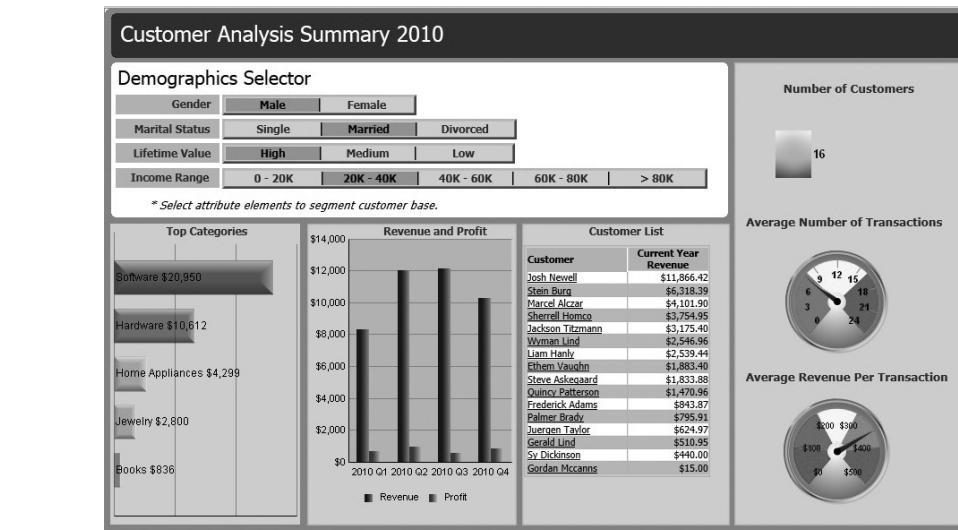


Figure 20-2 An example of a Customer Analysis Scorecard.

Financial Analysis

The Financial Analysis Module includes over 40 reports and 50 key performance indicators and business metrics used to obtain in-depth knowledge of financial information to increase corporate transparency and ensure Sarbanes-Oxley compliance. Its best practice analytics are fully documented in a 200-plus page reference guide.

1. Key Areas of Analysis

- Accounts Payable
- Accounts Receivable
- Balance Sheet
- Cash Flow
- Costs and Expenses
- Revenue and Forecasts
- Profit and Loss Statements
- Planning and Forecasting

2. Key Questions Answered

- What is the breakdown of costs by vendors, and what are the associated trends?
- Which bills are due this week, and for what amounts?
- What is the aging distribution of Accounts Payable and Accounts Receivable?
- Are there any customers with payment problems? If so, who needs to be notified?
- What is the value of assets, liabilities and owners' equity on a given date?
- What is the change in cash position period over period?
- What are the revenue trends for the business units?
- What are the trends in revenue-by-revenue types, if applicable?
- What is the breakdown of expenses by business unit?
- What is the forecasted revenue? Has this forecast changed, and what caused the change?
- What is the Actual Amount profit margin by business unit or region? What are the associated trends?
- Which business units are hitting their targets?

3. Key Attributes

- Account Type, Account Class, Vendor, Customer, Invoice, Time, Employee, Business Unit, Corporation

4. Key Performance Indicators and Business Metrics

- Accounts Receivable Aging, Accounts Payable Aging, Actual Amount, Forecast Amount, Planned Amount, Variance, Open Amount, Operating Income, Write-off Amount, Net Income

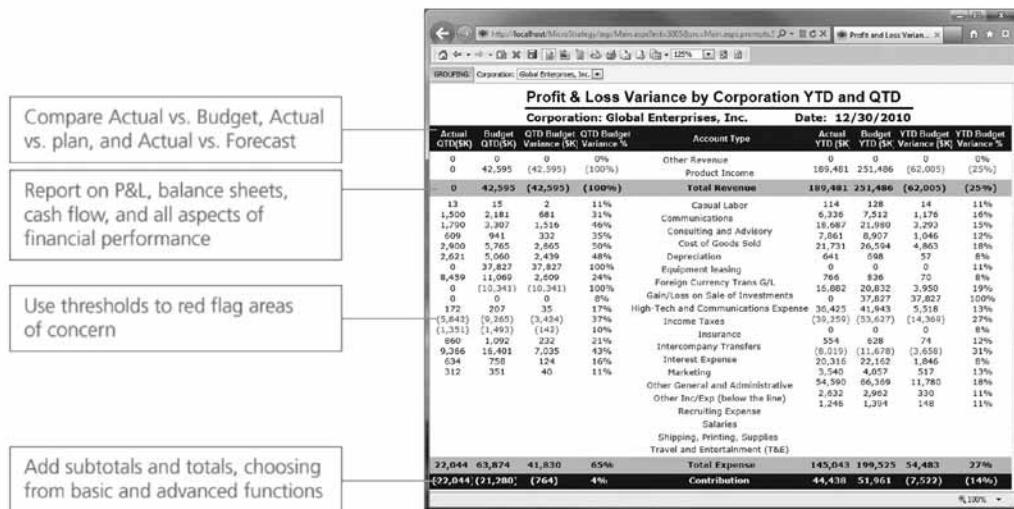


Figure 20-3 Financial reports in MicroStrategy meet exact printing and executive specifications, and allow for full investigative analysis.

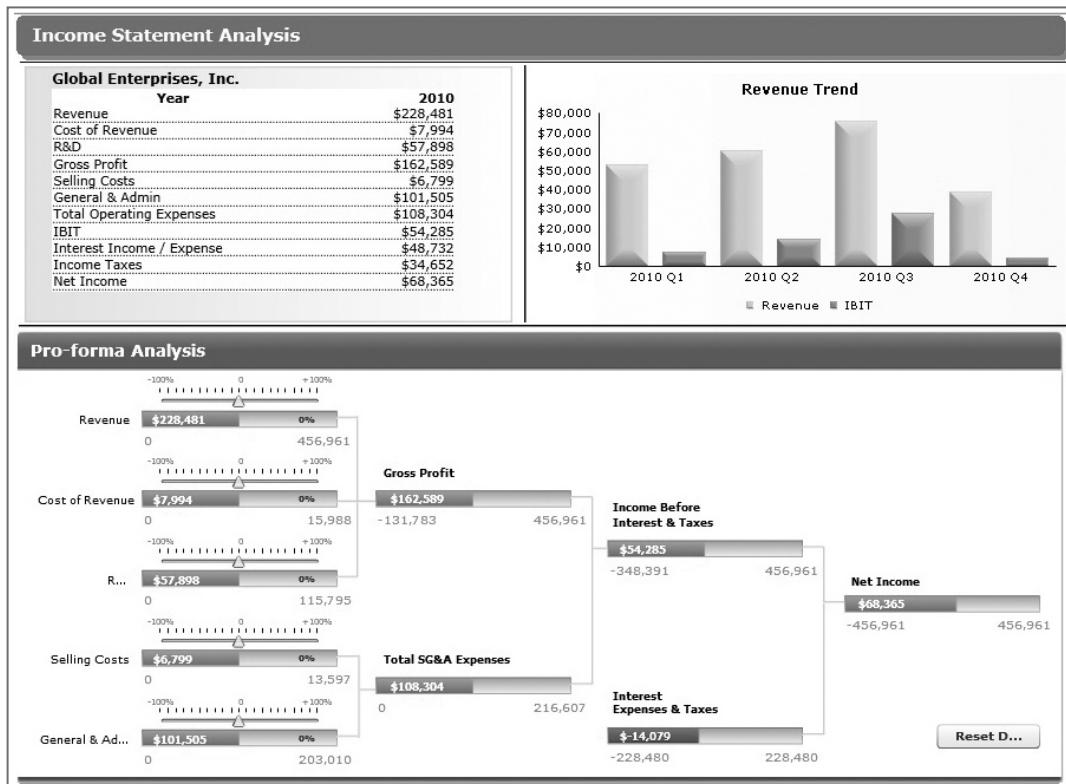


Figure 20-4 Sample Financial Analytics Report with What-If functionality.

Human Resource Analysis

The Human Resources Analysis Module includes over 45 reports and 80 key performance indicators and business metrics designed to help your HR professionals monitor employment and labor law compliance, optimize the tremendous fixed costs associated with human resources, and improve employee recruitment, retention, and performance. Its best practice analytics are fully documented in a 250-plus page reference guide.

1. Key Areas of Analysis
 - Workforce Analysis
 - Compensation Analysis
 - Benefit Analysis
 - Attrition Analysis
 - Recruitment Analysis
 - Workforce Development and Productivity

2. Key Questions Answered
 - How effective is the recruitment process?
 - Are human resources utilized productively?
 - What are the current and future staffing needs?
 - How can the best people be attracted, retained and motivated?
 - How can individual goals be aligned with corporate strategy?
 - Are the compensation and benefits correctly related to the title and job description?
 - How does compensation relate to market standards?
 - How is attrition managed?
 - What sources do employees have to plan their retirements?
 - Are performance appraisals used efficiently?

3. Key Attributes
 - Employee, Clearance Status, Work Experience, Demographics, Organization, Salary Level, Position Status, Benefit Type, Satisfaction Code, Recruiting Source, Job Start, End Date, Performance Score

4. Key Performance Indicators and Business Metrics
 - Headcount, Salary, Benefit, Attrition, Acquisition, Recruiting, Qualifications, Performance, Average Salary, Vacation Days, Sick Days, Bonus Potential, Bonus Received

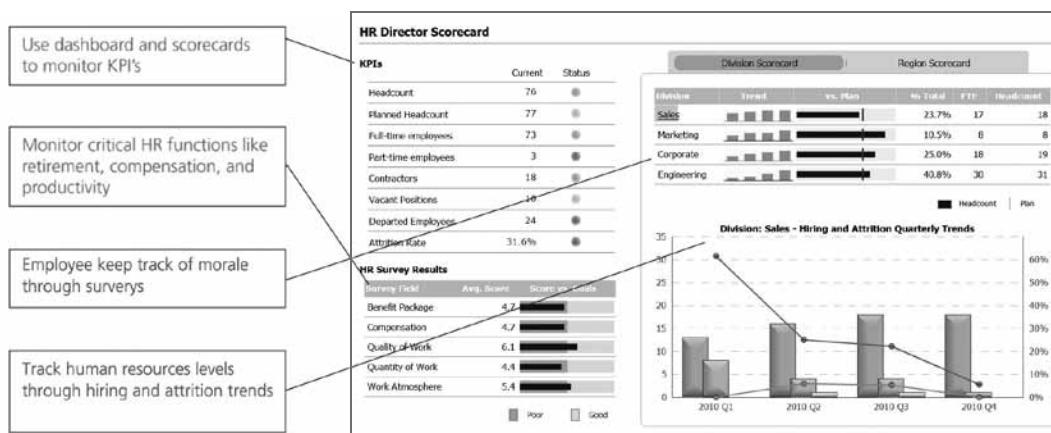


Figure 20-5 MicroStrategy's powerful graphing and visualization capabilities provide HR specialists with the analysis they desire in an intuitive and easy-to-understand format.

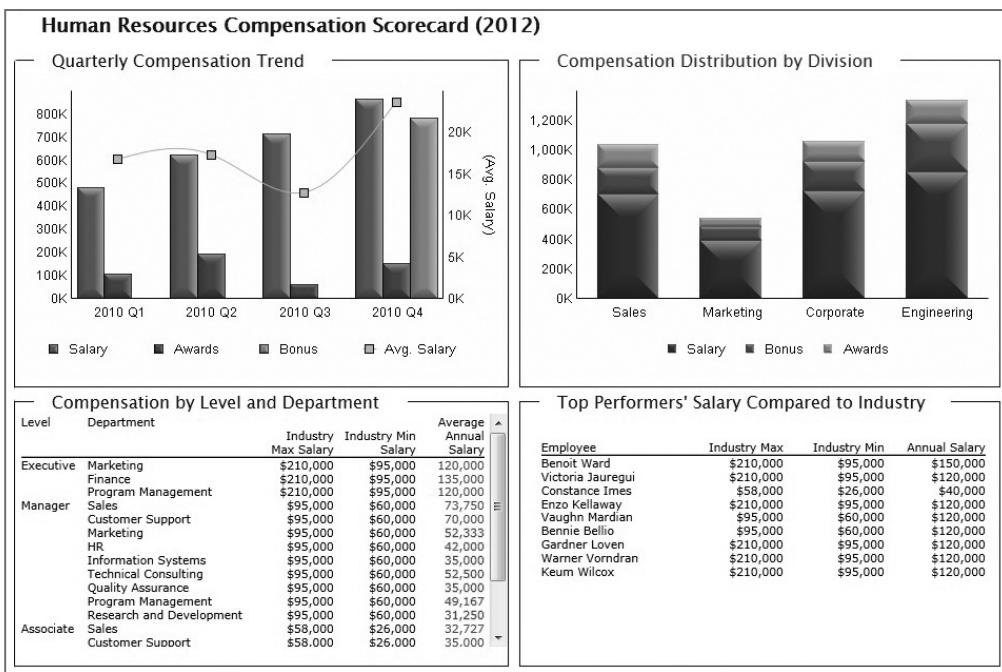


Figure 20-6 Human Resources Analytics Chart.

Sales Force Analysis

The Sales Force Analysis Module includes over 40 reports and 85 key performance indicators and business metrics to help organizations convert leads into sales, monitor sales pipelines, and rank sales executives. The Sales Force Analysis Module's best practice analytics are fully documented in a 200-plus page reference guide.

1. Key Areas of Analysis

- Lead Type and Quality Analysis
- Sales Pipeline Analysis
- Product Sales Analysis
- Sales Performance Analysis

2. Key Questions Answered

- What are the sources of my leads and what are the conversion rates for these leads?
- Are there any trends in leads and leads size?
- How long does it take to qualify a lead?
- What is the profile of prospects and customers?
- What are my expected revenues at any point in time and how does it compare to forecasts?
- What is the competitor activity in the various accounts and is there a trend?
- What is the most important reason for lost deals?
- What are my top selling products and who is buying them?
- What is the contribution to sales of my top products?
- Which products have increasing and declining sales momentum?
- Which products are sold together most often?
- How are sales representatives performing? Are they on track to meet their sales targets?

- Which sales regions performed the best and worst? Are there clear trends?
 - Who are the best performers in terms of percentage of opportunities closed and overall revenue?
3. Key Attributes
- Prospect, Company, Lead Source, Lead, Lead Type, Sales Organization, Opportunity, Order, Time, Competitor, Product, Order, Account or Customer
4. Key Performance Indicators and Business Metrics
- Lead Conversion Rate, Wins, Lost Deals, Win Percentage, Quota, % Quota Achieved, Opportunities, Pipeline Revenue, Orders, Opportunities, Pipeline Revenue, Target, % of Target.

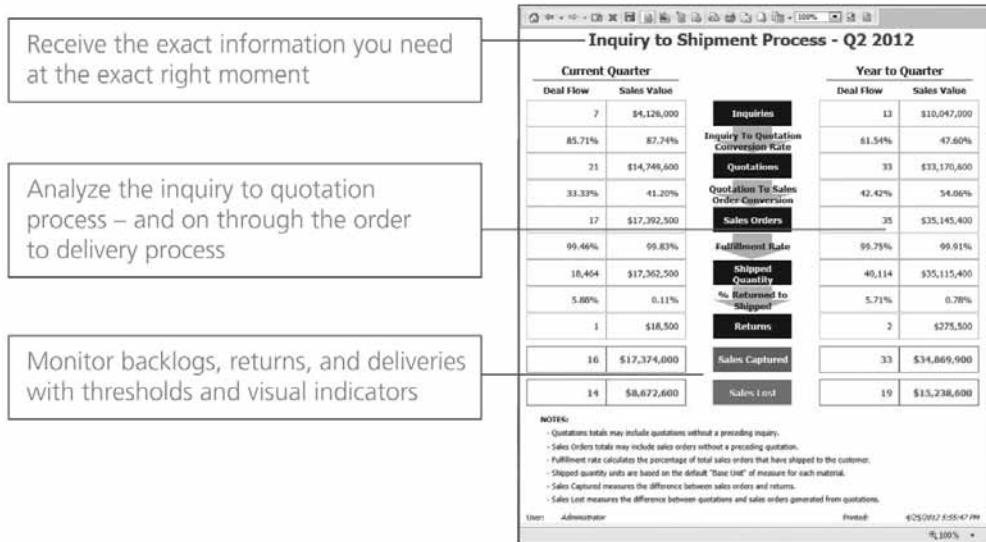


Figure 20-7 Process flow dashboards with MicroStrategy give at-a-glance convenience to reviewing mission critical information. Business users can receive this information directly in their E-mail inbox.

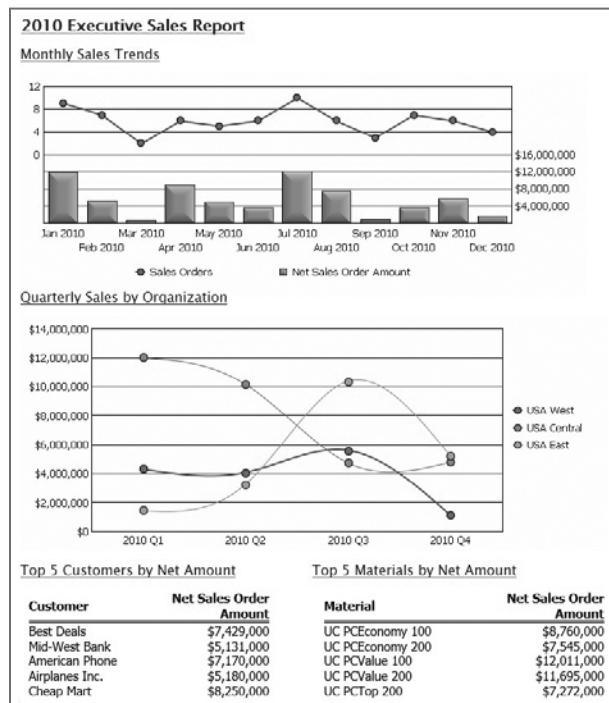


Figure 20-8 Sample Sales and Distribution Analytics Chart.

Sales Force Analysis

The Sales Force Analysis Module includes over 40 reports and 85 key performance indicators and business metrics to help organizations convert leads into sales, monitor sales pipelines, and rank sales executives. The Sales Force Analysis Module's best practice analytics are fully documented in a 200-plus page reference guide.

1. Key Areas of Analysis

- Lead Type and Quality Analysis
- Sales Pipeline Analysis
- Product Sales Analysis
- Sales Performance Analysis

2. Key Questions Answered

- What are the sources of my leads and what are the conversion rates for these leads?
- Are there any trends in leads and lead size?
- How long does it take to qualify a lead?
- What is the profile of prospects and customers?
- What are my expected revenues at any point in time and how does it compare to forecasts?
- What is the competitor activity in the various accounts and is there a trend?
- What is the most important reason for lost deals?
- What are my top selling products and who is buying them?
- What is the contribution to sales of my top products?
- Which products have increasing and declining sales momentum?
- Which products are sold together most often?
- How are sales representatives performing? Are they on track to meet their sales targets?
- Which sales regions performed the best and worst? Are there clear trends?

3. Key Attributes

- Prospect, Company, Lead Source, Lead, Lead Type, Sales Organization, Opportunity, Order, Time, Competitor, Product, Order, Account or Customer

4. Key Performance Indicators and Business Metrics

- Lead Conversion Rate, Wins, Lost Deals, Win Percentage, Quota, % Quota Achieved, Opportunities, Pipeline Revenue, Orders, Opportunities, Pipeline Revenue, Target, % of Target.

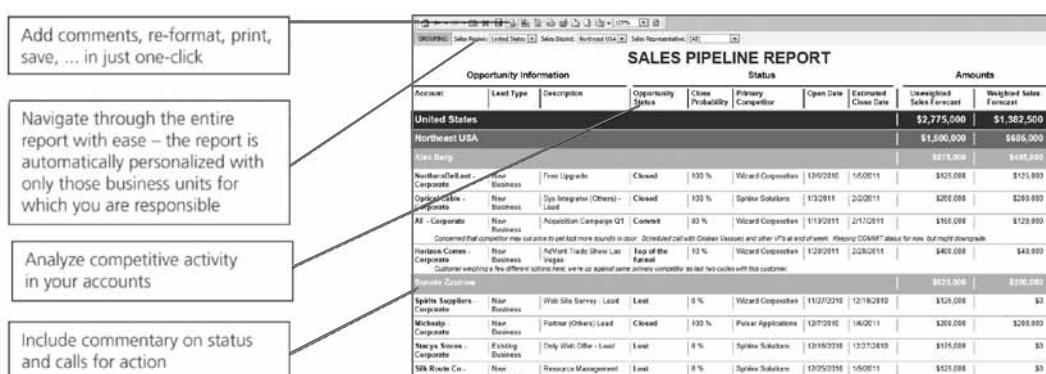


Figure 20-9 Sales performance reports and scorecards in MicroStrategy enable executives to identify top performers as well as areas of concern.

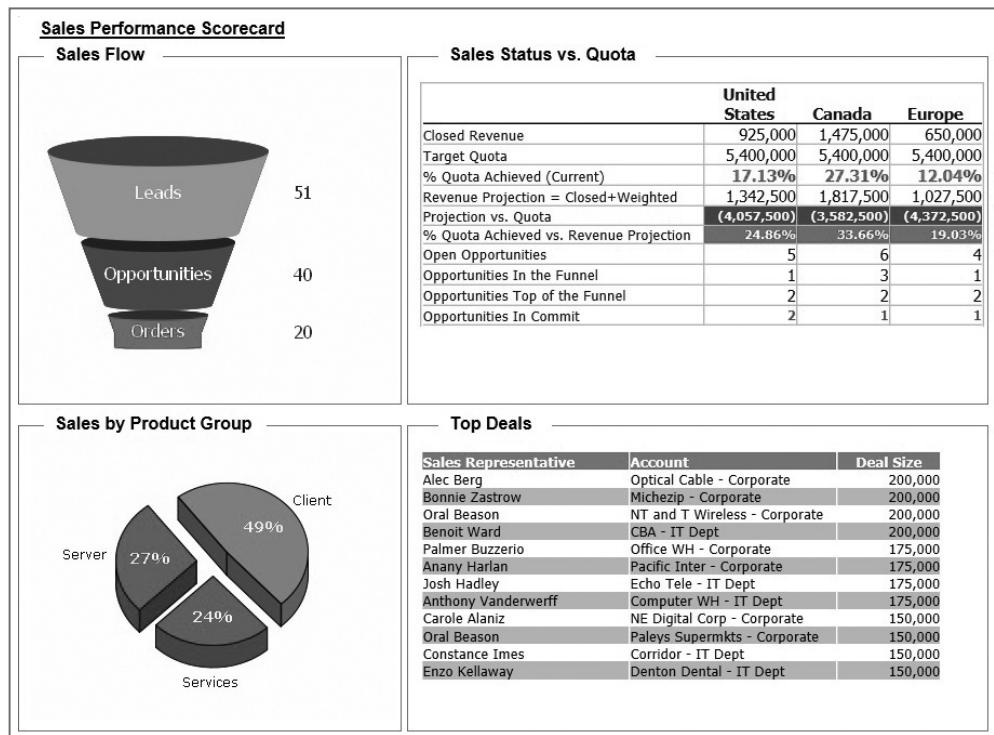


Figure 20-10 Sample Sales Force Analytics Scorecard.

20.4 SUMMARY

Each analytic application module is designed as a starter kit complete with scorecards, dashboards, reports, metrics, and KPIs. Business users access these analytics easily from: any Mobile device, any Web browser, E-mail program, or Microsoft Excel, PowerPoint, and Word.

Each module includes an application model (logical schema), a data warehouse model (physical schema), and extensive documentation. MicroStrategy Desktop Designer enables full extensibility and customizability. MicroStrategy Architect maps the logical schema to any physical schema. Customers can take full advantage of their existing data warehouse and ERP investments, and are able to develop applications rapidly.

MicroStrategy's application modules are built on the MicroStrategy platform, taking full advantage of all 5 styles of business intelligence, data scalability, user scalability, and centralized IT infrastructure. The analytic modules are unique because they combine the flexibility of building custom applications with the speed of deploying packaged applications, at a price well below the six-figure BI applications available in the market from other software companies.

21

EXTENSIBILITY AND THE SDK

A key component of an enterprise-class software platform is the ability to modify and extend existing functionality to meet the unique needs of an organization. For building customized applications, a robust BI platform must provide access to all of its functionality through application programming interfaces (APIs) that is available from a variety of programming languages. A thoroughly documented API will enable easy customization and extension of application functionality. Additionally, common BI features should also be available in a service format that is accessible from many different, external applications. Finally, a platform that serves as the basis for application development should include sample code and utilities that give guidance for modifying and extending the platform.

The following components in the MicroStrategy BI platform make customizations and extensions to any BI application possible and easy to maintain:

- A Service-Oriented Architecture
- Open APIs
- MicroStrategy platform as a BI service provider
- A plug-in architecture
- Documentation enabling access to the API and platform functionality
- Development kits, utilities and sample code

21.1 ACCESS ALL BI FUNCTIONALITY IN A SERVICE-ORIENTED ARCHITECTURE

The MicroStrategy BI platform is a modern, sophisticated service-oriented architecture designed for flexibility and extensibility so that it can grow with the varied demands of enterprise applications. Due to the openness of the MicroStrategy BI platform, external applications can instantly take advantage of BI functionality, performance, failover, load balancing, and end-user interfaces by integrating or plugging directly into the platform. The ability to plug into the MicroStrategy BI platform enables extensive user experience customizations and product innovations that help organizations make the most effective use of information to drive business performance.

MicroStrategy's service-oriented architecture guarantees efficiency for each service or component with no code redundancy. This architecture is in stark contrast to the service-oriented architectures of other BI vendors, which are a product of disparate, separately developed architectures, inherited from mergers, and mashed together. Unlike SOA's of other BI vendors that wrap code to hide redundant functionality and inconsistency across components from acquired technologies, MicroStrategy's organically developed architecture with feature-level granularity enables application developers to design lean customizations, those that do not invoke any redundant code and use those APIs that are needed.

MicroStrategy architecture exposes consistent BI functionality using thousands of documented interfaces and methods and encapsulated as Open APIs. External applications can invoke MicroStrategy Web functionality

as a service through the architecture's Task Framework. Additionally, MicroStrategy provides a plug-in architecture that enables application developers to extend out-of-the-box MicroStrategy Web code with stand-alone customizations without the need to change the source code, making upgrades to new versions without recoding possible.

Open APIs

An API provides a framework to access data and functionality from outside the MicroStrategy platform. The platform includes the following groups of API's:

- MicroStrategy Web API
- MicroStrategy Mobile API
- Visualization API
- MicroStrategy Office API
- MicroStrategy Intelligence Server API

These APIs are used to customize and extend the functionality of the MicroStrategy platform, build new applications, or embed MicroStrategy functionality into other applications.

Common Framework for Any Client or Protocol

In MicroStrategy Web, specific features and functionality are available as a service through the architecture's Task Framework. Each specific service is called a task and, once defined, can be used from many different applications regardless of the how this task is accessed. Once MicroStrategy functionality is encapsulated in a task, an HTTP request, a Web Service, COM or .NET framework, or any other application can use it without any extensive coding.

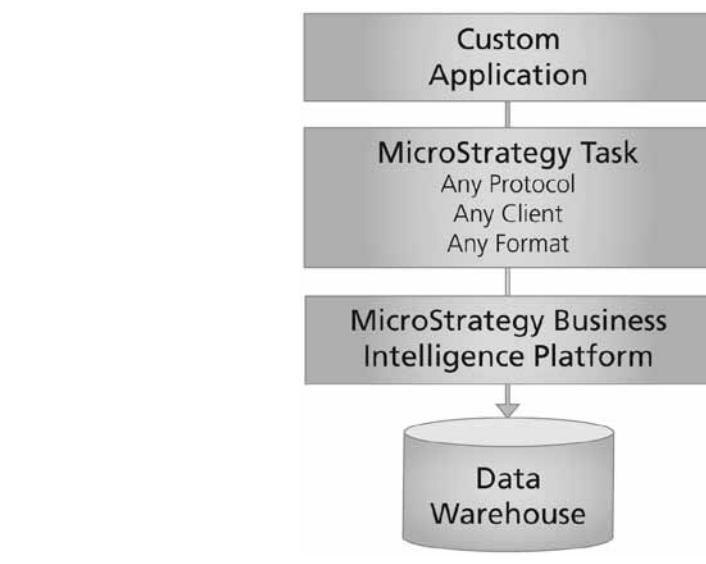


Figure 21-1 Common framework to integrate any client or protocol.

Plug-In Architecture for Customization Portability

MicroStrategy Web provides a plug-in architecture that packages customizations in a stand-alone piece of software or plug-in. A plug-in is a piece of code that is separate and distinct from the MicroStrategy Web source code and extends the original code. When using plug-ins, application developers do not need to make changes to out-of-the-box Web pages or functionality. A plug-in customization is deployed by placing the code files in a separate 'plug-ins' folder in the MicroStrategy Web installation. When the Web Server is re-started, the plug-in code merges with, and replaces, the out-of-the box MicroStrategy Web files. Since plug-ins are separate add-ons to the original source code, they can be added or deleted at will without affecting out-of-the box MicroStrategy Web or other customized MicroStrategy Web features. The plug-in architecture ensures portability of customizations and increases MicroStrategy Web's extensibility. MicroStrategy provides a Web Customization editor as part of the MicroStrategy SDK, which automatically creates plug-ins for customizations.

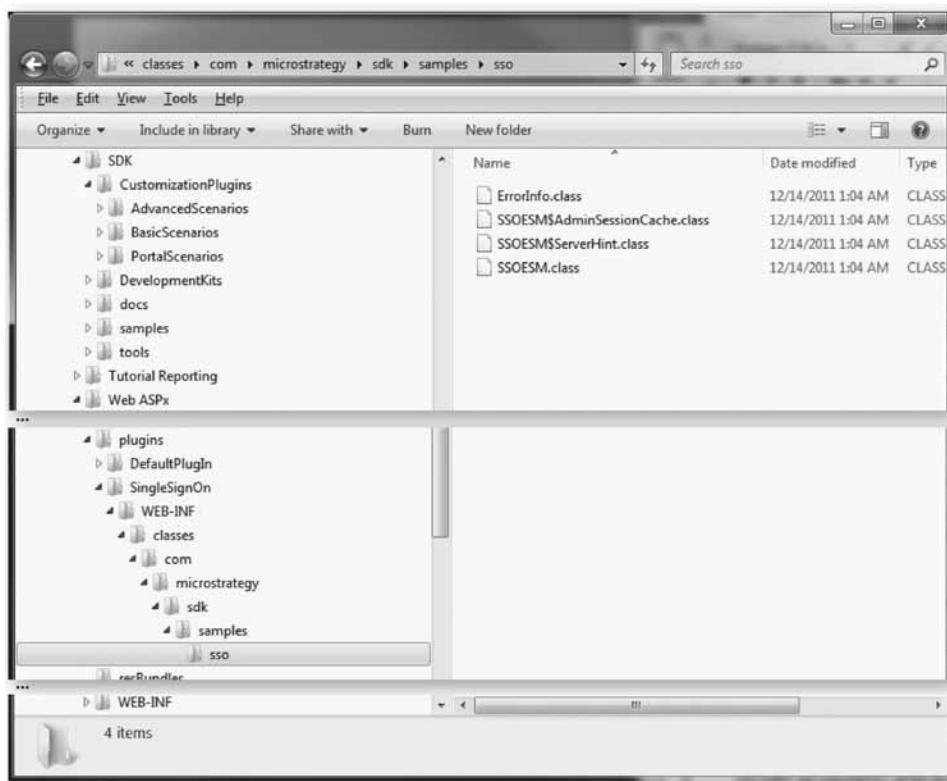


Figure 21-2 An SDK plug-in for Single Sign-on user authentication is contained in the plugins folder ensuring that customizations can be applied to different MicroStrategy versions.

21.2 MICROSTRATEGY WEB API

Most organizations that customize the MicroStrategy BI platform use the Web API. The Web API allows customization of the MicroStrategy Web interface and development of custom Web applications that provide interactive reporting functionality to any browser, portal, or third-party Web application. The Web application workflow involves four distinct stages:

1. Define the page structure, and instantiate BI contents displayed on the page
2. Execute user actions requested
3. Collect BI data from Intelligence Server
4. Render BI content, and arrange content on the page

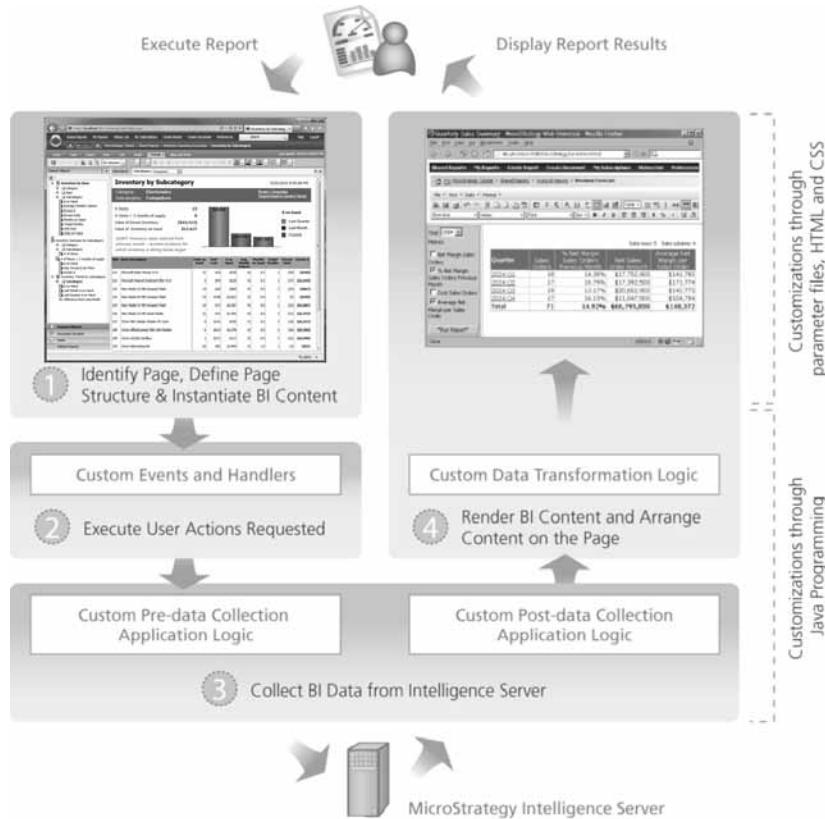


Figure 21-3 Web API enables extensive customization of every stage of the page execution workflow.

The first and last steps of the Web application workflow define layout of the Web interface, and present visually appealing rendering of BI content such as folder contents, report data, prompts, schedules, and subscriptions. Easy to edit and maintain XML configuration files, Cascading Style Sheets (CSS), and HTML-like Layout Definition files specify the page definition, and rendering of reports and other BI content on any page. No ASP.NET, JSP, or Java coding is required for most presentation-related customizations.

The second and third steps of the page execution workflow define the application functionality. An example of functionality customization is the integration of a mapping tool to display geospatial visualization of business data. These more advanced customizations require Java programming skills.

Modify Page Layout and Formatting Using Parameter Files

The layered Web architecture and comprehensive Web API allow developers to customize the structure, contents, layout, and formatting of every Web page through externalized parameter files. The benefits of externalization of common customization settings into configurable Parameter (XML), Style (Cascading Style Sheets), and Layout Definition (xHTML) files include:

- Easy customization of the application through textual changes to the parameters in configuration files
- Most common customizations do not require formal programming skills. Modifying configuration files, CSS, and Layout files does not require Java or .NET programming
- Seamless upgrades to future versions of MicroStrategy Web. The configuration files provide a ready catalog of all changes made to the default configuration of MicroStrategy Web, resulting in customizations that are easy to maintain, extend, or roll back.

MicroStrategy Web provides numerous preconfigured pages specifically tailored to the activity performed on each page. The default pages in MicroStrategy Web include a Login/Logout page, a Shared reports a page, a My Reports page, a Report Execution page, a Create Report page, a Create Document page, and a My Subscriptions page, among others.



Figure 21-4 A schematic representation of the Report Execution page displays the various page sections that comprise the page.

Every page of MicroStrategy Web has a well-defined structure containing numerous page sections. The page sections and the content displayed within them are defined independently of each other, giving developers the utmost design flexibility and application functionality. Additionally, each page can have multiple states, including an initial state, a final state, and any intermediate states that show the progress of the requested actions.



Catalogs All Pages that Comprise the Web Application

Figure 21-5 A Page configuration XML file defines the framework of the MicroStrategy Web application. It lists all the pages, the associated page settings, and common application settings such as mapping of system folders to pages, menus, and toolbars.

The Page Configuration file catalogs all pages available to the Web application in an easy to understand and manageable format. In addition to customizing the page structure and BI content of default Web pages, organizations can add custom Web pages very easily into the application by referencing the new pages in the Page Configuration file.

MicroStrategy Web provides an extensive set of Transforms that render BI information retrieved from Intelligence Server into viewable formats (e.g. HTML) for display in the Web interface. The Transforms use Styles and Layout Definition files to specify the presentation of BI content and its arrangement on a Web page. A Layout Definition file embeds MicroStrategy-specific tags into a familiar HTML structure. Externalization of presentation-related customizations into layout definition files and cascading style sheets (CSS) enables easy maintenance and upgradeability of customizations.

A centralized XML configuration file called Style Catalog (styleCatalog.xml) references all transforms, styles, and layout definitions available to an application.

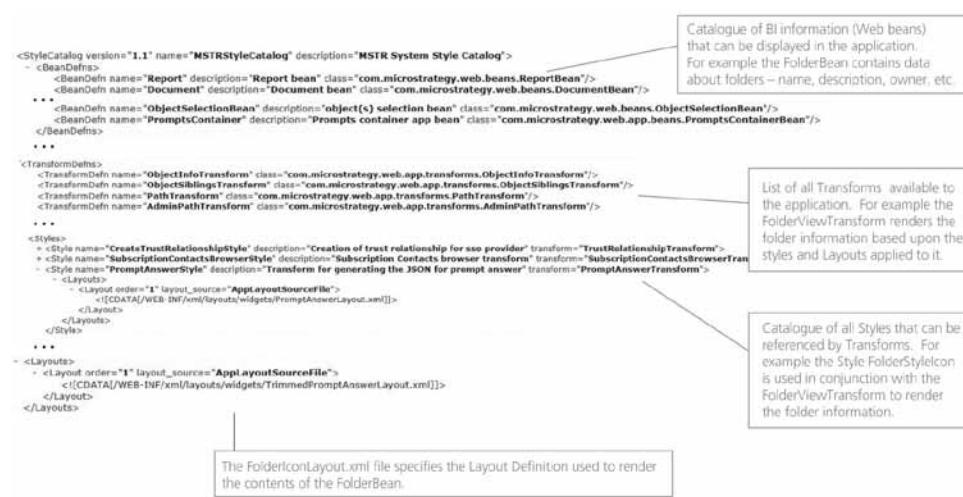


Figure 21-6 All the Transforms, Styles, and Layout files are cataloged in an easy to change XML configuration file called the Style Catalog.

Extend Application Functionality Through Java Programming

Three categories of customizations using the Web API enable highly customized application logic and presentation formats:

- Functionality customizations that affect requests only on a specific page
- Functionality customizations that affect requests on all pages of the application
- Advanced transformation of BI content to alternate presentation formats

Each of These Three Categories of Customization Requires Java Programming.

User actions that request specific application functionality, such as clicking on a report to execute a report, or clicking on a change-to-graph view icon represent “events” to the Web API. Events trigger the set of application tasks that modify the state of individual BI components or Web beans (Java beans) on a page. They occur before information requests are sent to the Intelligence Server. An event handler manages the execution of the event. For example, event handlers specify how to process user actions such as execute reports, change report formatting, sort report results, and pivot rows and columns of a grid. Custom application functionality that is specific to particular pages can be implemented through existing MicroStrategy Web event handlers or through new ones.

Unlike events, “add-ons” deliver application functionality that requires specific behavior every time any MicroStrategy Web page is loaded. Examples of common tasks performed by add-ons in MicroStrategy Web include setting execution and result flags, and specifying rules for incremental fetch on grids and graphs. Add-ons execute custom code before or after data are retrieved from Intelligence Server.

The third category of Web customizations extend existing transforms or create new transforms. Custom transforms display BI information retrieved from Intelligence Server in custom presentation formats required for the BI application. For example, custom transforms can render report results in eXtensible Business Reporting Language (XBRL), plain text, or binary data instead of HTML.

Perform Customizations From the Eclipse READY™ Web Customization Editor

Application Developers can customize MicroStrategy Web using the Eclipse READY™ customization editor in the MicroStrategy SDK. This editor integrates fully with the Eclipse IDE and provides complete functionality of the IDE when performing MicroStrategy Web Customizations. Complete with intuitive wizards, the editor eliminates the need to manually modify any of the configuration or presentation files. The editor also saves the customized plug-ins automatically in the plug-ins folder.

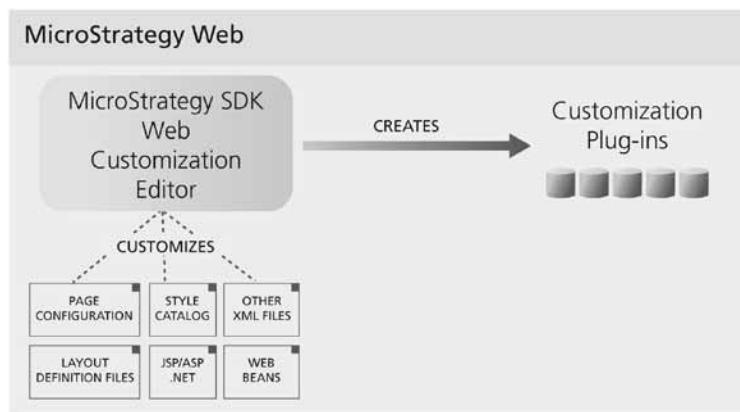


Figure 21-7 Eclipse READY Web customization editor specifically designed to create customizations visually.

The editor is equipped with wizards that enable application developers to create different types of customizations, such as Edit/Create Web pages, Edit HTML Menus, and Style Creation. The wizards also create new tasks using MicroStrategy Web functionality easily.

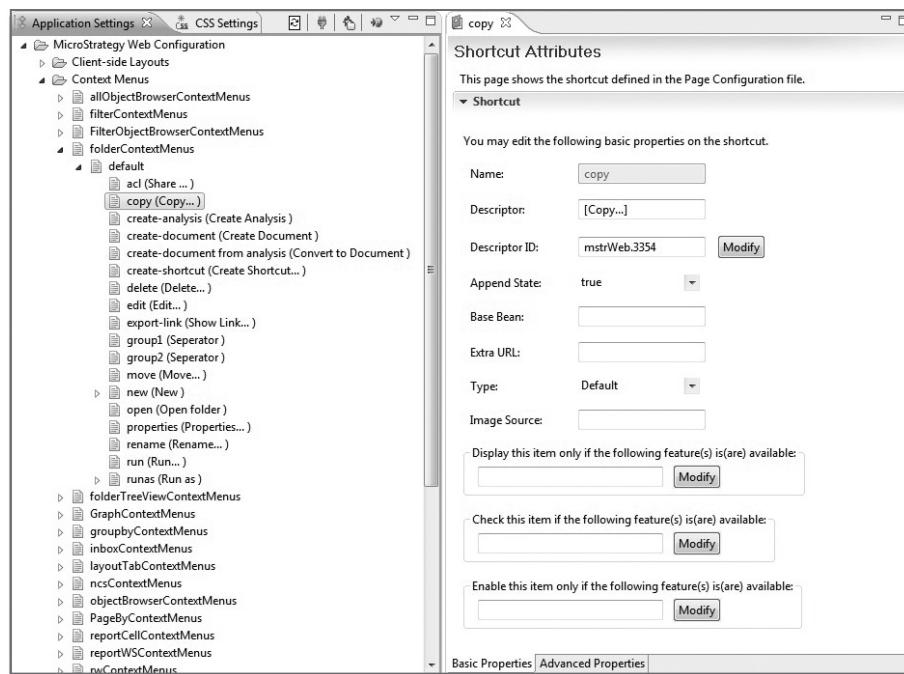


Figure 21-8 MicroStrategy Web Customization Editor integrated with the Eclipse enables application developers to customize MicroStrategy easily.

21.3 TASK FRAMEWORK

The Task Framework lets MicroStrategy Web behave as a service provider and delivers BI data as a service to external applications. Using this framework, an external application can access MicroStrategy functionality without needing to write custom Web API code. Each service, called a task in MicroStrategy, contains specific MicroStrategy Web functionality, such user login, report execution, element browsing, and metadata searches. External applications can discover MicroStrategy Web functionality without deploying it on another Web Server. Any protocol (HTTP, URL) or framework (.NET, COM, Java) can access a MicroStrategy Web service or task. Additionally, this Task Framework ensures that the external application accesses and receives the specific, relevant content rather than invoking all parts – necessary as well as unnecessary – of the custom code.

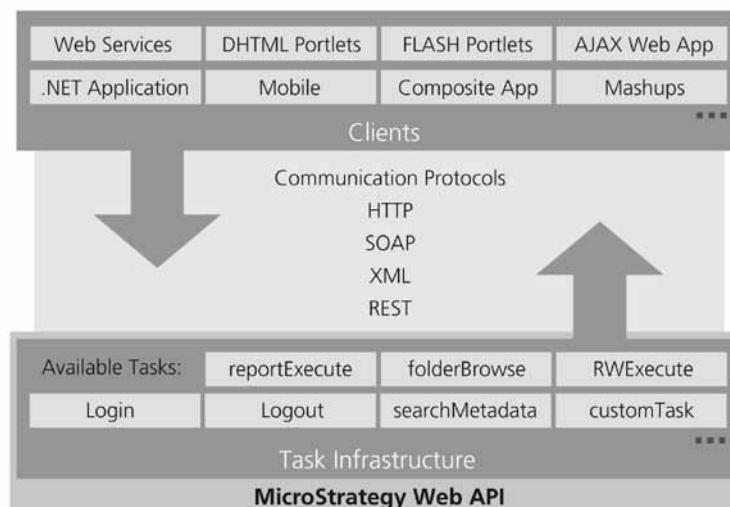


Figure 21-9 Task based Infrastructure facilitates MicroStrategy Web functionality to be accessible from any client or protocol.

It is quite common to provide MicroStrategy BI functionality embedded in a third party application using MicroStrategy Web's Task Framework where the third party application processes unformatted data retrieved from MicroStrategy.

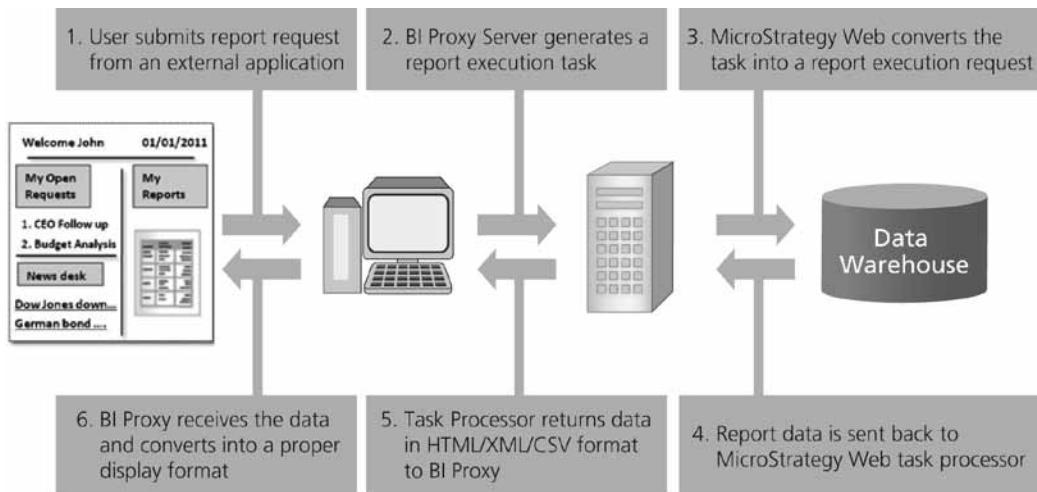


Figure 21-10 Report Execution Workflow with MicroStrategy used as a service provider.

A Comparison Between the MicroStrategy Web API and the Task Framework

The following table shows a comparison of the MicroStrategy Web API and the Task Framework.

Web API	Task Framework
Used to customize out-of-the-box MicroStrategy Web	Used for embedding MicroStrategy Web functionality into external applications
A Web API customization is page based, that is the entire page (customized as well out of the box content) is processed	Specific MicroStrategy Web functionality is bundled into individual tasks, which means only content in a specific, invoked task is processed
MicroStrategy Web behavior, such as look and feel, interfaces, wait pages, rendering etc. is changed with Web API based customizations	Tasks, in a services based infrastructure, deliver specific BI content to a third party application and have no control on the application's behavior
Usually, API based customizations require extensive coding, it is difficult to reuse them as-is and will require significant modifications	Since a feature or functionality is packaged into a specific task, each task can be reused from many different applications, as many times as needed

Figure 21-11 MicroStrategy Web is customized through the Web API while the Task Framework is used to integrate with other applications.

21.4 MICROSTRATEGY MOBILE API

The MicroStrategy Mobile API enables mobile app developers to customize MicroStrategy Mobile app on Android and iOS mobile devices by changing the look and feel of the app, creating custom visualizations, or developing new Mobile Apps that use MicroStrategy BI content. MicroStrategy provides different APIs for Android and iOS mobile devices.

The flexibility of the Mobile APIs enables app developers to build custom mobile applications that are consumers of BI data in two ways:

1. Customizing the MicroStrategy Mobile app project provided with the product
2. Integrating Mobile frameworks inside a custom X-code project

Using the Mobile SDK, Mobile application developers can create a customized version of the MicroStrategy Mobile App or design a completely new Mobile application based on MicroStrategy BI data. The Mobile SDK enables application developers to create custom visualizations as well to display BI data through the MicroStrategy Mobile App.

The Mobile SDK architecture utilizes the Task Framework and uses the Model-View-Controller (MVC) paradigm as shown below:

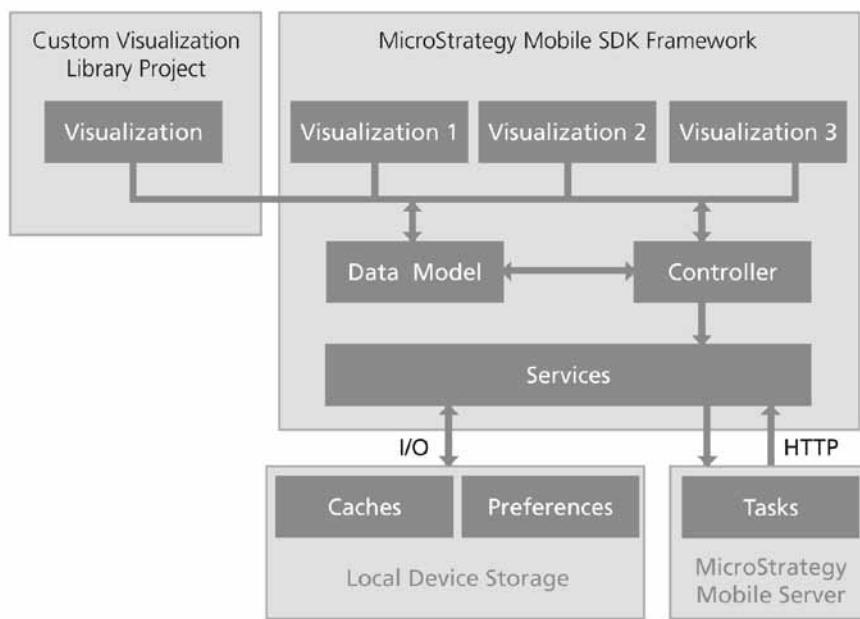


Figure 21-12 Mobile SDK architecture for iOS-based Mobile Applications.

The controller is the intermediary between BI content and the visualizations that render the BI data. Visualizations use the data from the data model and render it to the user. The Services layer communicates with the MicroStrategy Mobile Server through HTTP requests using the task infrastructure. A report execution task, for example, is created when a user taps on a report or document from the Mobile app. Using the BI content returned by the services layer, the controller populates the data model which holds the data and acts as the data source for the various visualizations used in the document. The data model communicates with the visualizations through the controller.

The Mobile SDK documentation also discusses the following customization scenarios

- Rebranding MicroStrategy Mobile App to change the name, icon and the splash screen.
- Deploying multiple MicroStrategy Mobile apps on the same mobile device
- Customizing the map visualization to use custom markers or a custom info window

21.5 MICROSTRATEGY VISUALIZATION API

The Visualization API is a framework that enables application developers to create custom visualizations for dashboards rendered in DHTML or Flash by using the Visualization Framework and Adobe Flex programming. Using the Visualization Framework, application developers can build new visualizations or extend visualizations to match specific business needs. Application developers can build rich external, stand-alone applications that integrate MicroStrategy reports and documents with other content as well incorporate interactivity and workflows that are not available in MicroStrategy Web.

The Visualization Framework provides several tools to build custom visualizations that include:

- A library of commonly used, out-of-the-box pre-wired controls, such as selector controls, graphs and grids, which can quickly be used as part of a custom visualization
- A capability to create new custom controls based on pre-wired controls
- An event-based mechanism to apply filtering conditions on the controls
- Editors to create derived models, setting targets, rendering, and for easy deployment
- Samples and documentation

The Visualization Framework is Integrated with Adobe Flex Builder

The Visualization Framework builds Flash-based dashboards that display MicroStrategy data in custom visualizations while taking full advantage of MicroStrategy's sophisticated architecture. The Visualization Framework has two components, a framework with editors, wizards and dialogs for designing custom visualizations and Adobe Flex helper objects, Adobe Flex controls, and pre-built widgets. The helper objects communicate with MicroStrategy Web using the Task Framework to retrieve BI content and pass it on to the visualization. The Visualization Framework provides a MicroStrategy Web container that provides out-of-the-box access to MicroStrategy functionality and sits on the standard Adobe Flex Architecture.

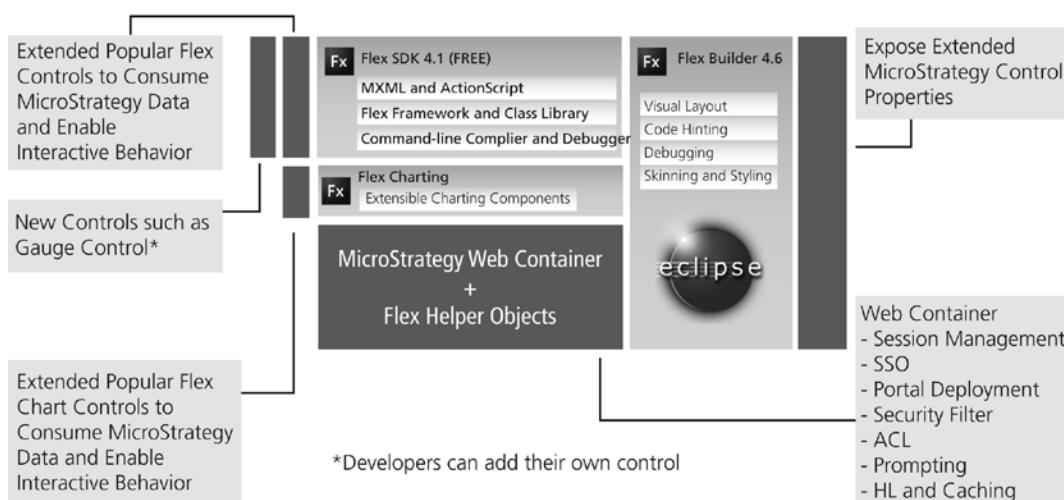


Figure 21-13 MicroStrategy Visualization Framework enhances the Adobe Flex Architecture.

As with the Mobile SDK, the Visualization Framework uses the Model-View-Controller framework consisting of three components

1. Controller – This class is the single point of entry and serves as the bridge between the Model and the View components. The Controller handles every single user action, for example a user pressing a button, and passes the event to Model.
2. Model – The Model stores all the raw data and provides it to the various views or controls. Depending on the type of user action, the model will either return a subset of data or the entire dataset to the view
3. View – The view or the visualization populates the data into the resulting visualization state from the user action.

Sample Visualizations are Included in the SDK

The Visualization Framework provides several sample visualizations, their descriptions and the sample code. The samples are grouped into six categories:

1. Controls which include labels, charts, and grids (tabular and cross-tab).
2. Data management samples that include aggregation and adding custom totals to visualizations.
3. Basic customizations that change the look and feel of grid and graph customizations with data layout different from out-of-the box samples.
4. Advanced customizations, such as implementing gauge controls, tree controls, and custom lists that support the loading of prompted data and implementing write-back from the visualization.
5. Two custom samples that showcase the power of visualizations.
6. Several Adobe AIR applications that store BI data offline.

21.6 MICROSTRATEGY OFFICE API

The MicroStrategy Office API allows developers to build custom applications that use Microsoft Office products as a user interface to the MicroStrategy BI platform. The MicroStrategy Office API creates powerful Word, Excel, and PowerPoint applications that incorporate insight gleaned from the MicroStrategy BI platform. The Office API provides COM interfaces for VBA, Visual Basic and C++ customizations and .NET interfaces for C# and VB.NET customizations.

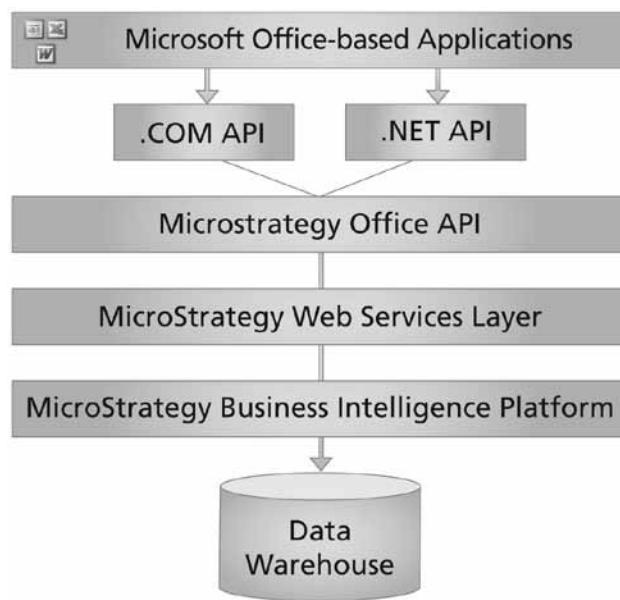


Figure 21-14 Diagram showing the MicroStrategy Office API Architecture.

MicroStrategy Office client applications exchange SOAP based messages with MicroStrategy Web Services infrastructure deployed on the Web server, which communicates with the Intelligent Server. The API facilitates access to multiple MicroStrategy projects, selection of one or more reports or dashboards, automatic prompts answering, and embedding of report results into Microsoft Office applications.

21.7 MICROSTRATEGY INTELLIGENCE SERVER API

MicroStrategy Intelligence Server contains an API that can access all aspects of application modeling and development, report execution and data manipulation, and platform administration and monitoring. The Intelligence Server API has the following features:

- Fully thread-safe
- Object-oriented, with more than 3,000 documented methods and properties
- Any third-party application can access Intelligence Server running on the Windows, Linux, or UNIX platforms.

The Intelligence Server API is not documented in the SDK documentation. It is used internally by the MicroStrategy products that connect to Intelligence Server.

21.8 COMPREHENSIVE DOCUMENTATION THAT DESCRIBES THE APIs IN DETAIL

A key component of an open platform is comprehensive documentation that describes the methods and properties used to access the objects and functionality of the platform. The MicroStrategy Developer Library provides:

- A comprehensive documentation of the various APIs of the MicroStrategy BI platform
- API references for the various categories of API
- Over 80 task-oriented customization and integration examples with sample code
- A Customization Explorer to easily navigate through samples of web page, features, and visualization customizations

All of these are available in a searchable and indexed format that is easy to navigate.

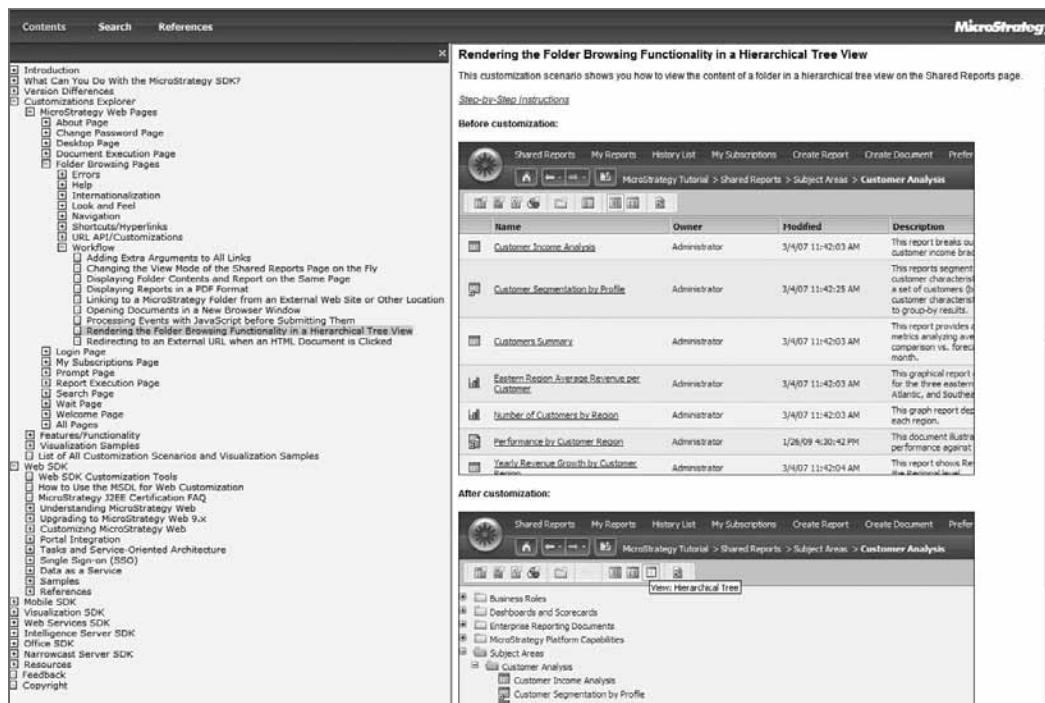


Figure 21-15 The Customization Explorer gives easy access to web pages, features, and visualization customization samples.

Topics covered in the MicroStrategy Developer Library include:

- Connecting to a BI application and browsing for reports, filters, and metrics
- Modifying the navigation and workflow of the Web interface
- Modifying the page layouts and content displayed
- Running reports and documents, and retrieving report results
- Retrieving prompt parameters and automatically answering prompts to specify report parameters
- Manipulating report results (drilling, sorting, pivoting, etc.)
- Implementing single sign-on (SSO)
- Setting up customized authentication and validation security paradigms
- Performing user management tasks such as creating users, organizing them in user groups, creating security filters, and assigning them to users
- Integrating with JSR-168 portals
- Integration with other external applications
- Creating custom plug-ins
- Setting up tasks that provide MicroStrategy Web functionality as a service

The API Reference provides detailed information of all packages along with a summary of classes and interfaces. The reference also lists the class hierarchies for the packages that comprise the API.

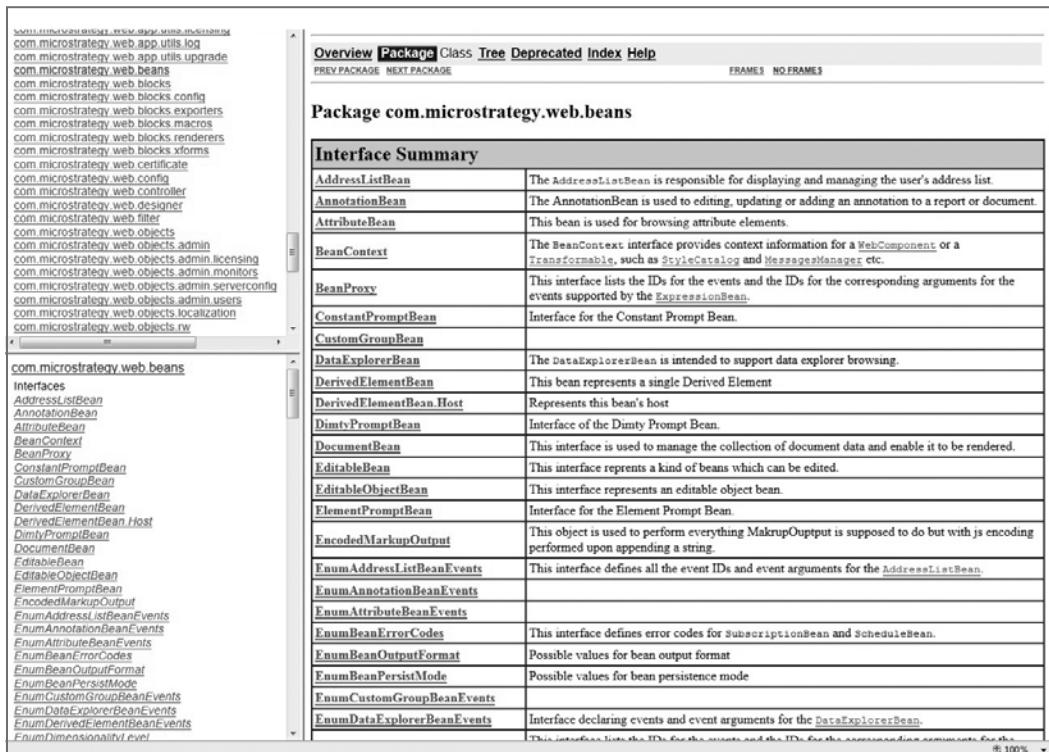


Figure 21-16 The MSDL provides a comprehensive reference of all categories of API using interactive HTML-based Java Docs.

21.9 PORTAL INTEGRATION

The MicroStrategy SDK provides detailed instructions necessary to incorporate the full range of BI functionality, formatting, and user interactivity available to the MicroStrategy Web interface in any commercial or homegrown portal. The integration with portal technologies stems from MicroStrategy conformance to industry standards, such as J2EE, .NET, XML, JSR 168, WSRP, and SOAP.

Deployment-Ready MicroStrategy Portlets

Deployment-ready portlets require no additional programming when delivering MicroStrategy Web reports and functionality in the following third-party portal and content server products:

- Microsoft SharePoint Portal
- IBM WebSphere Portal
- SAP Netweaver Portal
- Oracle WebLogic Portal
- Liferay Portal
- DotNetNuke
- Drupal

The MicroStrategy portlets embed folders, reports, documents, user history lists, and a search page into the portals through easy-to-configure screens. The portlets provide the full range of OLAP manipulations, such as sort, pivot, add subtotals, export, and add new calculations, as well as design functionality, such as changing the report display between grids and graphs, and toggling thresholds.

MicroStrategy Web contains detailed instructions for the installation, configuration, and deployment of the portlets. The portlets leverage the portal's credential storage mechanism to perform single sign-on. In addition, they also provide personalization by saving and retrieving user-specific information from the portal repository, and making it available to MicroStrategy at runtime.

Sample Code for Integration with Other Portals

A Portal Integration Kit includes sample code and documentation for integrating MicroStrategy Web with other enterprise portals. It illustrates the integration architecture, workflow, and usage of the API in the context of deploying Web functionality to portals. In addition, the SDK includes sample code to embed MicroStrategy Web functionality into any portlet that conforms to the Java Specification Request (JSR) 168 Portlet Specification.

Sample Code for Deploying Microstrategy within the Apache Struts Framework

Struts (<http://struts.apache.org/>) provide an open source framework for building Java Web applications. The core of the Struts framework is a flexible control layer based on standard technologies such as Java Servlets, JavaBeans, ResourceBundles, and XML. The SDK provides sample code to incorporate MicroStrategy functionality within a struts framework.

21.10 WEB SERVICES DEVELOPMENT

Web Services allow loosely coupled integration with other applications. External applications can use Simple Object Access Protocol (SOAP) to access any MicroStrategy functionality. A Web Services Development Kit

composed of sample code and documentation describes MicroStrategy's Web services API. It provides a starting point for the integration of MicroStrategy reports with other Web applications using Web Services.

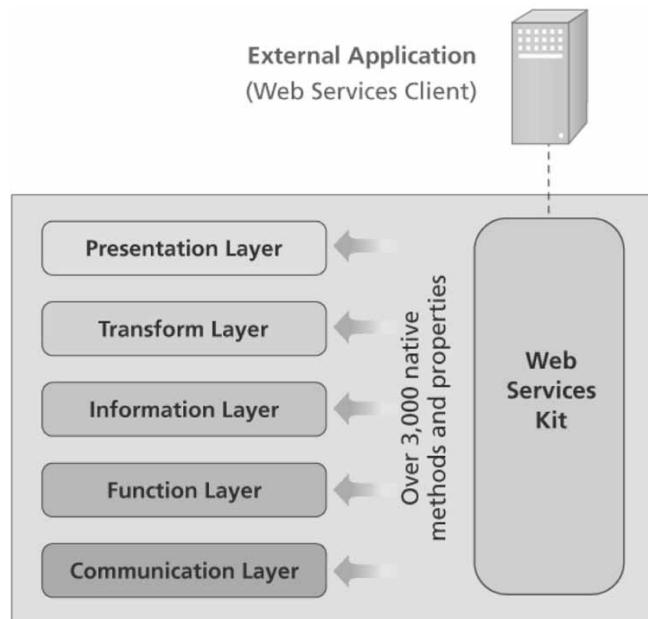


Figure 21-17 The Web Services Development Kit integrates BI applications with external applications using Web Services.

Some common functionality available through Web Services includes:

- Connect to and disconnect from the MicroStrategy-based application
- Browse for dashboards, reports, and other metadata objects
- Answer prompts
- Retrieve attribute element lists
- Run reports and documents
- Incrementally fetch more results
- Drill for more data
- Retrieve the history list for a user
- Support for Industry Established Standards

The Web services platform provides access to BI reports and documents through a SOAP query. The SOAP call returns data either as XML or other format to the client application. MicroStrategy's Web Services platform does not require third party adapters nor does it require a thick client implementation. MicroStrategy Web Services supports a range of industry-wide standards as stated below:

- SOAP standards
- HTTP communication protocols
- XML support for messages and XML schema support
- WSDL for Web services description languages
- DIME for attachments
- Document-literal-wrapped convention for SOAP requests
- SSL for secure communication

21.11 SAMPLE APPLICATIONS

The MicroStrategy SDK includes several sample applications and utilities that assist in the development process or administration of the BI infrastructure. These include:

- Single Sign-on Sample

This sample application demonstrates the creation of a single sign-on experience for the MicroStrategy user base.

- External Security Module

An application that demonstrates integration with external security sources for authentication of users through a third party security system.

- Session Helper Bridge

The Session Helper Bridge sample application illustrates how to create a session that connects to MicroStrategy Intelligence Server in a .NET environment.

- Extended Property Editor

Any object in the MicroStrategy metadata can store additional information in Extended Properties. For example, a developer can create an extended property on all reports that enables end users to write report data back to the data warehouse. The Extended Property Editor allows any developer to associate a series of extended properties with any metadata object.

- Thick Client Application

This application demonstrates the process to embed MicroStrategy Web into a third-party application and uses the URL API to display MicroStrategy content.

- Web Customization Upgrade Wizard

The Web Customization Upgrade Wizard, provided as a part of the Web Customization Editor, upgrade customizations from MicroStrategy 8.x to the current version. Every customization is converted into a plug-in and is deployed into the plug-ins folder.

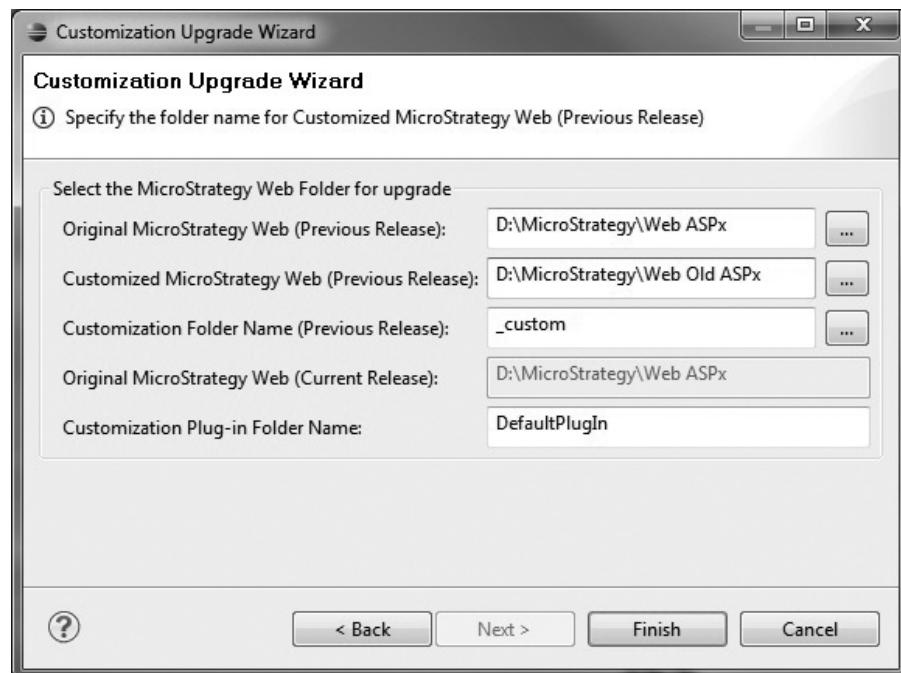


Figure 21-18 Customization upgrade wizard converts older customizations into plug-ins.

- Inventory Management Sample for the iPhone

This Mobile sample showcases an inventory management application which enables business users to track and modify orders.

Code Samples

MicroStrategy SDK includes the code samples to assist application developers to get started with customization tasks.

- Java class and source files
 - Beans
 - Dialogs
 - Events
 - Transforms
- XSL files

The SDK contains additional code samples in the MicroStrategy Developer Library documentation.



Figure 21-19 The MicroStrategy Developer Library contains numerous examples with sample code.

Categories of sample code in the MicroStrategy Developer Library include:

- Add-ons: Sample programs that illustrate the use of add-ons to answer prompts based upon URL query string contents and redirects to remote URL's
- Events: Sample code that illustrates how to add custom logic upon modification of a user's password, event handlers for reports, prompts, and Web bean-specific events

- Transforms: Sample code to edit, modify, and display contents in context-sensitive menus
- Security: Sample code to implement single sign-on solutions with Identity Management applications and user validation
- Tasks: Pre-created tasks such as report execution task, login task, and many more
- Visualizations: Sample code to display and manipulate data in Flash

21.12 CUSTOMIZATION AND EXTENSION EXAMPLES

Many MicroStrategy customers customize or integrate their MicroStrategy platform to some extent. The range of customization varies widely. Some customizations entail altering the MicroStrategy Web user interface to adhere to corporate look-and-feel guidelines. Other customizations add new user functionality directly to the MicroStrategy code set. In a third type, customers embed MicroStrategy functionality deep within other applications that they have written or purchased.

As a result, customizability and extensibility have been a major thrust of MicroStrategy engineering for the past several years. Specifically, MicroStrategy is adding more and more features to make MicroStrategy customizations easy to implement and maintain throughout the BI platform.

Customization Ease	Common Customization Goals	Enabling MicroStrategy Feature
Out-of-the-box	Embed MicroStrategy Web in a Portal	Portal Integration Kit
	Embed MicroStrategy Web in other Web Applications	Zero footprint Frame Integration
Parameter-driven	Modify the look and feel of the Web Interface. Define custom layouts and formats for BI data such as prompts and reports	Configuration files, CSS, and Layout Definition files
Basic Programming	Provide advanced rendering of BI content and new application functionality	J2EE™ compliant architecture with specialized Java constructs – Transforms, Events & Add-ons
Sample Code & Tools	Implement Single Sign-On (SSO) solutions	External Security Module
	Create integrated Microsoft Office applications	Web Services & MicroStrategy Office API
Full Programmatic Control	Access all MicroStrategy functionality from other applications	Service-Oriented Architecture with 3000+ methods and properties
	Integrate with system control and administration tools	Functionally rich Intelligence Server API
BI content as a Service	Access common MicroStrategy content from external third party applications as a service	Task Framework & Web services API
Plug-ins	Deploy extensions to MicroStrategy Web without modifying the source code	Plug-in Architecture

Figure 21-20 Common customizations and extensions performed on the MicroStrategy BI platform.

Embed MicroStrategy Web in Enterprise Portals

Organizations can extend the reach of BI reporting and analysis by deploying the full range of BI functionality, formatting, and user interactivity available to the MicroStrategy Web interface in any commercial or homegrown portal.

Embedding multiple MicroStrategy portlets on a single portal page offers organizations the ability to create dashboard-style applications. Each MicroStrategy portlet optionally calls a customized report execution page with just the interactive reporting capabilities required. MicroStrategy portlets provide the full range of functionality of MicroStrategy Web. These include:

- Displaying toolbars to change views of reports, format, and export data;
- Drilling for more details within the portlet or to a new window;
- Enabling context-sensitive right-click menus;
- Performing OLAP manipulations such sort, pivot, page by slices of the report, and add new calculations.

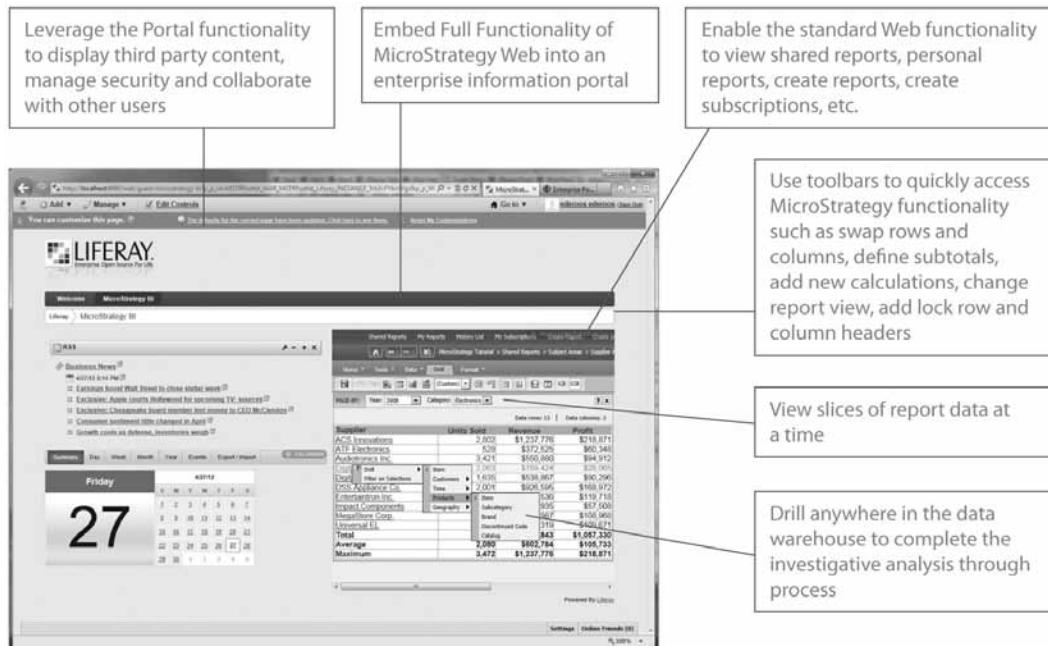


Figure 21-21 Embed the full range of MicroStrategy functionality into enterprise portals using deployment-ready portlets or Web parts, while leveraging the portal's security and collaborative capabilities.

On the other end of the interactivity spectrum, a portlet can provide alert-style reports with no user interactivity.

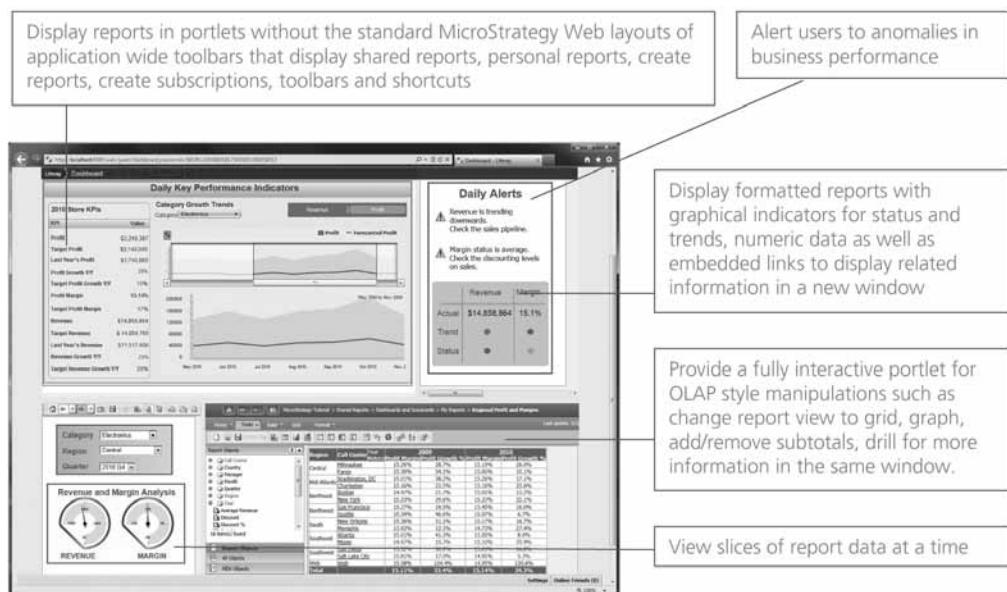


Figure 21-22 Create powerful dashboards that incorporate content from multiple data sources while providing a wide variety of end user functionality within individual portlets that comprise the dashboard.

Another requirement in portal applications is the ability to control the content displayed, and even the application logic of other applications based upon the state of a “master,” or controlling portlet. MicroStrategy portlets pass parameters such as report names, prompt answers, attribute elements, and metrics values to “servant” portlets that use the information for further processing. A simple illustration of this scenario involves a controlling portlet that displays report folders and report lists. Clicking on the report name in the master portlet displays the report results in one of the servant portlets. Parameters can be passed between MicroStrategy portlets and non-MicroStrategy portlets too.

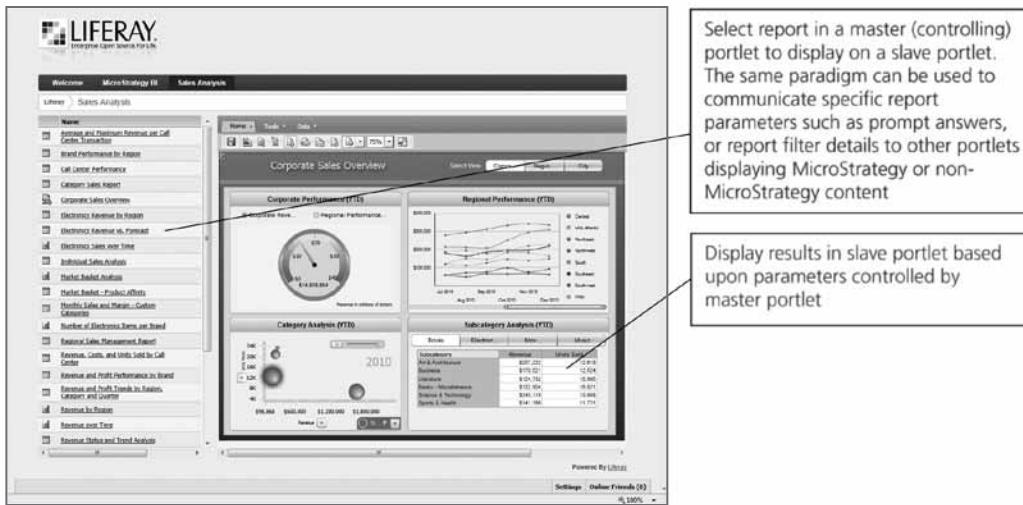


Figure 21-23 Build powerful portal applications that use MicroStrategy content such as report names, prompt answers, filter selections to control the information and actions of one or more slave portlets.

Building Custom Portlets

Custom Portlets are built when the out-of-the-box MicroStrategy Portlets packaged with MicroStrategy Web do not integrate with the portal product being used. A custom portlet will require coding that includes a MicroStrategy Web URL request from within a frame in the portal page. The request contains information for connection, authentication and report execution. The iFrame, when added to the code of the portal page, becomes a portlet. There are several ways to build a custom portlets in MicroStrategy as discussed below:

1. Using Sample JSR 168 MicroStrategy portlet
If the portal product complies with the JSR168 standard, then the MicroStrategy provided JSR-168 compliant portlet code can be plugged into the portal product. Other than storing user level information, all other out-of-the-box portlet functionality is available.
2. Using Sample JSR 168 MicroStrategy portlet as basis for building a custom portlet
If the portal product does not comply with the JSR168 standard, common actions such as helper methods to build URLs can be used. This involves replacing all the JSR 168 calls with the API calls supported by the portal server. Once the API calls are replaced, all out-of-the-box portlet functionality except storage of user level information is available.
3. Using a Portal Server’s iFrame Product
If the portal server has a proprietary iFrame portlet, then this built-in capability is useful to integrate MicroStrategy into the portal server. Using a Portal Server’s iFrame product restricts the portal developer to the properties supported by that specific portal product.

Embed MicroStrategy Web in Other Web Applications

Third-party applications easily incorporate a significant portion of MicroStrategy Web functionality through appropriately constructed URL's. OEM partners of MicroStrategy use their industry expertise to create custom analytic applications using the MicroStrategy platform. They then package these applications and sell them as their own branded products. This simple approach to integrate MicroStrategy with external applications serves numerous use cases:

- Link external applications or websites to any MicroStrategy application in the organization
- Browse contents of folders
- Execute reports and documents

Define Custom Report Workflow Using Linked Drilling From a Document to a Report or Other Documents

- Display complete MicroStrategy pages within third-party applications by passing the appropriate URL parameters from an embedded inline frame.
- Serve BI data to third-party applications in any format such as HTML, XML, and comma/tab-delimited.

Modify the Look and Feel of the Web Interface

The most common customizations of the Web interface are look-and-feel changes such as matching corporate standards, changing the content of the default MicroStrategy Web page and section layout or adding links to other Web sites. With MicroStrategy, one can accomplish most look-and-feel customizations using the Eclipse plug-in which changes parameters in the Page Configuration and Style Catalog files. In addition to these configuration files, MicroStrategy uses CSS and Layout definition files that allow application developers to apply formatting styles to the various elements displayed on a Web page. No Java or .NET programming is required to make these changes. As a result, customizations are easier to implement, deployment times are reduced, and support costs are lower.



Figure 21-24 Perform look and feel customizations such as including a customer logo, and modifying the page navigation paradigm easily using simple modifications of the parameter files and cascading style sheets.

Advanced Functional Customizations of the Web Interface

Some organizations modify the end user workflow in MicroStrategy Web to include custom functionality. Others build custom data transformation logic to render BI information to suit the needs of their applications. Using the MicroStrategy SDK, developers add the desired functionality or alter the report execution flow necessary for their specific applications.

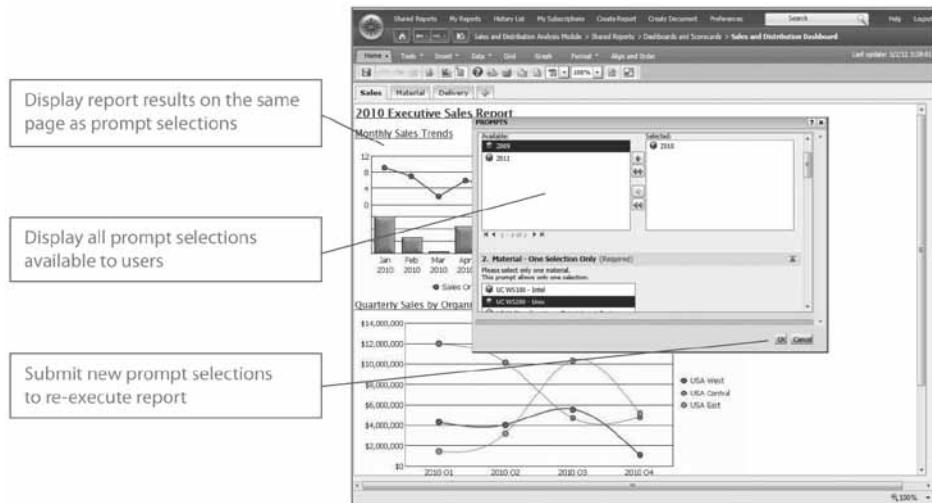


Figure 21-25 Add prompt selection to the report page. Users can change the prompt parameters, and re-execute the report to display the results on the same page.

For example, the default execution workflow of a report that contains prompts involves two distinct Web display steps:

- The first page displayed to the user contains the list of prompts in a variety of formats such as radio buttons, check boxes, and drop-down lists. End users make their selections, and click a button to proceed with the report execution. At this point, the prompts are closed and no more prompt selections are possible.
- The second page displays the report results from the report definition and the prompt selections made by the user.

Through changes to the prompt layout and report execution workflow, a customized report execution page can display a prompt dialog on top of the report. This modification can be implemented through simple changes in the page configuration file as well as the associated layout files. Using some Java code, the workflow can be customized so that the report is updated using the prompt selections.

Integrate with Third-Party Visualization Software

BI applications often need to address the specialized visualization needs of specific audiences. For example, pharmaceutical specialists, and biochemists require the ability to display complex visualizations such as chemical structures of pesticides or molecular structures of genes. Other applications that involve geospatial data require visualizations in map format. For these applications, the MicroStrategy BI platform serves as the data provider, while specialized visualization tools perform the data rendering.

Implement Single Sign-On Solutions

Organizations today face extensive scrutiny regarding access to sensitive data. To comply with these requirements, they implement sophisticated authentication mechanisms such as biometric authentication and

two-factor authentication to verify user credentials. Additionally, organizations require each user to enter his or her credentials when initially accessing the corporate network. Irrespective of how the users are stored in the rest of their enterprise applications, organizations expect every enterprise application to recognize the same user without requiring re-entry of the credentials.

MicroStrategy offers numerous options for implementing Single Sign-On (SSO). An External Security Module, sample code, and comprehensive documentation available in the SDK make it possible to integrate MicroStrategy with portals and identity management systems.

- Single Sign-On within a Portal Deployment: When deployed in an enterprise portal, the MicroStrategy portlet retrieves the user credentials from the credentials vault of the portal server, and passes the credentials to MicroStrategy Web. Organizations also have the flexibility of creating a user session in Intelligence Server prior to executing the user request using the External Security Module.
- Single Sign-On with Identity Management applications: When external authentication is required for all enterprise application users, as in the case of biometric and two-factor authentication, the policy server of the identity management system authenticates the user against the user directory. The identify management system passes a security token to MicroStrategy confirming the identity of the user. Custom code implemented using the External Security Module validates the token against the policy server, and authorizes user actions in MicroStrategy Web.

Integration with Office Productivity Software

Another common requirement is that BI applications be accessible from a variety of interfaces. With MicroStrategy Office, business users never need to leave their Microsoft Office application to access MicroStrategy reports. Using the MicroStrategy Office API, organizations create custom applications that incorporate reports from multiple BI applications. These applications automatically use the MicroStrategy Office API to select reports, answer prompts, and embed report results in Microsoft Office. Custom business logic that uses VBA, Visual Basic, Visual Basic.NET, C++, and C# programming is very easy to incorporate on top of the MicroStrategy Office API. As a result, Microsoft Excel, PowerPoint, and Word files become live repositories of scorecards, reports, charts, PivotTables, and Pivot Charts, continually linked back to the MicroStrategy BI Platform for the latest data.

Integration with Other Enterprise Applications

Given the openness of the MicroStrategy BI platform, external applications can instantly take advantage of BI functionality by plugging directly into the platform. The ability to plug into the MicroStrategy platform enables innovative integration with other enterprise applications. Web services and Task Framework provide a cost-effective Enterprise Application Integration (EAI) option. MicroStrategy's open, Service-Oriented Architecture makes it a superior business intelligence platform for Web Service applications. The BI content available as services or tasks enables external applications to access BI functionality built available in the MicroStrategy BI platform. In addition to executing reports in MicroStrategy, external applications can incorporate user selections of prompt answers or report parameters to control the behavior of third-party applications.

Integrate with System Management Tools

Automated management and system control is critical to non-stop operations of enterprise-class application deployments. Organizations regularly build sophisticated system control and administration applications or use third-party system management tools to manage their BI infrastructure. Common examples of custom applications include:

- Automatic disconnection and reconnection of database threads: Any third-party Windows application can access Intelligence Servers running on the Windows, Linux, or UNIX operating systems to disconnect all Intelligence Server connections to the database automatically upon the start of the ETL process. The custom programs reconnect the database connections automatically when the data warehouse load is complete.
- Logon metering: Average response times can suffer during periods of unusually heavy application usage. Custom rules can analyze detailed performance statistics and counters in real-time to load management. For instance, under peak loads, additional Web users can be prevented from connecting to the application.
- Smart Load Management: System Management tools can use the MicroStrategy APIs to continuously monitor Intelligence Server loads by checking the number of open jobs, user connections and open projects and terminate jobs and user connections when loads exceed a pre-set limits.

21.13 SUMMARY

The organic service-oriented MicroStrategy BI architecture exposes consistent, encapsulated BI functionality using thousands of documented interfaces and methods. Through documentation of the API, sample code, and utilities, plug-in architecture the MicroStrategy SDK empowers developers to implement highly customized, functional, and powerful Web reporting applications. The task framework enables application developers to access BI content as a service from an external application without every needing a Web Server. Using the Eclipse Ready customization editor, look and feel customizations of the Web interface can be changed by editing and maintaining XML configuration files, Cascading Style Sheets and HTML-like Layout Definition files. No ASP.NET, JSP, or Java coding is required for such presentation-related customizations. More advanced functionality customizations are possible using Java code. All of these capabilities ensure that the MicroStrategy SDK is provides a powerful and flexible layer to build customized applications.

APPENDICES

APPENDIX A: SCHEMA SUPPORT

In addition to the modeling techniques described in the chapters, the MicroStrategy BI platform contains features that extend the support for existing data warehouse schemas, from star to snowflake:

- Normalized and denormalized database schemas
- Attribute relationships
- Multiple facts tables and aggregate tables
- Heterogeneous column mapping
- Business hierarchies
- Attribute forms
- Expressions
- Dimensional extensions
- Logical tables
- Partition tables
- Inner and outer joins
- Data internationalization

For support of operational databases, data marts and Web services, the freeform query engine supports the following:

- Freeform SQL editor
- Query Builder
- XQuery reports

Support of Multi-dimensional cube databases, such as SAP BW, Microsoft Analysis Services, Oracle Hyperion Essbase, and IBM Cognos TM1, is achieved through the dynamic MDX engine:

- OLAP Cube Catalog to select cubes for reporting
- Automatic Metadata translation to convert cube metadata to MicroStrategy metadata

Normalized and Denormalized Database Schemas

MicroStrategy supports both normalized and denormalized lookup information, and combinations of these schemas. In a normalized schema, lookup tables have information about a single attribute with foreign keys to related attributes. Denormalized schemas have information about more than one attribute in a lookup table. Normalization results in less database storage needed to hold the data, but could require more table joins when aggregating data. A hierarchical structure defines the relationships between related attributes usually by their primary/foreign key constraints. Attributes are related to facts by their foreign keys in the fact tables.

Attribute Relationships

Attributes in a hierarchy are defined by parent-child relationships, which can be one-to-one, one-to-many, or many-to-many. These relationships describe how many child attribute values there can be for each parent attribute value. With many-to-many relationships, a separate “relationship” table is needed to define the valid parent and child combinations. MicroStrategy supports all three relationship types. Attributes are related to facts by their foreign keys in the fact tables.

Multiple Fact Tables and Aggregate Tables

With MicroStrategy, queries can extract fact information from many fact tables to satisfy the user's request. With MultiSource Option, these facts can reside in different databases. Facts can be defined on a single database column, or by using an arithmetic calculation on more than one column. Facts are stored at a level in the database corresponding to the attributes contained in the fact table. To improve performance of the database, fact columns can be aggregated to higher hierarchical attribute levels and stored in separate aggregate tables. MicroStrategy automatically takes base fact tables and aggregate tables into account when generating the most optimal query for a report or dashboard. The table used can be overridden for a specific report or metric as long as the table contains the fact(s) needed by the query. In addition, MicroStrategy has the flexibility to prevent a fact from being displayed at a particular level.

Heterogeneous Column Mapping

Data warehouses frequently combine information from disparate source systems. As a result, an attribute represented by a MONTH_ID column in one table might be represented by MNTH_ID in another table, and MON_ID in another table. Rather than requiring the batch load process to standardize the column names, MicroStrategy can designate the three column names as being used for the attribute Month. MicroStrategy Intelligence Server will take this equivalency into account when generating SQL and creating joins.

Business Hierarchies

Business analysts often need to pursue drill paths in the course of investigating the data that bear little or no resemblance to the physical warehouse structures. For example, a business analyst may need to drill from a particular store to the promotions that store ran in the past month. Promotions may be stored in a different physical warehouse table than stores, yet MicroStrategy can define a hierarchy that allows the user to seamlessly drill down from Store to Promotion.

Attribute Forms

An attribute such as Month may have several descriptors or attribute forms. For example, Month may be viewed as “January 2012” by some users, “2012 01” to others, “2012 12” to users of a fiscal calendar that ends in January, or “Janvier 2012” to French users. MicroStrategy allows the definition of multiple forms for each attribute that enable each user to see exactly the form that is most relevant to them. Attribute forms even allow the inclusion of HTML image tags in a report, and allow images to be displayed on a report.

Expressions

Ideally, business analysts should not be restricted to only the data that is physically stored in the data warehouse. They should also be able to manipulate particular columns. Taking the month attribute example discussed in heterogeneous column mapping, a month attribute can be defined as “MONTH(date)” using one of MicroStrategy's many built-in functions. Another example is a customer's age that is best calculated by taking the difference between the customer's birth date and the current date, using a database function. Expressions allow such calculations without requiring changes to the ETL process. Expressions can be used to define attributes, facts, transformations, and metrics. These expressions provide unlimited functionality in supporting any schema available.

Fact Allocations

Fact allocations allow a fact to be reported at a level lower than the level at which it is stored. For example, a business unit's budget might be stored at the Quarter level. However, a user might want to report on budget at the Month level, with each month's allocation being one third of the quarter total. Although a new table could be built in the data warehouse to represent this data, that approach adds time to the batch load, and requires additional space in the data warehouse. A fact allocation defined in the metadata using MicroStrategy Architect can calculate a fact expression for Budget at the Month level as "quarterly budget"/ 3.

Fact Extensions

Fact extensions allow facts to be reported on levels at which they are not stored. For example, the Unit Cost of a product may vary only by product but remain the same across all stores. A report computing the total cost for a specific store must use the Unit Cost fact at the Store level even though the fact is not stored at the store level. A fact extension defined in MicroStrategy Architect allows the unit cost metric to be computed at the store level by extending the unit cost from the product level to the number of specific products sold at that store.

Fact allocations and extensions eliminate unnecessary data storage, and reduce the amount of time needed for data extract, transform, and load processes. Without them, the data warehouse would need to contain additional tables to enable some types of reporting. By defining a fact allocation or extension in MicroStrategy, the SQL for a report can derive the necessary data rather than requiring it in the data warehouse.

Logical Tables

Logical tables allow architects to create data structures different from the underlying table structures in the database by typing in a SQL statement to create a logical table in MicroStrategy. This is similar to the creation of database views, but these logical tables are not created in the database, but are stored in the MicroStrategy metadata.

Logical tables allow architects to deal with complex data modeling issues such as recursive hierarchies which may be flattened, and slowly changing dimensions.

Partition Tables

Partition tables are used to split up a very large table into smaller tables, called partitions, to improve query processing time. The information stored in each table partition can be defined either in the MicroStrategy metadata repository, or in a mapping table in the database. There are advantages to using the metadata repository. For example, the partition table can store different levels of information. Sales data for the current year can be stored at the daily level, while previous historical data can be stored at the monthly level. This is not possible if the partitions are managed by the database.

Inner and Outer Joins

Inner and outer joins relate to how data is displayed in reports in MicroStrategy, although SQL query generation will include the syntax for inner, full outer, left outer or right outer joins when necessary. Joins can be managed at the metric level or at the report level. An inner join will only display data that have metric values i.e. there will be no null values in the report. An outer join display all of the data including rows that have null values.

Data Internationalization

MicroStrategy supports the internationalization of your data into the languages required for your users. This allows data to be displayed in various languages that can reflect the user's language preferences without

needing to define separate reports or metadata objects for each of the different languages. Translated data can be stored in separate databases, in separate tables in the same database, in separate columns in the same table, or a combination of these techniques.

Freeform SQL Editor

Users can create reports against operational databases and other databases with schema not optimal for OLAP reporting by using fixed SQL statements. Users can copy/paste or type in any ANSI 92-compliant SQL statement into the freeform SQL editor. Multi-pass SQL statements can be used if the ODBC driver supports this. Users can reference existing stored procedures, define user prompts, and also insert security filters into these SQL reports.

Query Builder

The Query Builder is another way to access databases with a fixed SQL query but without needing to write the SQL statement. Simply pick the tables from a list of tables in the database, identify the columns that join the tables together, define the filtering criteria to apply, and choose the columns needed for the query. MicroStrategy will automatically generate a SQL statement to extract the data from the chosen tables. Prompts and security filters can be added to Query Builder reports.

XQuery Reports

MicroStrategy can report against Web services and XML data using xQuery statements, a language for processing XML data. Any Web service that uses the REST architecture or SOAP protocol can be accessed by using the MicroStrategy Freeform Query Editor to write XQuery statements. MicroStrategy provides an xQuery Editor and Generator (XEG) to automatically create an xQuery statement to query a given Web Services data source.

Hadoop Reports

MicroStrategy can report against Apache Hadoop distributions using HiveQL and Pig Latin statements. MicroStrategy connects to Hadoop using ODBC drivers and submits either generated HiveQL statements from a Hive schema definition, or HiveQL or Pig Latin freeform query statements typed or copied into the freeform query editor.

OLAP Cube Catalog

Designers can create reports against multi-dimensional cube databases such as SAP BW using MicroStrategy's dynamic MDX engine. The OLAP cube catalog allows users to browse the list of available cubes in the cube database, and select the ones that may be exposed to users for use in reporting.

Automatic Metadata Mapping

Once a cube has been selected in the OLAP cube catalog, the MicroStrategy MDX engine automatically translates the cube metadata into corresponding MicroStrategy metadata. For example, SAP characteristics are converted to MicroStrategy attributes, variables to MicroStrategy prompts, and key figures to metrics.

Though the translation process is automatic, developers can edit the names and properties of the translated metadata elements to customize them. Architects can also specify join properties between various data sources. For example, if Customer name information is stored both in SAP BW as well as the data warehouse, architects can specify a common attribute for both. Once a common attribute has been specified, users can join data from these multiple data sources along the conforming dimensions.

APPENDIX B: VERY LARGE DATABASE (VLDB) PROPERTIES

VLDB properties are settings that customize the SQL and MDX generated by MicroStrategy Intelligence Server. The settings provide a way to manipulate SQL join patterns, SQL insert statements, and table creation properties without manually altering SQL Scripts as well as how the Analytical Engine manages certain results. By adjusting the SQL and MDX statements using VLDB properties, specific optimizations can further enhance the performance of queries, and also provide the capability to easily incorporate and take advantage of new RDBMS features introduced in new versions. VLDB properties can be set at different levels within a MicroStrategy project. The levels are used in the following sequence of decreasing priority:

- Applied only when a specific report or Intelligent Cube is run (Report & Intelligent Cube)
- Every time a template is used with any filtering criteria (Template)
- Whenever processing a metric regardless of its use in reports, filters and/or other metrics (Metric & Transformation)
- For all queries accessing a specific database connection (Database Instance)
- Globally across the project (Project)

The table below lists the VLDB properties that are available to a developer.

Category	Property	Description	Level Available
Analytical Engine	Custom Group Display for Joint Elements	Control which attribute elements to display in joint element lists, first attribute only or all attributes	Project
	Display Null On Top	Control the display null values at the top or the bottom of a list when sorting data	Report, Template, Project
	Evaluation Ordering	Control whether to use the default or user-specified evaluation order	Report, Intelligent Cube, Template, Project
	Metric Level Determination	Select only lowest-level attributes or include higher-level attributes in metric level	Report, Project
	Null Checking For Analytical Engine	Enable or disable checking for null values in the Analytical Engine	Report, Intelligent Cube, Template, Metric, Project
	Subtotal Dimensionality Aware	Control whether the Analytical Engine takes into account the dimensionality of the subtotal calculation	Report, Template, Metric, Project
	Subtotals Over Consolidations Compatibility	Control the evaluation of consolidations by their elements only or by the elements and the underlying attributes	Report, Template, Project

Category	Property	Description	Level Available
Dynamic Sourcing	Aggregate Table Validation	Define whether aggregate tables contain the same amount of data as detail tables	Report, Template, Project
	Attribute Validation	Define the relationship between fact and lookup tables and the presence of NULL values	Project
	Enable Cube Parse Log in SQL View	Enable or disable the cube parse log in the SQL View	Intelligent Cube, Template, Project
	Enable Dynamic Sourcing for Report	Enable or disable dynamic sourcing with Intelligent Cubes	Report, Template, Project
	Enable Extended Mismatch Log in SQL View	Enable or disable the extended cube mismatch log in the SQL View	Report, Template, Project
	Enable Mismatch Log in SQL View	Enable or disable the cube mismatch log in the SQL View	Report, Template, Project
	Enable Report Parse Log in SQL View	Enable or disable the report parse log in the SQL View	Report, Template, Project
	Metric Validation	Enable or disable dynamic sourcing for metrics	Metric, Project
	String Comparison Behavior	Control the string comparison with dynamic sourcing - case insensitive or disabled	Project
Freeform SQL	Ignore Empty Result for Freeform SQL	Control SQL messages for queries that do not return results e.g. UPDATE statements	Database Instance
	XQuery Success Code	Define the expected XQuery success code	Database Instance
Governing	Maximum SQL/MDX Size	Maximum size in bytes of the SQL or MDX that can be accepted by the ODBC/BAPI/XMLA driver	Database Instance
	Intermediate Row Limit	Set the maximum number of rows allowed in intermediate tables	Report, Intelligent Cube
	Results Set Row Limit	Set the maximum number of rows allowed in report results	Report, Intelligent Cube
	SQL Time Out (Per Pass)	Time out in seconds for each pass of SQL	Report, Intelligent Cube, Database Instance
Indexing	Allow Index On Metric	Control whether indexes can be created on metric columns	Report, Intelligent Cube, Template, Database Instance
	Index Post String	A string that will be appended at the end of CREATE INDEX statement.	Report, Intelligent Cube, Template, Database Instance
	Index Prefix	Prefix to use when automatically creating indexes for intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Index Qualifier	String to insert between the CREATE and INDEX keywords.	Report, Intelligent Cube, Template, Database Instance
	Intermediate Table Index	Determine whether and when to create index for intermediate table.	Report, Intelligent Cube, Template, Database Instance
	Max Columns in Column Placeholder	Define the maximum number of columns in the column placeholder	Report, Intelligent Cube, Template, Database Instance
	Max Columns in Index	Determine the maximum number of columns that could be included in partition key or index.	Report, Intelligent Cube, Template, Database Instance
	Primary Index Type	Define whether to use primary index or partitioning key on temporary tables	Report, Intelligent Cube, Template, Database Instance
	Secondary Index Order	Control whether to create index before or after inserting data into the temporary table	Report, Intelligent Cube, Template, Database Instance
	Secondary Index Type	Control whether to create a compound index or separate indexes for attributes in a temporary table	Report, Intelligent Cube, Template, Database Instance

Category	Property	Description	Level Available
Joins	Attribute to Join when Key from Neither Side can be Supported from Other Side	Control whether to join on the common key only or on all common attributes	Report, Intelligent Cube, Template, Database Instance
	Base Table Join for Template	Control whether to use temp tables or fact table join when retrieving facts from different fact tables	Report, Intelligent Cube, Template, Database Instance
	Cartesian Join Evaluation	Control whether to reevaluate Cartesian joins in SQL passes	Report, Intelligent Cube, Template, Database Instance
	Cartesian Join Warning	Actions to take when a report contains a Cartesian join.	Report, Intelligent Cube, Template, Database Instance
	Downward Outer Join Option	Set the outer join behavior for metrics that are calculated at a level higher than the report or Intelligent Cube	Report, Intelligent Cube, Template, Database Instance
	DSS Star Join	Set the star join behavior to off or partial	Report, Intelligent Cube, Template, Database Instance
	From Clause Order	Control the order of the tables in the FROM clause	Report, Intelligent Cube, Template, Database Instance
	Full Outer Join Support	Set full outer join support on or off	Report, Intelligent Cube, Template, Database Instance
	Join Type	Set the ANSI standard syntax for joining tables (89 or 92/99)	Report, Intelligent Cube, Template, Database Instance
	Lookup Table in Outer Join	Set the order of the lookup tables when joining tables	Report, Intelligent Cube, Template, Database Instance
	Max Tables in Join	Set the maximum number of table allowed in the join	Database Instance
	Max Tables in Join Warning	Determine the behavior when the maximum number of tables in the join is exceeded	Report, Intelligent Cube, Template, Database Instance
	Nested Aggregation Outer Join	Control outer joins when performing nested aggregations	Report, Intelligent Cube, Template, Database Instance
	Preserve All Final Pass Result Elements	Control whether to include all elements or only the common elements in the final pass	Report, Intelligent Cube, Template, Database Instance
	Preserve All Lookup Table Elements	Determine which columns should be preserved when combining the intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
MDX	Format for Date/Time Values	Define the format of the date and time values coming from the MDX data source	Report, Intelligent Cube, Database Instance, Project
	MDX Add Fake Measure	Control whether to add a measure if the report only contains attributes	Report, Intelligent Cube, Database Instance
	MDX Add Non Empty	Control whether to add the non-empty keyword to the MDX query	Report, Intelligent Cube, Database Instance
	MDX Cell Formatting	Define if metric values are formatted by column or by cell	Report, Intelligent Cube, Database Instance
	MDX Level Number Calculation Method	Define whether to use the actual level number of a generation number	Report, Intelligent Cube, Database Instance
	MDX Non Empty Optimization	Set the non-empty optimization off, or on the default, first or all measures	Report, Intelligent Cube, Database Instance
	MDX TopCount Support	Set whether to use TopCount, or Rank and Order	Report, Intelligent Cube, Database Instance
	MDX Verify Limit Filter Literal Level	Set whether to verify the level of literals in limit or filter expressions	Report, Intelligent Cube, Database Instance

Category	Property	Description	Level Available
Metrics	Absolute Non-Agg Metric Query Type	Control whether to use a subquery or temporary tables when calculating absolute non-aggregatable metrics	Report, Intelligent Cube, Template, Database Instance
	Compute Non-Agg before/after OLAP Function Calculated in Analytical Engine	Set non-aggregatable metrics to calculate before or after OLAP functions or RANK	Report, Intelligent Cube, Template, Database Instance
	COUNT Compound Attribute	Set COUNT of compound attributes in a single expression or as a separate SQL pass	Database Instance
	COUNT(column) Support	Calculate COUNT metric as a count on the column or as COUNT(*)	Database Instance
	Data Type	Define the data type to use for the metric in SQL statements	Metric
	Default to Metric Name	Determine whether to use the metric name as the column alias in the SQL query	Database Instance
	Integer Constant In Metric	Apply a decimal to integer values that are used in calculations to prevent datatype conversions of decimals to integers	Metric, Database Instance
	Max Metric Alias Size	Set the maximum length of a metric alias name	Database Instance
	Metric Column Alias	Define an alias to use for a metric in SQL statements	Metric
	Metric Join Type	Define whether to use inner or outer joins when evaluating a metric.	Metric, Database Instance
	Non-Agg Metric Optimization	Control fact table access with issue multipass SQL for non-aggregatable metrics - less fact table access or smaller temp tables	Report, Intelligent Cube, Template, Database Instance
	Null Check	Indicates how to handle null values in SQL statements	Report, Intelligent Cube, Template, Metric, Database Instance
	Separate COUNT DISTINCT	Determine how COUNT DISTINCT is generated when more than one is needed from one fact table	Database Instance
Pre/Post SQL Statements	Transformable AggMetric	Define metrics that should be used to perform transformations on compound metrics that use nested aggregation	Metric
	Transformation Role Processing	Choose whether to apply transformation processing to all attributes or only to highest level child attribute applicable to multiple attributes	Transformation
	Zero Check	Indicates how to handle division by zero checking in SQL statements	Report, Intelligent Cube, Template, Metric, Database Instance
	Cleanup Post Statement	Adds a SQL statement/comment at the end of the generated SQL	Report, Intelligent Cube, Template, Database Instance
	Datamart SQL to be Executed After Datamart Creation	Adds a SQL statement/comment at the end of the generated SQL for a datamart	Database Instance
	Datamart SQL to be Executed Before Inserting Data	Adds a SQL statement/comment before the INSERT statement of a datamart	Database Instance
Data Sources	Datamart SQL to be Executed Prior to Datamart Creation	Adds a SQL statement/comment at the beginning of the generated SQL for a datamart	Database Instance
	Drop Database Connection	Control whether to drop the database connection after running the user-defined SQL statements	Report, Intelligent Cube, Template, Database Instance
	Insert Mid Statement	Adds a SQL statement/comment after each individual insert statement	Report, Intelligent Cube, Template, Database Instance

Category	Property	Description	Level Available
Pre/Post SQL Statements	Insert Post Statement	Adds a SQL statement/comment after the final insert statement	Report, Intelligent Cube, Template, Database Instance
	Insert Pre Statement	Adds a SQL statement/comment before the final insert statement	Report, Intelligent Cube, Template, Database Instance
	Report Post Statement	Adds a SQL statement/comment prior to dropping the first intermediate table	Report, Intelligent Cube, Template, Database Instance
	Report Pre Statement	Adds a SQL statement/comment prior to creating the first intermediate table	Report, Intelligent Cube, Template, Database Instance
	Table Post Statement	Adds a SQL statement/comment after creating each intermediate table	Report, Intelligent Cube, Template, Database Instance
	Table Pre Statement	Adds a SQL statement/comment prior to creating each intermediate table	Report, Intelligent Cube, Template, Database Instance
Query Optimizations	Additional Final Pass Option	Control whether the final SQL pass can perform aggregations and lookups or whether a separate pass is needed	Report, Intelligent Cube, Template, Database Instance
	Apply Filter Options	Determine whether to apply filter conditions to intermediate SQL statements and/or the final SQL statement	Report, Intelligent Cube, Template, Database Instance
	Count Distinct With Partitions	Control whether or not to select distinct elements from each table partition	Report, Intelligent Cube, Template, Metric, Database Instance
	Custom Group Banding Count Method	Define how to define custom groups when using the count method - use CASE, insert band into database and join, normal calculation	Report, Template, Database Instance
	Custom Group Banding Points Method	Define how to define custom groups when using the points method - use CASE, insert band into database and join, normal calculation	Report, Template, Database Instance
	Custom Group Banding Size Method	Define how to define custom groups when using the size method - use CASE, insert band into database and join, normal calculation	Report, Template, Database Instance
	Data Population for Intelligent Cubes	Determine how to populate data in Intelligent Cubes - normalize in cube or database, direct load, no normalization	Intelligent Cube, Database Instance
	Data Population for Reports	Determine how to populate results for reports - normalize in cube or database, no normalization	Report, Template, Database Instance
	Dimensionality Model	Generate SQL according to the relational model or dimensional model	Report, Intelligent Cube, Template, Database Instance
	Engine Attribute Role Options	Enable or disable the feature that treats attributes defined on the same column with the same expression as attribute roles	Database Instance
	MD Partition PreQuery Option	Choose whether to use a COUNT or a constant when initially querying partition tables	Report, Intelligent Cube, Template, Database Instance
	Multiple Data Source Support	Choose the method to access multiple data sources - MultiSource Option or database gateway	Database Instance, Project
	OLAP Function Support	Determine how metrics that use OLAP functions are calculated for backwards compatibility	Report, Intelligent Cube, Template, Database Instance
	Rank Method If DB Ranking Not Used	Define whether ranking calculations are performed by ODBC ranking or Analytical Engine if the database does not support ranking	Report, Intelligent Cube, Template, Database Instance
	Remove Aggregation Method	Determines whether to keep or remove aggregations in SQL queries	Report, Intelligent Cube, Template, Database Instance
	Remove Group By Option	Determines whether GROUP BY and aggregations are used for attributes with the same primary key	Report, Intelligent Cube, Template, Database Instance

Category	Property	Description	Level Available
Query Optimizations	Remove Repeated Tables for Outer Joins	Determines whether an optimization for outer join processing is enabled or disabled	Report, Intelligent Cube, Template, Database Instance
	Set Operator Optimization	Enable or disable the use of set operators in sub queries to combine multiple filter qualifications	Report, Intelligent Cube, Template, Database Instance
	SQL Global Optimization	Determines the level by which reports are optimized to reduce the number of SQL passes	Report, Intelligent Cube, Template, Database Instance
	Sub Query Type	Control if EXISTS, IN or intermediate tables are used when performing subqueries	Report, Intelligent Cube, Template, Database Instance
	Transformation Formula Optimization	Defines whether to improve performance of reports that use expression-based transformations	Report, Intelligent Cube, Template, Database Instance
	Unrelated Filter Options	Determines whether the Analytical Engine should keep or remove the unrelated filter	Report, Intelligent Cube, Template, Database Instance
	Unrelated Filter Options for Nested Metrics	Determines whether the Analytical Engine should keep or remove the unrelated filters when using nested metrics	Report, Intelligent Cube, Template, Database Instance
	WHERE Clause Driving Table	Determine whether to apply filters to the lookup tables or to the fact tables	Report, Intelligent Cube, Template, Database Instance
Select/Insert	Attribute Form Selection Option For Intermediate Pass	Choose to select only the Attribute ID or all Attribute forms on the report in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Attribute Selection Option For Intermediate Pass	Choose to select the Attribute ID at the level of the report or all other Attribute ID levels in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Bulk Insert String	Defines the SQL prefix string for bulk insert statements	Report, Intelligent Cube, Template, Database Instance
	Constant Column Mode	Choose whether to use and the construction of GROUP BY when working with a column that is a constant	Report, Intelligent Cube, Template, Database Instance
	Custom Group Interaction with Report Filter	Defines how a report filter interacts with a custom group	Database Instance
	Datamart Column Order	Determines the order in which datamart columns are created	Report, Intelligent Cube, Template, Database Instance
	Date Format	Define the format for Date data	Report, Intelligent Cube, Template, Database Instance
	Date Pattern	Define the syntax pattern for Date data	Report, Intelligent Cube, Template, Database Instance
	Decimal Separator	Choose period or comma as the decimal separator	Report, Intelligent Cube, Template, Database Instance
	Default Attribute Weight	Determine if attributes not in the attribute weights list are treated as higher or lower than for those that are in the list	Database Instance
	Disable Prefix in WH Partition Table	Choose whether or not to use the table prefix in partition queries	Report, Intelligent Cube, Template, Database Instance
	Distinct/Group By Option (when no aggregation and not table key)	Choose whether to use DISTINCT, GROUP BY or none of these when selecting information that is not being aggregated	Report, Intelligent Cube, Template, Database Instance
	GROUP BY ID Attribute	Determines how to generated GROUP BY when using expressions	Report, Intelligent Cube, Template, Database Instance
	GROUP BY Non-ID Attribute	Choose whether to use MAX or GROUP BY when retrieving non-ID Attribute forms	Report, Intelligent Cube, Template, Database Instance
	Insert Post String	Adds a SQL string to the end of INSERT statements when populating intermediate tables	Report, Intelligent Cube, Template, Database Instance

Category	Property	Description	Level Available
Select/Insert	Insert Table Option	Adds a SQL string immediately preceding the SELECT statement of INSERT statements when populating intermediate tables	Report, Intelligent Cube, Template, Database Instance
	Long Integer Support	Determines whether to map long integers of a certain length as BigInt data types	Intelligent Cube, Database Instance
	Max Digits in Constant	Sets the maximum number of digits in a constant literal in an insert values statement	Database Instance
	Merge Same Metric Expression Option	Choose whether to select a metric more than once if it is used more than once in a report	Report, Intelligent Cube, Template, Database Instance
	Select Post String	Adds a SQL string to the end of the SELECT statements in the final SQL pass	Report, Intelligent Cube, Template, Database Instance
	SQL Hint	Adds a SQL hint immediately after the word, SELECT, in all SQL passes	Report, Intelligent Cube, Template, Database Instance
	SQL Time Format	Define the format for Time data	Database Instance
	Timestamp Format	Enter the mask used for Timestamp datatypes in SQL statements	Report, Intelligent Cube, Template, Database Instance
	UNION Multiple INSERT	Choose whether to use UNION to populate an intermediate table or to use more than one INSERT statement	Report, Intelligent Cube, Template, Database Instance
Tables	Alias Pattern	Defines the SQL string used to alias columns	Database Instance
	Attribute ID Constraint	Adds a SQL string used as constraints to ID columns when creating temporary tables	Database Instance
	Character Column Option	Defines how to support multiple character sets	Database Instance
	Column Pattern	Used to alter the syntax pattern for column names.	Database Instance
	Commit After Final Drop	Define whether to issue COMMIT SQL statement after the final intermediate table has been dropped	Report, Intelligent Cube, Template, Database Instance
	Commit Level	Define whether to issue a COMMIT SQL statement after DDL instructions, after DML instructions, after both DDL and DML instructions or never	Report, Intelligent Cube, Template, Database Instance
	CREATE and INSERT Support	Enable or disable CREATE and INSERT statements for creating temporary tables	Database Instance
	Create Post String	Adds a SQL string to the end of the CREATE TABLE statements in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Drop Temp Table Method	Define whether to clean up the intermediate tables immediately or not	Report, Intelligent Cube, Template, Database Instance
	Fallback Table Type	Choose whether to use true temporary tables or permanent tables for fallback tables	Report, Intelligent Cube, Template, Database Instance
	Hexadecimal Character Transformation	Allows string characters to be converted into specific character encoding required for some Unicode implementations	Database Instance
	Intermediate Table Type	Choose whether to use true temporary tables, derived tables, common table expressions, temporary views or permanent tables for intermediate tables	Report, Intelligent Cube, Template, Database Instance
	Maximum SQL Passes Before Fallback	Determines how many SQL passes are allowed for temporary tables before reverting to the Fallback Table Type	Report, Intelligent Cube, Template, Database Instance
	Maximum Tables in FROM Clause Before Fallback	Determines how tables are allowed in the FROM clause for temporary tables before reverting to the Fallback Table Type	Report, Intelligent Cube, Template, Database Instance
	Table Creation Type	Define whether intermediate tables are created implicitly in a single SQL pass or explicitly in two SQL passes	Report, Intelligent Cube, Template, Database Instance

Category	Property	Description	Level Available
Tables	Table Descriptor	Adds a SQL string immediately after the words CREATE TABLE in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Table Option	Adds a SQL string immediately after the table name in the CREATE TABLE statements in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Table Prefix	Adds a prefix to the table name in the CREATE TABLE statements in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Table Qualifier	Adds a SQL string between the words CREATE and TABLE in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
	Table Space	Adds a SQL string immediately after the column list in the CREATE TABLE statements in intermediate SQL passes	Report, Intelligent Cube, Template, Database Instance
Report Data Options	Attribute Join Type	Define which attributes should use an inner or outer join against the other attributes on a template based on SQL passes and filtering criteria	Report, Template
	Evaluation Order	Determine the sequence in which calculations in Analytical Engine should be evaluated	Report, Intelligent Cube, Template
	Metric Join Type	Define which metrics should use an inner or outer join against the other metrics on a template	Report, Intelligent Cube, Template
	Report Limit	Applies additional filtering conditions to metrics after calculation of the metrics	Report, Intelligent Cube, Template

APPENDIX C: ANALYTICAL FUNCTIONS

Basic Functions

Add
Average of List of Arguments
Average of Values in a List
Count
First
Geometric Mean
Greatest
Last
Least
Maximum
Median
Minimum
Mode
Multiply a List of Arguments
Product of Values in a List
Standard Deviation of Population
Standard Deviation of Sample
Sum
Variance of Population
Variance of Sample

Date and Time Functions

Add Days to a Date
Add Month to a Date
Current Date
Current Datetime
Current Time
Date
Day of Month
Day of Week
Day of Year
Days Between
Hour
Millisecond
Minute
Month
Month End Date
Month Start Date
Months Between
Quarter
Second
Week
Year

Internal Functions

Apply Aggregation
Apply Comparison
Apply Freeform SQL Security Filter
Apply Logic
Apply OLAP
Apply Optional
Apply Simple
Banding based on Count
Banding based on Percent
Banding based on Size
Banding on Metric
Case
Case Value
Coalesce
Except
Intersect
Intersect In
Repeat
Tuple
Union

Null/Zero Functions

Is Not Null
Is Null
Null To Zero
Zero To Null

OLAP Functions

Exponential Weight Moving Average
Exponential Weight Running Average
First Value in Range
Last Value in Range
Lag
Moving Average
Moving Count
Moving Difference
Moving Maximum
Moving Minimum
Moving Standard Deviation of Population

Moving Standard Deviation of Sample

Moving Sum
OLAP Average using Moving Window
OLAP Count using Moving Window
OLAP Max using Moving Window
OLAP Min using Moving Window
OLAP Rank using Moving Window
OLAP Sum using Moving Window
Running Average
Running Count
Running Maximum
Running Minimum
Running Standard Deviation of Population
Running Standard Deviation of Sample
Running Sum
Running Total

Rank and NTile Functions

N-Tile
N-tile by Step
N-tile by Value
N-tile by Step and Value
Percentile
Rank

String Functions

Concatenate
Concatenate with a space
Initial Capitalization
Left String
Left Trim
Length
Lower
Position
Right String
Right Trim
SubString
Trim
Upper

Mathematical Functions		
Absolute Value	Growth of Single Value	Coupon Period, Days from Settlement to Next Coupon
Arc cosine	Heteroscedastic Ttest	Cumulative Interest Paid on Loan
Arc cosine hyperbolic	Homoscedastic Ttest	Cumulative Principal Paid on Loan
Arc sine	Hypergeometric Distribution	Discount Rate for a Security
Arc sine hyperbolic	Intercept	Double-Declining Balance Method
Arc tangent	Inverse of Beta Distribution	Effective Annual Interest Rate
Arc tangent of x- and y-coordinates	Inverse of Chi-Square Distribution	Fixed-Declining Balance Method
Arc tangent hyperbolic	Inverse of F Probability Distribution	Future Value
Ceiling	Inverse of Fisher Transformation	Future Value Schedule
Combine	Inverse of Gamma Distribution	Interest Rate
Cosine	Inverse of Lognormal Cumulative Distribution	Interest Payment
Cosine hyperbolic	Inverse of the Normal Cumulative Distribution	Internal Rate of Return
Degrees	Inverse of the Standard Normal Cumulative Standard	Interest Rate per Period
Exponent	Inverse of T-Distribution	Macaulay Duration
Factorial	Kurtosis	Modified Duration
Floor	Lognormal Cumulative Distribution	Modified Internal Rate of Return
Integer	Mean	Net Present Value
Log	Mean T-Test	Next Coupon Date After
Log Base 10	Negative Binomial Distribution	Settlement Date
Modulus	Normal Cumulative Distribution	Nominal Annual Interest Rate
Natural Log	Paired T-test	No. of Coupons Between
Power	Pearson Product Moment Correlation Coefficient	Settlement and Maturity
Quotient	Permutation	Number of Investment Periods
Radians	Poisson Distribution	Odd First period Price
Random Number Between	RSquare	Odd First period Yield
Round	Skew	Odd Last Period Price
Round with Precision	Slope of Linear Regression	Odd Last Period Yield
Sine	Standardize	Payment
Sine hyperbolic	Standard Normal Cumulative Distribution	Payment Principal
Square Root	Standard Error of Estimates	Previous Coupon Date Before
Tangent	Trend of Group Value	Settlement Date
Tangent hyperbolic	Trend of Singe Value	Price Per \$100 Face Value
Truncate	T-Distribution	Price, Discounted
	Variance	Price at Maturity
	Variance Test	Present Value
	Weibull Distribution	Straight Line Depreciation
Statistical Functions		
Average Deviation		Sum-Of-Years' Digits Depreciation
Beta Distribution		Treasury Bill Equity
Binomial Distribution		Treasury Bill Price
Chi-Square Distribution		Treasury Bill Yield
Chi-Square Test		Variable Declining Balance
Confidence Interval	Accrued Interest	Yield
Correlation Coefficient	Accrued Interest Maturity	Yield for Discounted Security
Covariance	Amount Received at Maturity	Yield at Maturity
Criterion Binomial Distribution	Convert Dollar Price from Decimal to Fraction	Financial Functions
Exponential Distribution	Convert Dollar Price from Fraction to Decimal	
Fisher Transformation	Coupon Period, Days Beginning to Settlement	
Forecast of Group Value	Coupon Period, Days with Settlement	
Forecast of Single Value		Data Mining Functions
F-Probability Distribution		
F-Test		
Gamma Distribution		
Growth of Group Value		

Mining Model, Non-Numeric	Arithmetic Operators	Not Between
Neural Network, Numeric	Addition	Not Contains
Neural Network, Non-Numeric	Division	Not Ends With
Regression, Numeric	Multiplication	Not Equal
Regression, Non-Numeric	Subtraction	Not In
Ruleset, Numeric	Unary Minus	Not Like
Ruleset, Non-Numeric		
Support Vector Machine, Numeric	Comparison Operators	Comparison for Rank Operators
Support Vector Machine, Non-Numeric	Begins With	Between
Time Series	Between	Equal
Train Association Model	Contains	Greater Than or Equal
Train Clustering Model	Ends With	Less Than or Equal
Train Decision Tree Model	Equal	Not Between
Train Regression Model	Greater Than	Not Equal
Train Regression Model with Tree	Greater Than or Equal	
Train Time Series	In	Logical Operators
Tree Model, Numeric	Less Than	And
Tree Model, Non-Numeric	Less Than or Equal	If
	Like	Not
	Not Begins With	Or

Note that R scripts and charts can be embedded into the MicroStrategy platform and displayed in MicroStrategy Web.

APPENDIX D: SUPPORTED GRAPH TYPES

Visualizing data is a key requirement in modern BI products. MicroStrategy approaches this in 3 ways:

1. Advanced visualizations that are easy to read with interactive features in MicroStrategy Mobile and MicroStrategy Web
2. Data discovery visualizations to quickly spot outliers for further analysis in Visual Insight
3. Standard graphs and charts for traditional uses, such as printing, in all MicroStrategy user interfaces

Advanced Visualizations

Visualization	Web		Mobile		
	DHTML	Flash	iPhone	iPad	Android
Area Mashup	✓				
Bubble Grid	✓	✓			
Cylinder		✓			
Data Cloud	✓	✓	✓	✓	
Date Selection	✓	✓		✓	
Fisheye	✓	✓			
Funnel	✓	✓			
Gauge	✓	✓			
Google Graph	✓				
Google Map	✓	✓	✓	✓	✓
Google Map + Network Visualization		✓	✓	✓	✓
Graph Matrix		✓		✓	
Graph Matrix (deprecated)	✓	✓			
Heat Map	✓	✓		✓	
Image Map	✓				
Image Viewer			✓	✓	✓
Interactive Bubble Graph	✓	✓			
Interactive Grid			✓	✓	✓
Interactive Stacked Graph		✓			
Line Chart	✓				
Marker Mashup	✓				
Media	✓	✓			
Microcharts	✓	✓		✓	
Multimedia Widget			✓	✓	
Photo Uploader			✓	✓	✓
RSS Reader	✓	✓	✓	✓	
Store Layout	✓				
Table	✓				
Thermometer		✓			
Time Series Slider		✓	✓	✓	✓
Timeline	✓			✓	
USA Map	✓				
Waterfall	✓	✓			
Weighted List Viewer	✓	✓			
What-If Control	✓				

Data Discovery Visualizations

Graph Matrix
Google Map Visualization
Heat Map

Standard Graphs and Charts

3D Surface — Standard	Bar — Vertical absolute	Line — Vertical percent
3D Surface — Standard with sides	Bar — 3D rectangular riser	Line — 3D group
3D Surface — Honeycomb	Bar — 3D columnar riser	Line — 3D series
Area — Horizontal absolute	Bar — 3D conical riser	Pareto — Standard
Area — Horizontal stacked	Bar — 3D floating cube	Pareto — Percent
Area — Horizontal bipolar absolute	Bar — 3D floating sphere	Pie — Standard
Area — Horizontal bipolar stacked	BoxPlot — Box and whiskers	Pie — Ring
Area — Horizontal dual axis absolute	Bubble — Standard	Pie — Multiple
Area — Horizontal dual axis stacked	Bubble — Dual axis	Pie — Multiple ring
Area — Horizontal percent	Combination — Bar and area	Pie — Multiple proportional
Area — Vertical absolute	Combination — Bar and line	Pie — Multiple proportional ring
Area — Vertical stacked	Combination — Area and line	Pie — 3D basic
Area — Vertical bipolar	Combination — Dual axis bar and area	Pie — 3D ring
Area — Vertical bipolar stacked	Combination — Dual axis bar and line	Pie — 3D multiple
Area — Vertical dual axis absolute	Combination — dual axis area and line	Pie — 3D multiple ring
Area — Vertical dual axis stacked	Funnel — Standard	Pie — 3D multiple proportional
Area — Vertical percent	Gantt — Standard	Pie — 3D multiple proportional ring
Area — 3D group	Gauge — Standard	Polar — Coordinate
Area — 3D series	Histogram — Horizontal	Polar — Dual axis coordinate
Bar — Horizontal side by side	Histogram — Vertical	Radar — Line
Bar — Horizontal stacked	Line — Horizontal absolute	Radar — Stacked line
Bar — Horizontal dual axis side by side	Line — Horizontal stacked	Radar — Dual axis line
Bar — Horizontal dual axis stacked	Line — Horizontal bipolar absolute	Radar — Dual axis stacked line
Bar — Horizontal bipolar side by side	Line — Horizontal bipolar stacked	Radar — Area
Bar — Horizontal bipolar stacked	Line — Horizontal dual axis absolute	Radar — Stacked area
Bar — Horizontal percent	Line — Horizontal dual axis stacked	Radar — Dual axis area
Bar — Horizontal absolute	Line — Horizontal percent	Radar — Dual axis stacked area
Bar — Vertical side by side	Line — Vertical absolute	Scatter — X-Y
Bar — Vertical stacked	Line — Vertical stacked	Scatter r — X-Y
Bar — Vertical dual axis side by side	Line — Vertical bipolar absolute	Scatter — Dual axis X-Y
Bar — Vertical dual axis stacked	Line — Vertical bipolar stacked	Scatter — X-Y-Z
Bar — Vertical bipolar side by side	Line — Vertical dual axis absolute	Stock — Hi Lo
Bar — Vertical bipolar stacked	Line — Vertical dual axis stacked	Stock — Hi Lo Close
Bar — Vertical percent		Stock — Hi Lo open close

Note that R scripts and charts can be embedded into the MicroStrategy platform and displayed in MicroStrategy Web.

APPENDIX E: PERFORMANCE COUNTERS AND KEY PERFORMANCE INDICATORS

Intelligence Server Job Counters		Intelligence Server User Counters
Average report queue time	Total cached memory in the server process heap	Average user session length
Average report response time	Total completed 4-tier report manipulations	Invalid login attempts
Element browse submission rate	Total completed documents	Open project sessions
Element server cache	Total completed Intelligent Cube reports	Open sessions
Executing reports	Total completed Intelligent Cubes	Total project sessions
Largest free memory region	Total completed reports	Total sessions
Memory for Excel exporting	Total completed Web reports	Total timed out users
Memory for Flash exporting	Total document cache index size	Working Set Cache RAM Usage
Memory for HTML exporting	Total document response time	
Memory for PDF exporting	Total documents	
Memory for Plain Text exporting	Total element browse requests	
Memory for XML generation	Total error documents	
Memory used by report cache (MB)	Total error element browse requests	
Number of active Analytical Engine child processes	Total error object browse requests	
Number of active DB child processes	Total error reports	
Number of active DS child processes	Total in-use memory in server process heap	
Number of document cache swaps	Total Intelligent Cube report response time	
Number of document caches in memory	Total Intelligent Cube response time	
Number of Intelligent Cube cache swaps	Total local cube cache size	
Number of Intelligent Cube caches in memory	Total local document cache size	
Number of local document caches	Total local report cache size	
Number of local Intelligent Cube caches	Total MCM denials	
Number of other active child processes	Total MSI Table memory usage	
Number of report cache swaps	Total number of document cache hit	
Number of report caches in memory	Total object browse requests	
Object browse submission rate	Total report cache checks	
Object Server cache	Total report cache hits	
Open reports	Total report cache index size	
Report cache hit ratio	Total report requests	
Report submission rate	Total report response time	
Report completion rate	Total requests	
SmartHeap fragmentation	Virtual bytes fragmentation	
Timed-out jobs		

Document execute total counts	Resolution server elapse time	COM Marshal Slave
Document formatting CPU time	Resolution server total counts	Common Dialogs
Document formatting elapse time	SQL engine server CPU time	Common Editor Controls
Document formatting total counts	SQL engine server elapse time	Common Utility Classes
Document manipulation CPU time	SQL engine server total counts	Connection Mapping
Document manipulation elapse time	Update cube CPU time	Consolidation Editor
Document manipulation total counts	Update cube elapse time	Cube Server
Document sender CPU time	Update cube total counts	Custom SQL Dialog
Document sender elapse time	Update report cache CPU time	Custom SQL Lib
Document sender total counts	Update report cache elapse time	Data Slice Editor
Element server CPU time	Update report cache total counts	Database Classes
Element server elapse time	Intelligence Server Object Instance	Database Instance Wizard
Element server total counts	Total chart objects	Database Module
Error message sender CPU time	Total command server context objects	Datamart Configuration
Error message sender elapse time	Total document definition objects	Datamart Executor
Error message sender total counts	Total document instance objects	Date Lib
Export engine CPU time	Total job objects	DB Connection Configuration
Export engine elapse time	Total job task objects	DB Element Server
Export Engine total counts	Total message objects	DB Role Configuration
Find cube CPU time	Total process context objects	Desktop
Find cube elapse time	Total report definition objects	Desktop Actions
Find cube total counts	Total report instance objects	Desktop Properties
Find report cache CPU time	Total server message objects	Desktop Properties Dialog
Find report cache elapse time	Total SQL objects	DFC DateTime
Find report cache total counts	Total XML context objects	Diagnostics Configuration
Get report instance CPU time	Total XML document objects	Dimensional Storage
Get report instance elapse time	Total XML node objects	DMX
Get report instance total counts	Total user session objects	Document Server
Object server CPU time	Total workspace objects	Drill Map Editor
Object server elapse time		Drill Map Editor Tool
Object server total counts		DSS Components
Output message sender CPU time	Intelligence Server	Dynamic Sourcing
Output message sender elapse time	Diagnostic Components	Editor Container
Output message sender total counts	Admin Editor Container	Editor Manager
Persist result CPU time	Agent	Editors Lib
Persist result elapse time	Analytical Engine	Editors Support Lib
Persist result total counts	Analytical Services	Element Net Client
Post processing CPU time	Asynch Lib	Element Net Server
Post processing elapse time	Authentication Server	Element Server
Post processing total counts	Calculation Server	Encoder
Query engine server CPU time	Catalog Net Server	Enterprise Manager
Query engine server elapse time	Catalog Server	Enterprise Manager ETL
Query engine server total counts	Catalog Source	Enterprise Manager Service
Report net server CPU time	Client Configuration	ETL Server
Report net server elapse time	Client Connection	ETL Support
Report net server total counts	Client Services	Events Editor
Request execute CPU time	Code Page	Explorer Lib
Request execute elapse time	Column Editor	Export Engine
Request execute total counts	COM	Export Lib
Resolution server CPU time	COM Helper	Expression Box Lib

File Transmitter	Pageby Lib	Source Net Client
Filter	Parser XGI	Source Server
Filter Lib	Partition Editor	State Logger
Formatting Dialog	Performance Monitor	State Monitor
Form Categories Editor	Performance Object	Synchronization Classes
FTR Container Lib	Performance Profiler	System Documentation Component
Function Server	Print Core	System Dimension
Functions Dialog	Print Graph Interface	Table Editor
GDI	Print Grid Interface	Thread Classes
Generic Holders	Print Schema	Transformation Editor
Generics	Print Transmitter	UI Control
GH Dialogs	Priority Map Editor	User Editor
Graph Viewer	Privileges Editor	User to Project Editor
Graph Lib	Process Manager	View Designer
Graph Themes Lib	Progress Indicator	View Lib
Grid Lib	Project Builder	VLDB Editor
Grid Support Lib	Project Builder Slides	Web SDK Bridge Base
GUI Locking Manager	Project Configuration	WH Catalog
GUI Object Lib	Project Creation Lib	Xerces COM Wrapper
GUI Shared Lib	Project Preferences	XMLA
Helper Controls Lib	Project Upgrade Lib	XQuery Wrapper
Helper Objects	Prompt Editors Lib	XTAB Factory
Hierarchy Editor	Prompt Ordering	
HTREE	Prompts Lib	
Kernel	Prompt Styles	
Kernel XML API	Properties Control Lib	
Logon Lib	Property Sheet Lib	
MA CPU	Query Engine	
MA CPUWin7	Registry Classes	
MA HTTP	Report Control	
MA Query Environment	Report Core Lib	
MA Reg Client	Report Data Options	
MD Translator Library	Report Data Options Lib	
MD Update	Report Drilling Lib	
Measure Editor Lib	Report Format Lib	
Metadata Server	Report Helper	
Merge User Utility	Report Net Client	
MTH Transformer	Report Net Server	
MIME	Report Server	
Mobile Transmitter	Report Sort Dialog	
Module Global Constants	Report Sort Lib	
MSI Graph	Report Subtotal Lib	
MSQL	Schema Manipulation	
Multi Process	Script Engine	
NCS Base	Search Editor Lib	
NCS Dialogs	Search Service	
NCS Lib	Security Role Editor	
Network	Security Role Viewer	
Network Classes	Security Subsystem	
Object Context	Server Admin	
Object Manager	Server Configuration	
Object Server	Server Control	
Objects Selector Lib	Server Control Plugin	
ODBC Connector	SMTP Sender	

Enterprise Manager Metrics

Jobs Aggregating
Jobs Filtered
Jobs Grouped By
Jobs Hitting Table
of Report-Attribute Relationships
of Report-Metric Relationships
Aggregation Probability
Average Connection Duration hh:mm:ss
Average Connection Duration hh:mm:ss (Intelligence Server)
Average Connection Duration per User hh:mm:ss
Average DP Average Number of Datasets per Job (Project)
Average DP Number of Jobs (Project)
Average DP Number of Jobs with Error (Project)
Average Execution Duration (Report)
Average Number of Active Sessions
Average Number of Active Users
Average Number of Recipients per Subscription
Average Number of Sessions (Days)
Average Number of Sessions (Project, Days)
Average Number of Users (Project, Days)
Average RP Export Engine Jobs (Project)
Average RP Number of Ad-Hoc Jobs (Project)
Average RP Number of Jobs (Project)
Average RP Number of Jobs with Cache Hit (Project)
Average RP Number of Jobs with Element Loading (Project)

Average RP Number of Jobs with Error (Project)	DP Average Number of Prompts (Project)	HL Number of Executions of Content
Average RP Number of Report Jobs from Document Execution (Project)	DP Average Number of Reports per Document	HL Number of Messages
Average RP Number of Result Rows (Project)	DP Average Queue Duration hh:mm:ss (Project)	HL Number of Messages with Errors
Average RP Number of Scheduled Jobs (Project)	DP Average Queue Duration per Job hh:mm:ss	HL Number of Read Messages
Average RP Number of SQL Passes (Project)	DP Elapsed Duration hh:mm:ss	HL Number of Report Jobs
Average RP Number of WH SQL Passes (Project)	DP Elapsed Duration hh:mm:ss (Project)	HL Number of Requests for Content
Average RP User Report Requests (Project)	DP Execution Duration hh:mm:ss	HL Time Since Last Action (Days)
Average Subscription Exec Duration hh:mm:ss	DP Execution Duration hh:mm:ss (Project)	HL Time Since Last Read (Days)
CM Delete Attributes	DP Execution Popularity	HL Time Since Last Request (Days)
CM Delete DB Instances	DP First Exec Finish Timestamp	Last 3 Months
CM Delete Documents	DP First Exec Request Timestamp	Last 6 Months
CM Delete Filters	DP First Exec Start Timestamp	Last Connect Date
CM Delete Metrics	DP Last Exec Finish Timestamp	Last Connect Timestamp
CM Delete Reports	DP Last Exec Request Timestamp	Last Disconnect Timestamp
CM Delete Schedules	DP Last Exec Start Timestamp	Last Month
CM Delete Security Filters	DP Number of Jobs	Last Subscription Exec Start
CM Delete Templates	DP Number of Jobs (Project)	Timestamp
CM Delete User	DP Number of Jobs w/o Error	Last Two Weeks
CM Delete User Groups	DP Number of Jobs with Cache Hit	Last Week
Column Access Probability	DP Number of Jobs with Error	Max Cache Update Subscription Count
Connect Date	DP Number of Prompted Jobs	Max Connection Duration
Connect Offset Duration	DP Percentage of Jobs with Cache Hit	hh:mm:ss
Connection Duration hh:mm:ss	DP Percentage of Jobs with Error	Max Email Subscription Count
Connection Duration hh:mm:ss (Intelligence Server)	DP Percentage of Jobs with Error	Max Executing Reports Counter
Connection Limit	DP Percentage of Jobs with Error (Project)	Max File Subscription Count
Day Date - Prompted	DP Queue Duration hh:mm:ss	Max History List Subscription Count
Disconnect Offset Duration	DP Queue Duration hh:mm:ss	Max Jobs Per Project
Distance to Mean	DP Queue Duration hh:mm:ss (Project)	Max Jobs Per Server
DP Average Elapsed Duration hh:mm:ss (Project)	DP User Count	Max Memory Used by Report Caches Counter
DP Average Elapsed Duration per Job hh:mm:ss	DRP Number of Jobs	Max Number of Active Sessions
DP Average Elapsed Duration per Job seconds	Executing Reports Counter	Max Number of Active Users
DP Average Execution Duration hh:mm:ss (Project)	Filtering Probability	Max Number of Report Caches Counter
DP Average Execution Duration per Job hh:mm:ss	First Connect Timestamp	Max Open Project Sessions Counter
DP Average Number of Datasets per Job	First Disconnect Timestamp	Max Open Sessions Counter
DP Average Number of Datasets per Job (Project)	Group By Probability	Max Print Subscription Count
DP Average Number of Jobs per Session	HIT_IS_SCHED_RELATE_IU	Max Report Cache Count
DP Average Number of Prompts	HL Average Number of Actions per Message	Max Report Cache Disk Space
	HL Last Action Date	Max Report Cache Memory Consumption
	HL Last Read Date	Max Report Execution Time
	HL Last Request Date	Max Used Private Bytes
	HL Number of Actions	Max Used Virtual Bytes
	HL Number of Actions with Errors	Max User Connection Per Project
	HL Number of Document Jobs	Max User Connection Per Server
		Max Warehouse Job Execution Time
		Median (Subscription Analysis Metric)

Median Session Duration	Number of Reports	RP Average Elapsed Duration per Web Job hh:mm:ss
Memory Used by Report Caches (KB) Counter	Number of Reports	RP Average Exec Duration per Job hh:mm:ss
Memory Used by Report Caches (MB) Counter	Number of Schedules	RP Average Exec Duration per Job hh:mm:ss (Project)
Min Connection Duration hh:mm:ss	Number of Security Filters	RP Average Exec Duration per Job msec
Min Number of Active Sessions	Number of Sessions	RP Average Intelligent Cube Size (KB)
Min Number of Active Users	Number of Sessions (Intelligence Server)	RP Average Number of Ad-Hoc Jobs per Report
Minute Level Session Concurrency	Number of Sessions (Project)	RP Average Number of Ad-Hoc Jobs per Session
Minute Level User Concurrency	Number of Subscriptions	RP Average Number of Ad-Hoc Jobs per User
Number of Attribute Forms	Number of Tables	RP Average Number of Cancelled Jobs per Report
Number of Attributes	Number of Templates	RP Average Number of Cancelled Jobs per Session
Number of Closed Sessions	Number of Transformations	RP Average Number of DB Result Rows per Job
Number of Columns	Number of Users	RP Average Number of DB Result Rows per Session
Number of Consolidations	Number of Users	RP Average Number of DB SQL Passes per Job
Number of Continuous Connected Users	Number of Users (All Intelligence Servers)	RP Average Number of Document Jobs per Report
Number of Continuous Sessions	Number of Users (Intelligence Server)	RP Average Number of Document Jobs per Session
Number of Custom Groups	Number of Users (no group-by Week of Year)	RP Average Number of DB Result Rows per User
Number of Days	Number of Users (Project)	RP Average Number of DB SQL Passes per Job
Number of Days - Document Jobs	Number of Web Sessions	RP Average Number of Document Jobs per User
Number of Days - Report Jobs	Number of Web Users	RP Average Number of Drill Jobs per Report
Number of DB Connections	Open Project Sessions Counter	RP Average Number of Drill Jobs per Session
Number of DB Instances	Open Sessions Counter	RP Average Daily Use Duration per job hh:mm:ss
Number of Disconnected Users	PM Avg Counter Value	RP Average Elapsed Duration per Job hh:mm:ss
Number of Distinct Users	PM Max Counter Value	RP Average Elapsed Duration per Job hh:mm:ss (Project)
Number of Document Subscriptions	PM Min Counter Value	RP Average Elapsed Duration per Job hh:mm:ss.000
Number of Documents	PM Total Counter Value	RP Average Elapsed Duration per Job msec
Number of E-mail Subscriptions	RP Activity Analysis Metric	RP Average Elapsed Duration per Subset Report Job hh:mm:ss
Number of Encapsulated Connected Users	RP Analytical Engine Duration hh:mm:ss	RP Average Elapsed Duration per User hh:mm:ss
Number of Encapsulated Sessions	RP Average Analytical Engine Exec Duration hh:mm:ss	
Number of Errored Subscriptions	RP Average CPU Duration per Job hh:mm:ss (Project)	
Number of Events	RP Average CPU Duration per Job (msec) (Project)	
Number of Executions	RP Average Daily Use Duration per job hh:mm:ss	
Number of Facts	RP Average Elapsed Duration per Job hh:mm:ss	
Number of File Subscriptions	RP Average Elapsed Duration per Job hh:mm:ss	
Number of Filters	RP Average Elapsed Duration per Job hh:mm:ss (Project)	
Number of Hierarchies	RP Average Elapsed Duration per Job hh:mm:ss.000	
Number of History List Subscriptions	RP Average Elapsed Duration per Job msec	
Number of Intelligence Servers	RP Average Elapsed Duration per Job hh:mm:ss	
Number of Metrics	RP Average Elapsed Duration per Job hh:mm:ss (Project)	
Number of New Connected Users	RP Average Elapsed Duration per Job hh:mm:ss	
Number of New Opened Sessions	RP Average Elapsed Duration per Job hh:mm:ss.000	
Number of Print Subscriptions	RP Average Elapsed Duration per Job msec	
Number of Projects	RP Average Elapsed Duration per Job hh:mm:ss	
Number of Projects 1	RP Average Elapsed Duration per Job hh:mm:ss (Project)	
Number of Prompts	RP Average Elapsed Duration per Job hh:mm:ss	
Number of Recipients	RP Average Elapsed Duration per Job hh:mm:ss	
Number of Report Caches Counter	RP Average Elapsed Duration per User hh:mm:ss	
Number of Report Subscriptions		

RP Average Number of Jobs w/o Cache Creation per User	RP Average Number of Jobs with Datamart Creation per User	RP Average Number of Non-Prompted Jobs per User
RP Average Number of Jobs w/o Cache Hit per Report	RP Average Number of Jobs with DB Error per Report	RP Average Number of Non-Scheduled Jobs per Report
RP Average Number of Jobs w/o Cache Hit per Session	RP Average Number of Jobs with DB Error per Session	RP Average Number of Non-Scheduled Jobs per Session
RP Average Number of Jobs w/o Cache Hit per User	RP Average Number of Jobs with DB Error per User	RP Average Number of Non-Scheduled Jobs per User
RP Average Number of Jobs w/o Datamart Creation per Report	RP Average Number of Jobs with Element Loading per Report	RP Average Number of Prompted Jobs per Report
RP Average Number of Jobs w/o Datamart Creation per Session	RP Average Number of Jobs with Element Loading per Session	RP Average Number of Prompted Jobs per Session
RP Average Number of Jobs w/o Datamart Creation per User	RP Average Number of Jobs with Element Loading per User	RP Average Number of Prompted Jobs per User
RP Average Number of Jobs w/o DB Error per Report	RP Average Number of Jobs with Error per Report	RP Average Number of Scheduled Jobs per Report
RP Average Number of Jobs w/o DB Error per Session	RP Average Number of Jobs with Error per Session	RP Average Number of Scheduled Jobs per Session
RP Average Number of Jobs w/o DB Error per User	RP Average Number of Jobs with Error per User	RP Average Number of Scheduled Jobs per User
RP Average Number of Jobs w/o Element Loading per Report	RP Average Number of Jobs with Security Filter per Report	RP Average Number of SQL Passes per Session
RP Average Number of Jobs w/o Element Loading per Session	RP Average Number of Jobs with Security Filter per Session	RP Average Number of SQL Passes per User
RP Average Number of Jobs w/o Element Loading per User	RP Average Number of Jobs with Security Filter per User	RP Average Prompt Answer Time per Job hh:mm:ss
RP Average Number of Jobs w/o Error per Report	RP Average Number of Non-Ad-Hoc Jobs per Report	RP Average Prompt Answer Time per Job hh:mm:ss (Project)
RP Average Number of Jobs w/o Error per Session	RP Average Number of Non-Ad-Hoc Jobs per Session	RP Average Prompt Duration per Job msec
RP Average Number of Jobs w/o Error per User	RP Average Number of Non-Ad-Hoc Jobs per User	RP Average Queue Duration per Job hh:mm:ss
RP Average Number of Jobs w/o Security Filter per Report	RP Average Number of Non-Cancelled Jobs per Report	RP Average Queue Duration per Job hh:mm:ss (Project)
RP Average Number of Jobs w/o Security Filter per Session	RP Average Number of Non-Cancelled Jobs per Session	RP Average Queue Duration per Job msec
RP Average Number of Jobs w/o Security Filter per User	RP Average Number of Non-Cancelled Jobs per User	RP Average SQL Execution Duration hh:mm:ss
RP Average Number of Jobs with Cache Creation per Report	RP Average Number of Non-Document Jobs per Report	RP Average SQL Execution Duration w/o Queue Time hh:mm:ss_deprecated
RP Average Number of Jobs with Cache Creation per Session	RP Average Number of Non-Document Jobs per Session	RP Average SQL Generation Duration hh:mm:ss
RP Average Number of Jobs with Cache Creation per User	RP Average Number of Non-Document Jobs per User	RP Average SQL Generation Duration w/o Queue Time hh:mm:ss_deprecated
RP Average Number of Jobs with Cache Hit per Report	RP Average Number of Non-Drill Jobs per Report	RP Average Subset Report Result Rows
RP Average Number of Jobs with Cache Hit per Session	RP Average Number of Non-Drill Jobs per Session	RP Cancelled Jobs with SQL exec
RP Average Number of Jobs with Cache Hit per User	RP Average Number of Non-Drill Jobs per User	RP Cancelled Jobs without SQL exec
RP Average Number of Jobs with Datamart Creation per Report	RP Average Number of Non-Prompted Jobs per Report	RP Count PU Jobs
RP Average Number of Jobs with Datamart Creation per Session	RP Average Number of Non-Prompted Jobs per Session	RP CPU Duration (msec)
		RP CPU Duration (msec) (Project)

RP Elapsed Duration by Non-Web Users hh:mm:ss	RP Min Number of DB Result Rows per Job	RP Number of Jobs with Element Loading
RP Elapsed Duration by Web Users hh:mm:ss	RP Min Number of DB Tables Accessed per Job	RP Number of Jobs with Element Loading (Project)
RP Elapsed Duration hh:mm:ss	RP Min Number of SQL Passes per Job	RP Number of Jobs with Error
RP Elapsed Duration hh:mm:ss (Project)	RP Min Queue Duration per Job hh:mm:ss	RP Number of Jobs with Error (Project)
RP Exec Duration - Project Level	RP Number of Ad-Hoc Jobs	RP Number of Jobs with Security Filter
RP Exec Duration hh:mm:ss	RP Number of Ad-Hoc Jobs (Project)	RP Number of Jobs with SQL Execution
RP Exec Duration hh:mm:ss (Project)	RP Number of Cancelled Jobs	Rp Number of Jobs with Unanswered Prompts (For Prompt answers reporting only)
RP Execution Duration of Cache hit jobs	RP Number of Cube Publishes	RP number of Narrowcast Server jobs
RP Execution Duration of Non-Cache hit jobs	RP Number of DB Tables Accessed	RP Number of Non-Ad-Hoc Jobs
RP Export Engine Jobs	RP Number of Drill Jobs	RP Number of Non-Cancelled Jobs
RP Export Engine Jobs (Project)	RP Number of Dynamically Sourced Report Jobs against Intelligent Cubes	RP Number of Non-Drill Jobs
RP First Exec Finish Timestamp	RP Number of Jobs	RP Number of Non-Prompted Jobs
RP First Exec Request Timestamp	Rp Number of Jobs (For Prompt answers reporting only)	RP Number of Non-Scheduled Jobs
RP First Exec Start Timestamp	RP Number of Jobs (Project)	RP Number of Prompted Jobs
RP First Job Request Timestamp	RP Number of Jobs accessing DB table/column	RP Number of Prompts
RP Intelligent Cube Row Size	RP Number of Jobs Containing Prompt Answer Value (For Prompt answers reporting only)	RP Number of Prompts (For Prompt answers reporting only)
RP Intelligent Cube Size (KB)	RP Number of Jobs For Concurrency Reporting	RP Number of Report Jobs exclusively from Report Execution
RP Jobs with No Data Returned	RP Number of Jobs near Mean	RP Number of Report Jobs from Document Execution
RP Last Exec Finish Timestamp	RP Number of Jobs Not Containing Prompt Answer Value (For Prompt answers reporting only)	RP Number of Report Jobs from Document Execution (Project)
RP Last Exec Request Date	RP Number of Jobs Today	RP Number of Reports Used
RP Last Exec Request Timestamp	RP Number of Jobs w/o Cache Creation	RP Number of Reports Used (at Project Level)
RP Last Exec Start Timestamp	RP Number of Jobs w/o Cache Hit	RP Number of Result Rows
RP Last Intelligent Cube Size (KB)	RP Number of Jobs w/o Datamart Creation	RP Number of Result Rows (Project)
RP Last Job Request Timestamp	RP Number of Jobs w/o DB Error	RP Number of Scheduled Jobs
RP Max CPU Duration per Job (msec)	RP Number of Jobs w/o Element Loading	RP Number of Scheduled Jobs (Project)
RP Max Elapsed Duration per Job hh:mm:ss	RP Number of Jobs w/o Error	RP Number of SQL Passes
RP Max Exec Duration per Job hh:mm:ss	RP Number of Jobs w/o Security Filter	RP Number of SQL Passes (Project)
RP Max Number of DB Result Rows per Job	RP Number of Jobs with Cache Creation	RP Number of Subset Report Jobs
RP Max Number of DB Tables Accessed per Job	RP Number of Jobs with Cache Hit	RP Number of Web Jobs
RP Max Number of SQL Passes per Job	RP Number of Jobs with Cache Hit (Project)	RP Number of Web Jobs w/o Cache Hit
RP Max Queue Duration per Job hh:mm:ss	RP Number of Jobs with Cube Hit	RP Number of Web Jobs with Cache Hit
RP Max SQL Execution Duration hh:mm:ss	RP Number of Jobs with Datamart Creation	RP Number of WH SQL Passes
RP Median(Activity Analysis Metrics)	RP Number of Jobs with DB Error	RP Number of WH SQL Passes (Project)
RP Median(Document # of Jobs)		RP OLAP View Report Job Elapsed Duration hh:mm:ss
RP Min CPU Duration per Job		RP OLAP View Report Result Rows
RP Min Elapsed Duration per Job hh:mm:ss		RP Percent of Ad-Hoc Jobs
RP Min Exec Duration per Job hh:mm:ss		

RP Percent of Ad-Hoc Jobs (Project)	RP Percentage of Non-Ad-Hoc Jobs	RP Schedule Relations
RP Percentage of Cancelled Jobs	RP Percentage of Non-Cancelled Jobs	RP Session Count
RP Percentage of Drill Jobs	RP Percentage of Non-Drill Jobs	RP SQL Engine Duration hh:mm:ss
RP Percentage of Jobs w/o Cache Creation	RP Percentage of Non-Prompted Jobs	RP SQL Execution Duration
RP Percentage of Jobs w/o Cache Hit	RP Percentage of Non-Scheduled Jobs	hh:mm:ss
RP Percentage of Jobs w/o Datamart Creation	RP Percentage of Prompted Jobs	RP SQL Size
RP Percentage of Jobs w/o DB Error	RP Percentage of Report Jobs exclusively from Report Execution	RP Timed out jobs
RP Percentage of Jobs w/o Element Loading	RP Percentage of Report Jobs from Document Execution	RP User Count
RP Percentage of Jobs w/o Error	RP Percentage of Scheduled Jobs	RP User Report Requests
RP Percentage of Jobs w/o Security Filter	RP Percentage of Scheduled Jobs (Project)	RP User Report Requests (Project)
RP Percentage of Jobs with Cache Creation	RP Percentage Usage by Non-Web Users	RP Waiting for AutoPrompt Time
RP Percentage of Jobs with Cache Hit	RP Percentage Usage by Web Users	RP Waiting for Execution Time
RP Percentage of Jobs with Cache Hit (Project)	RP Prompt Answer Time hh:mm:ss	Running Sum (Subscription Analysis Metric)
RP Percentage of Jobs with Datamart Creation	RP Prompt Answer Time hh:mm:ss (Project)	SQL Diversity
RP Percentage of Jobs with DB Error	RP Query Engine Duration	Subscription Analysis Metric
RP Percentage of Jobs with Element Loading	RP Queue Duration hh:mm:ss	Subscription Exec Duration
RP Percentage of Jobs with Error	RP Queue Duration hh:mm:ss (Project)	hh:mm:ss
RP Percentage of Jobs with Error (Project)	RP Report Popularity	Successful Document Jobs < 1 min
RP Percentage of Jobs with Security Filter	RP Request Date	Successful Document Jobs > 5 min
RP Percentage of Narrowcast Server jobs	RP RunningSum(Activity Analysis Metrics)	Successful Document Jobs 1-5 min
	RP RunningSum(Document # of Jobs)	Successful Jobs < 1 min
	RP Schedule Document Relations	Successful Jobs > 5 min
		Successful Jobs 1-5 min
		Sum(0)
		Time since Last Activity
		Today
		Total Number of Application Objects
		Total Number of Configuration Objects
		Total Number of Schema Objects

APPENDIX F: SUPPORTED DATA SOURCES

Data Source *	Windows Environment	UNIX/Linux Environment
ANSI SQL92 Generic database	Data Warehouse	Data Warehouse
Aster Data nCluster 4.5.1	Data Warehouse	Data Warehouse (Linux, Solaris)
Composite 5.1	Data Warehouse	
Composite 5.2	Data Warehouse	Data Warehouse (Linux, Solaris)
EXASolution 3.3	Data Warehouse	
Greenplum 3.2.x	Data Warehouse	Data Warehouse
Greenplum 3.3.x	Data Warehouse	Data Warehouse
Greenplum 4.0.x	Data Warehouse	Data Warehouse
Greenplum 4.1.x	Data Warehouse	Data Warehouse
Hadoop Hive 0.5/0.6 on CDH 2.0	Data Warehouse	Data Warehouse (Linux)
Hadoop Hive 0.7 on CDH 3.0	Data Warehouse	Data Warehouse (Linux)
HP Neoview 2.4	Data Warehouse	Data Warehouse
HP Neoview 2.5	Data Warehouse	Data Warehouse
IBM Cognos TM1 9.4.1	Data Warehouse	Data Warehouse
IBM Cognos TM1 9.5.x	Data Warehouse	Data Warehouse
IBM DB2 UDB for z/OS 8.1	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
IBM DB2 UDB for z/OS 9.1	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
IBM DB2 UDB for iSeries V5R4	Data Warehouse	Data Warehouse
IBM DB2 UDB for iSeries 6.1	Data Warehouse	Data Warehouse
IBM DB2 UDB for iSeries 7.1	Data Warehouse	Data Warehouse
IBM DB2 UDB 9.1 for Linux, UNIX, & Windows	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
IBM DB2 UDB 9.5 for Linux, UNIX, & Windows	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
IBM DB2 UDB 9.7 for Linux, UNIX, & Windows	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
IBM WebSphere Information Integrator 8.1/8.2	Data Warehouse	
Infobright 3.4.2	Data Warehouse	Data Warehouse
Infobright 3.5.2	Data Warehouse	Data Warehouse
Informix Dynamic Server 11.1	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
Informix Dynamic Server 11.5	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
Informix Dynamic Server 11.7	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
Informix Extended Parallel Server 8.5	Data Warehouse	Data Warehouse
Kognitio WX2 7.1.x	Data Warehouse	
MetaMatrix 5.5.3	Data Warehouse	

Data Source *	Windows Environment	UNIX/Linux Environment
Microsoft Access 2000	Data Warehouse Metadata Repository	Data Warehouse
Microsoft Access 2002	Data Warehouse Metadata Repository	Data Warehouse
Microsoft Access 2003	Data Warehouse Metadata Repository	Data Warehouse
Microsoft Access 2007	Data Warehouse Metadata Repository	Data Warehouse
Microsoft Analysis Services 2000	Data Warehouse	Data Warehouse
Microsoft Analysis Services 2005	Data Warehouse	Data Warehouse
Microsoft Analysis Services 2008	Data Warehouse	Data Warehouse
Microsoft Excel 2000	Data Warehouse	Data Warehouse
Microsoft Excel 2003	Data Warehouse	Data Warehouse
Microsoft Excel 2007	Data Warehouse	Data Warehouse
Microsoft SQL Server 2000 SP3/SP3a/SP4	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Microsoft SQL Server 2005 SP1/SP2/SP3/SP4	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Microsoft SQL Server 2008 SP1/SP2/R2	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Microsoft SQL Server 2008 ParallelData Warehouse	Data Warehouse	Data Warehouse
MySQL Community Server 5.0	Data Warehouse Metadata Repository	Data Warehouse (Linux)
MySQL Community Server 5.1	Data Warehouse Metadata Repository	Data Warehouse (Linux)
MySQL Community Server 5.5.8	Data Warehouse Metadata Repository	Data Warehouse (Linux)
MySQL Enterprise 5.0	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
MySQL Enterprise 5.1	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
MySQL Enterprise 5.5.8	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
Netezza 4.6	Data Warehouse	Data Warehouse
Netezza 5.0.x	Data Warehouse	Data Warehouse
Netezza 6.0.x	Data Warehouse	Data Warehouse
Oracle 9i and 9i R2	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Oracle 10g and 10g R2	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Oracle 11g and 11g R2	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Oracle Hyperion Essbase 7.x EDS	Data Warehouse	Data Warehouse
Oracle Hyperion Essbase 9.1, 9.2 HAS	Data Warehouse	Data Warehouse
Oracle Hyperion Essbase 9.3 APS	Data Warehouse	Data Warehouse
Oracle Hyperion Essbase 11 APS	Data Warehouse	Data Warehouse
Oracle Transparent Gateway 9i	Data Warehouse	
ParAccel 2.5	Data Warehouse	Data Warehouse (Linux)
ParAccel 3.0	Data Warehouse	Data Warehouse (Linux)

Data Source *	Windows Environment	UNIX/Linux Environment
PostgreSQL 8.3	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
PostgreSQL 8.4	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
PostgreSQL 9.0	Data Warehouse Metadata Repository	Data Warehouse Metadata Repository
Red Brick 6.3	Data Warehouse	Data Warehouse
Salesforce.com	Data Warehouse	
SAND CDBMS 6.1	Data Warehouse	
SAP BW 3.1	Data Warehouse	Data Warehouse
SAP BW 3.5	Data Warehouse	Data Warehouse
SAP BW 7.0	Data Warehouse	Data Warehouse
Sybase Adaptive Server 15	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Sybase Adaptive Server 15.5	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Sybase IQ 12.7	Data Warehouse	Data Warehouse
Sybase IQ 15	Data Warehouse	Data Warehouse
Sybase IQ 15.1	Data Warehouse	Data Warehouse
Sybase IQ 15.2	Data Warehouse	Data Warehouse
Teradata V2R6.2.x	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Teradata V12	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Teradata V13	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Teradata V13.1	Data Warehouse Metadata Repository Statistics Logging	Data Warehouse Metadata Repository Statistics Logging
Text Files	Data Warehouse	Data Warehouse
Vertica 3.5.x	Data Warehouse	Data Warehouse (Solaris)
Vertica 4.0.x	Data Warehouse	Data Warehouse (Solaris)
Vertica 4.1.x	Data Warehouse	Data Warehouse (Solaris)
Web services & XML using XQuery	Data Warehouse	Data Warehouse

As at the MicroStrategy 9 Release 6. See the latest Readme files installed with the platform for the most current list of supported data sources.

APPENDIX G: OPERATING PLATFORMS

Operating Systems

MicroStrategy Product	Environment	Operating System
MicroStrategy Intelligence Server	Windows 32-bit (x86)	Windows Server 2003 Enterprise Edition SP2
		Windows Server 2003 Standard Edition SP2
		Windows Server 2003 Enterprise Edition R2 SP2
		Windows Server 2003 Standard Edition R2 SP2
		Windows XP Professional Edition SP3
		Windows Vista Business Edition SP1 or SP2
		Windows Vista Enterprise Edition SP2
		Windows 7 Professional Edition SP1
		Windows 7 Enterprise Edition SP1
		Windows Server 2003 Enterprise Edition SP2
	Windows 64-bit (x64)	Windows Server 2003 Standard Edition SP2
		Windows Server 2003 Enterprise Edition R2 SP2
		Windows Server 2003 Standard Edition R2 SP2
		Windows Server 2008 Enterprise Edition SP2
		Windows Server 2008 Standard Edition SP2
		Windows Server 2008 Enterprise Edition R2 SP1
		Windows Server 2008 Standard Edition R2 SP1
		Windows Server 2008 Enterprise Edition SP1
		Windows Server 2008 Standard Edition SP1
		Windows Vista Business Edition SP2
	Linux 64-bit (x64)	Windows Vista Enterprise Edition SP2
		Windows 7 Professional Edition SP1
		Windows 7 Enterprise Edition SP1
		Red Hat Enterprise Linux 5.2
		Red Hat Enterprise Linux 5.3
		Red Hat Enterprise Linux 5.4
		Red Hat Enterprise Linux 5.5
		Red Hat Enterprise Linux 5.6
		Red Hat Enterprise Linux 6.0
		Oracle Linux
	Oracle Solaris 64-bit (SPARC)	SUSE Linux Enterprise Server 11
		Solaris 9.x
	IBM AIX 64-bit (POWER)	Solaris 10.x
		AIX 5.3
	HP HP-UX 64-bit (Itanium)	AIX 6.1
		HP-UX 11iv2
	Virtualization Software	HP-UX 11iv3
		LPAR for AIX
		Solaris Zones for Solaris 10.x
		VMWare
		Xen Hypervisor for Red Hat Enterprise Linux

MicroStrategy Product	Environment	Operating System
MicroStrategy Web and Mobile	ASP implementation 32-bit	Windows 2003 Standard Edition SP2 Windows 2003 Enterprise Edition SP2 Windows 2003 Standard Edition R2 SP2 Windows 2003 Enterprise Edition R2 SP2 Windows XP Professional Edition SP3 Windows Vista Business Edition SP1 or SP2 Windows Vista Enterprise Edition SP2 Windows 7 Professional Edition SP1 Windows 7 Enterprise Edition SP1
	JSP implementation 32-bit	Windows 2003 Standard Edition SP2 Windows 2003 Enterprise Edition SP2 Windows 2003 Standard Edition R2 SP2 Windows 2003 Enterprise Edition R2 SP2 Windows XP Professional Edition SP3 Windows Vista Business Edition SP1 or SP2 Windows 7 Professional Edition SP1 Windows 7 Enterprise Edition SP1
	JSP implementation 64-bit	Windows 2003 Standard Edition SP2 Windows 2003 Enterprise Edition SP2 Windows 2003 Standard Edition R2 SP2 Windows 2003 Enterprise Edition R2 SP2 Windows 2008 Standard Edition SP1, SP2 Windows 2008 Enterprise Edition SP1, SP2 Windows 2008 Standard Edition R2 SP1 Windows 2008 Enterprise Edition R2 SP1 Red Hat Enterprise Linux 5.2 Red Hat Enterprise Linux 5.3 Red Hat Enterprise Linux 5.4 Red Hat Enterprise Linux 5.5 Red Hat Enterprise Linux 5.6 Red Hat Enterprise Linux 6.0 SUSE Linux Enterprise Server 11.x Oracle Linux Oracle Solaris 9.x Oracle Solaris 10.x AIX 5.3 AIX 6.1 HP-UX 11iV2 HP-UX 11iV3

Other Web Server Components

Component	Environment	Component Version
Web Server	MicroStrategy Web 32-bit (ASP)	Microsoft Internet Information Services 5.1
		Microsoft Internet Information Services 6.0
		Microsoft Internet Information Services 7.0
		Microsoft Internet Information Services 7.5
	MicroStrategy Web 64-bit (ASP)	Microsoft Internet Information Services 5.1
		Microsoft Internet Information Services 6.0
		Microsoft Internet Information Services 7.0
		Microsoft Internet Information Services 7.5
	MicroStrategy Web 64-bit (JSP)	Apache HTTP Server 2.0
		Apache HTTP Server 2.2
		IBM HTTP Server 6.0.2
		IBM HTTP Server 6.1
		Oracle iPlanet Web Server, Enterprise Edition 6.1
		Oracle iPlanet Web Server, Enterprise Edition 7.0
Java Application Server	MicroStrategy Web 64-bit (JSP)	Apache Tomcat 5.5
		Apache Tomcat 6.0
		IBM WebSphere 6.1
		IBM WebSphere 7
		JBoss Application Server 4.2
		JBoss Enterprise Application Platform 4.2
		Oracle Web Application Server 10g Release 3
		Oracle WebLogic Server 9.2
		Oracle WebLogic Server 10.3
		SAP Netweaver Application server 7.1
		Sun Java System Application Server 9.1
Java Environment	MicroStrategy Web 32-bit (ASP)	Sun JRE 1.6.0 (32-bit)
		IBM JDK 1.5 (32-bit)
		IBM JDK 1.6 (32-bit)
		Oracle JRockit Mission Control 3.1.0 for Java V6 (32-bit)
		SAP JDK 1.5 (32-bit)
		Sun JDK 1.5 (32-bit)
		Sun JDK 1.6 (32-bit)
	MicroStrategy Web 64-bit (JSP)	HP-UX JDK 1.5 (32-bit & 64-bit)
		HP-UX JDK 1.6 (32-bit & 64-bit)
		IBM JDK 1.5 (32-bit & 64-bit)
		IBM JDK 1.6 (32-bit & 64-bit)
		Oracle JRockit Mission Control 3.1.0 for Java V6 (32-bit & 64-bit)
		SAP JDK 1.5 (32-bit & 64-bit)
		Sun JDK 1.5 (32-bit & 64-bit)
		Sun JDK 1.6 (32-bit & 64-bit)

Component	Environment	Component Version
Portal Server	MicroStrategy Web 32- & 64-bit (ASP & JSP)	DotNetNuke 5.6 Drupal 7.0 IBM WebSphere Portal 6.1 IBM WebSphere Portal 7.0 Liferay Portal 5.2 Liferay Portal 6.0 Microsoft Office SharePoint Portal 2007 Microsoft SharePoint Portal 2010 Oracle WebLogic Portal 9.2 Oracle WebLogic Portal 10.3.2 SAP NetWeaver Composition Environment 7.1
GIS (Mapping) Server	MicroStrategy Web 32- & 64-bit (ASP & JSP)	ESRI ArcGIS 9.3.1 ESRI ArcGIS 10 Google Maps
Web Browser	Windows Client	Firefox 3.x Firefox 4.x Google Chrome 9.x Google Chrome 10.x Microsoft Internet Explorer 7.x Microsoft Internet Explorer 8.x Microsoft Internet Explorer 9.x
	iPad, iPhone, iPod Touch Client	Safari Web browser on iOS 3.2.x Safari Web browser on iOS 4.2.x Safari Web browser on iOS 4.3.x
	Linux Client	Firefox 3.x Firefox 4.x
	Mac OS Client	Firefox 3.x Firefox 4.x Safari 5.x
Export Applications	MicroStrategy Web 32- & 64-bit (ASP & JSP)	Adobe Acrobat Reader 8.x Adobe Acrobat Reader 9.x Adobe Acrobat Reader 10.x Microsoft Office 2002 SP3 Microsoft Office 2003 SP3 Microsoft Office 2007 SP2 Microsoft Office 2010 Microsoft Office for Mac 2011
Flash Player	MicroStrategy Web 32- & 64-bit (ASP & JSP)	Adobe Flash Player 10.1 Adobe Flash Player 10.2

Security Authentication Servers

MicroStrategy Product	Authentication Component	Component Version
MicroStrategy Intelligence Server	LDAP Servers	IBM Tivoli LDAP Server
		Microsoft Active Directory
		Novell NDS eDirectory 8.7
		Open LDAP Server 3
		Oracle LDAP Server
		Sun ONE Directory Server 5.2
		Sun ONE Directory Server 6.3
	Trusted Authentication System	Computer Associates Siteminder 6.0 SP5
		Oracle Identity Manager 11g
		Tivoli Access Manager for eBusiness 6.0
		Tivoli Access Manager for eBusiness 6.1.1

Supported Interface Languages

Language
Chinese (Simplified)
Chinese (Traditional)
Danish
Dutch (Netherlands)
English (US)
English (UK)
French
French (Belgium)
French (Switzerland)
German
German (Switzerland)
Italian
Italian (Switzerland)
Japanese
Korean
Polish
Portuguese (Brazilian)
Russian
Spanish
Swedish

As of the MicroStrategy 9 Release 6. See the latest Readme files installed with the platform for the most current list of supported operating platforms.

APPENDIX H: SECURITY PRIVILEGES

The table below lists all the security privileges that govern access to the BI functionality in the MicroStrategy BI platform. These can be granted by user group, security role, or individual user.

End-User Functionality	Development Power	Administration Control
Run Report Services documents	Create Report Services documents	Create/edit security filters
Change preferences	Create interactive reports	Manage clusters
Change grid/graph view	Format grid properties	Manage caches
Drill up and down	Format graph properties	Manage database connections
Search	Design Intelligent Cubes	Create/edit DB instances & connections
Print	Create autostyles	Monitor jobs
Rerun report on the data source	Modify report data options	Configure projects
Schedule E-mail delivery	Use data explorer	Create/edit security roles
Schedule printer delivery	Define report for bulk export	Monitor user connections
Subscribe to reports	Create consolidations	Create/edit users and user groups
Send report via E-mail	Create custom groups	Configure MicroStrategy Web & Mobile
Multi-level Sort	Create attributes	Assign security filters
Use page-by	Create hierarchies	Use Command Manager
View history list	Create facts	Manage History List
Add/remove objects to/from report	Create metrics	Configure Intelligence Server
Create and format derived metrics	Create templates	Configure delivery contact security
Show Intelligent Cube objects	Create filters	Manage delivery transmitters and devices
Create view filter	Use find and replace	Define E-mail address for deliveries
Add report to history list	Create cube report	Define file location for deliveries
Drill anywhere	Create subtotals	Define print location for deliveries
Define column aliases	Create prompts	Use Integrity Manager
Choose attribute form	Change VLDB properties	Manage cubes
Configure toolbars	Create Freeform SQL/XQuery report	Manage jobs
Drill on metrics	Create custom drill maps	Manage subscriptions
Export very large file	Create transformations	Manage user connections
Run datamart report	Create, train and deploy data mining metric	Assign security roles
Export report (Excel, Flash, PDF, HTML, text)	Create project documentation	Audit change journal
Filter by selection	Create new project	Bypass object security access checks
Manage objects	Import OLAP cube structure	Configure caches
Add subtotals	Import data warehouse structure	Configure change journaling
Pivot grid rows, columns, page-by	Publish Intelligent Cube	Configure connection maps
View report details	Define HTML container in documents	Configure governing

End-User Functionality	Development Power	Administration Control
View report SQL/MDX	Manage document datasets	Configure group membership
Save to personal folder	Configure transaction	Configure history list
Save to shared folders	Define advanced report options	Configure language settings
Schedule file delivery	Define links to reports and documents	Configure security settings
Format graph (simple)	Use translation editor	Configure statistics
Asynchronous report execution	Define dynamic datamart report	Create/edit contacts and addresses
Enable header locking	Import data mining or custom function	Create/edit database logins
Drill in Intelligent Cube	Use Object Manager	Create/edit schedules and events
Create folder		Duplicate project
Use report caches		Edit project status
Run interactive reports		Web-based Intelligence Server admin
Use MicroStrategy Office		Enable user
Enable Web online training		Grant/revoke security privilege
Link to reports and documents		Idle/resume project
Use MicroStrategy Web		Link users and groups to external accounts
Create derived elements		Load/unload project
Create Visual Insight analysis		Monitor caches
Save Visual Insight analysis		Monitor cubes
View Visual Insight analysis		Monitor database connections
Define alert for report delivery		Monitor History List
Use Report Writer & Wizard to create report		Monitor project
Add and edit notes		Monitor subscriptions
Define access control for objects		Reset user password
Use visual and advanced threshold editor		Monitor performance counters
Subscribe other to report deliveries		
Use dynamic sourcing		
Run documents with transactions		
Create shortcuts		
Save prompt answers		
Use MicroStrategy Mobile		
Run MultiSource report		
Use MicroStrategy Desktop		
Import data from file or database in Web		

APPENDIX I: API CLASSES, METHODS, PROPERTIES, AND INTERFACES

MicroStrategy Web API Packages

com.microstrategy.plugins.google.transforms	com.microstrategy.sdk.tools.mergeXML	com.microstrategy.sdk.tools.WebConfiguration.editors.styles
com.microstrategy.plugins.portlet2portlet.addons	com.microstrategy.sdk.tools.WebConfiguration	com.microstrategy.sdk.tools.WebConfiguration.editors.tasks
com.microstrategy.plugins.portlet2portlet.beans	com.microstrategy.sdk.tools.WebConfiguration.actions	com.microstrategy.sdk.tools.WebConfiguration.editors.transformdefinitions
com.microstrategy.plugins.portlet2portlet.transforms	com.microstrategy.sdk.tools.WebConfiguration.config	com.microstrategy.sdk.tools.WebConfiguration.editors.visualization
com.microstrategy.plugins.portlet2portlet.transforms.contextmenus	com.microstrategy.sdk.tools.WebConfiguration.config.preferences	com.microstrategy.sdk.tools.WebConfiguration.editors.webBrowser
com.microstrategy.sdk.addons	com.microstrategy.sdk.tools.WebConfiguration.CSS	com.microstrategy.sdk.tools.WebConfiguration.editors.webproperties
com.microstrategy.sdk.appobjects	com.microstrategy.sdk.tools.WebConfiguration.dialogs	com.microstrategy.sdk.tools.WebConfiguration.editors.widgets
com.microstrategy.sdk.beans	com.microstrategy.sdk.tools.WebConfiguration.editors	com.microstrategy.sdk.tools.WebConfiguration.popupMenus
com.microstrategy.sdk.externalsecurity	com.microstrategy.sdk.tools.WebConfiguration.editors.browsersettings	com.microstrategy.sdk.tools.WebConfiguration.preferences
com.microstrategy.sdk.kits.webservices.reflection	com.microstrategy.sdk.tools.WebConfiguration.editorerrors	com.microstrategy.sdk.tools.WebConfiguration.search
com.microstrategy.sdk.kits.webservices.tasks	com.microstrategy.sdk.tools.WebConfiguration.editorevents	com.microstrategy.sdk.tools.WebConfiguration.upgrade
com.microstrategy.sdk.kits.webservices.tasks.objects	com.microstrategy.sdk.tools.WebConfiguration.exportformats	com.microstrategy.sdk.tools.WebConfiguration.utils
com.microstrategy.sdk.samples.beans	com.microstrategy.sdk.tools.WebConfiguration.editorfolderlinks	com.microstrategy.sdk.tools.WebConfiguration.wizards
com.microstrategy.sdk.samples.customBeanAndEventHandler	com.microstrategy.sdk.tools.WebConfiguration.editorfoldermaps	com.microstrategy.sdk.tools.WebConfiguration.wizards.pages
com.microstrategy.sdk.samples.dialog	com.microstrategy.sdk.tools.WebConfiguration.editormstrsettings	com.microstrategy.sdk.tools.WebConfiguration.wizards.tasks
com.microstrategy.sdk.samples.events	com.microstrategy.sdk.tools.WebConfiguration.editorpages	com.microstrategy.sdk.tools.WebConfiguration.wizards.webservies
com.microstrategy.sdk.samples.externalsecurity	com.microstrategy.sdk.tools.WebConfiguration.editorpapersizes	com.microstrategy.sdk.tools.WPSPIK
com.microstrategy.sdk.samples.monitoring	com.microstrategy.sdk.tools.WebConfiguration.editorpreference	com.microstrategy.utils.cache
com.microstrategy.sdk.samples.sessionhelper	com.microstrategy.sdk.tools.WebConfiguration.editorribbon	com.microstrategy.utils.concurrent
com.microstrategy.sdk.samples.transforms	com.microstrategy.sdk.tools.WebConfiguration.editorribbonlist	com.microstrategy.utils.config
com.microstrategy.sdk.samples.transforms.contextmenus	com.microstrategy.sdk.tools.WebConfiguration.editorshortcut	com.microstrategy.utils.crypto
com.microstrategy.sdk.samples.transforms.excel97	com.microstrategy.sdk.tools.WebConfiguration.editorshortcutlist	com.microstrategy.utils.licensing
com.microstrategy.sdk.samples.usermodelmanagement	com.microstrategy.sdk.tools.WebConfiguration.editorshortcutlistset	com.microstrategy.utils.localization
com.microstrategy.sdk.samples.webobjects		com.microstrategy.utils.log
com.microstrategy.sdk.samples.writeback		com.microstrategy.utils.log.editor
com.microstrategy.sdk.tasks.writeback		com.microstrategy.utils.serialization
com.microstrategy.sdk.tasks.writeback.beans		
com.microstrategy.sdk.tasks.writeback.transform		

com.microstrategy.utils.subscription	com.microstrategy.web.platform	AbstractEditableLoggingComponent
com.microstrategy.utils.xml	com.microstrategy.web.portalkit	AbstractEditorTransform
com.microstrategy.utils.xml.builders	com.microstrategy.web.portalkit.struts	AbstractElementList
com.microstrategy.web.admin	com.microstrategy.web.portlets	AbstractEventHandlerFactory
com.microstrategy.web.admin.beans	com.microstrategy.web.portlets.bea	AbstractExpressionTransform
com.microstrategy.web.admin.transforms	com.microstrategy.web.portlets.common	AbstractExternalSecurity
com.microstrategy.web.app	com.microstrategy.web.portlets.ibm	AbstractFilterElementTransform
com.microstrategy.web.app.addons	com.microstrategy.web.portlets.liferay	AbstractFilterOperatorContextMenuItemBuilder
com.microstrategy.web.app.beans	com.microstrategy.web.portlets.oracle	AbstractFindDisplayUnitsVisitor
com.microstrategy.web.app.gui	com.microstrategy.web.portlets.pluto	AbstractFindDisplayUnitVisitor
com.microstrategy.web.app.gui.shortcuts	com.microstrategy.web.portlets.sap	AbstractFolderContextMenuItemBuilder
com.microstrategy.web.app.mobile.config	com.microstrategy.web.portlets.utils	AbstractFolderParentMenu
com.microstrategy.web.app.platform	com.microstrategy.web.preferences	AbstractFolderTransform
com.microstrategy.web.app.preferences	com.microstrategy.web.servant	AbstractForm
com.microstrategy.web.app.taglibs	com.microstrategy.web.servlets	AbstractFormatTabTransform
com.microstrategy.web.app.taglibs.taskproc	com.microstrategy.web.tags	AbstractFrameEditorBean
com.microstrategy.web.app.tags	com.microstrategy.web.tasks	AbstractGetGraphPropertiesTask
com.microstrategy.web.app.tasks	com.microstrategy.web.tasks.config	AbstractGridDrillTask
com.microstrategy.web.app.tasks.config	com.microstrategy.web.transform	AbstractGuiElement
com.microstrategy.web.app.transforms	com.microstrategy.webapi	AbstractHandler
com.microstrategy.web.app.transforms.contextmenus	com.microstrategy.webservices	AbstractHandlerWithText
com.microstrategy.web.app.utils	com.microstrategy.webservices.Axis2	AbstractHeaderFooterTransform
com.microstrategy.web.app.utils.cache	com.microstrategy.webservices.jbridge:	AbstractIfTag
com.microstrategy.web.app.utils.compression		AbstractIfTagHelper
com.microstrategy.web.app.utils.licensing		AbstractInboxContextMenuBuilder
com.microstrategy.web.app.utils.log		AbstractInstance
com.microstrategy.web.app.utils.upgrade		AbstractLayoutSource
com.microstrategy.web.beans		AbstractLayoutSource.LayoutCacheHint
com.microstrategy.web.blocks		AbstractLayoutTransform
com.microstrategy.web.blocks.config		AbstractLimitExpressionBean
com.microstrategy.web.blocks.macros		AbstractLocalBeanFactory
com.microstrategy.web.blocks.renderers		AbstractMobileSubscriptionTask
com.microstrategy.web.blocks.xforms		AbstractMobileSubscriptionTask.SimpleSubscription-Transform
com.microstrategy.web.certificate		AbstractModule
com.microstrategy.web.config		AbstractMojoTransform
com.microstrategy.web.controller		AbstractNoBodyTag
com.microstrategy.web.designer		AbstractNoBodyTagHelper
com.microstrategy.web.filter		AbstractObjectBrowserTabBean
com.microstrategy.web.objects		AbstractObjectBrowserTabTransform
com.microstrategy.web.objects.admin.licensing		AbstractObjectExplorerBean
com.microstrategy.web.objects.admin.monitors		AbstractObjectExplorerEventHandler
com.microstrategy.web.objects.admin.serverconfig		AbstractObjectExplorerTransform
com.microstrategy.web.objects.admin.users		AbstractObjectExplorerTreeTransform
com.microstrategy.web.objects.localization		AbstractObjectInfoListTransform
com.microstrategy.web.objects.rw		AbstractOptionsTransform
com.microstrategy.web.pik		AbstractPagebyContextMenuItemBuilder
		AbstractParameterBuilder
		AbstractParameterBuilder
		AbstractParameterBuilder.Parameter
		AbstractParameterBuilder.Parameters
		AbstractPersistable
		AbstractPersistableAppComponent
		AbstractPersistableFrameElement

AbstractPortletLogHandler	AbstractTransform	AdminNavBarTagHelper
AbstractPreferencesTransform	AbstractTransformable	AdminPageTreeObject
AbstractPrivilegesEditorTransform	AbstractUserEntitiesTreeCartTransform	AdminParamInfosTag
AbstractPromptEditorTransform	AbstractUserMgrContextMenuItemBuilder	AdminParamInfosTagHelper
AbstractPromptEditorTransform.ListItem	AbstractViewBeanParserBean	AdminParamInputTableTag
AbstractPromptExpressionTransform	AbstractWebBeanTransform	AdminParamInputTableTagHelper
AbstractPromptObjectTransform	AbstractWebComponent	AdminPathTransform
AbstractPromptWidgetTransform	AbstractWebFeatures	AdminPropertiesPage
AbstractRenderer	AbstractWritebackControl	AdminSelectTagHelper
AbstractReportCellComposed-ContextMenuItem-Builder	AbstractWritebackWriter	AdminSelectTaskContentTypeTag
AbstractReportCellContextMenuItemBuilder	AbstractXMLConfigModule	AdminSelectTaskContentTypeTagHelper
AbstractReportCellInsertMetric	AccessControlListSample	AdminSelectTaskEnvTag
AbstractReportCellMove	AccordionTabBeanTransform	AdminSelectTaskEnvTagHelper
AbstractReportCellParentMenu	AccordionTabManagerBeanFeaturesImpl	AdminSelectTaskIDTag
AbstractReportCellSort	AccordionTabManagerBeanImpl	AdminSelectTaskIDTagHelper
AbstractReportCellThresholds	AccordionTabManagerEventHandler	AdminServer
AbstractReportCellThresholds.ThresholdsContext-MenuBuilder	AccordionTabManagerTransform	AdminServersHelper
AbstractReportDataTransform	ActualParameter	AdminServersList
AbstractReportDataVisualizationTransform	ActualParameterImpl	AdminServlet
AbstractReportExportTransform	ActualParameterInfo	AdminTabBean
AbstractReportGridCellHeader	ActualParameters	AdminTable
AbstractReportGridCellHeader.AxisDepth	ActualParametersImpl	AdminTaskInfosTag
AbstractReportGridCellTitle	ActualParametersPage	AdminTaskInfosTagHelper
AbstractReportGridCellTitle.TitlePivotButtonInfo	AddDataSetObjectExplorerEventHandler	AdminTaskURLTag
AbstractReportGridDisplayCell	AddDataSetWizardTransform	AdminTaskURLTagHelper
AbstractReportGridDisplayCell.PivotButtonInfo	AddDimensionAttributeToGridOrFilter	AdminTitleTag
AbstractReportGridExportTransform	AddDimensionElementToFilter	AdminTitleTagHelper
AbstractReportGridTransform	AdditionalTaskPropertiesPage	AdvancedSearchLinkTag
AbstractReportOutlineModeExportTransform	AddonInfo	AdvancedSearchLinkTagHelper
AbstractReportTransform	AddonInfoList	AdvancedShortcutProperties
AbstractReportXMLTransform	AddonParametersCartDialog	AdvancedThresholdsEditorTransform
AbstractResultSetBinaryResultsTask	AddressDeleteTransform	AggregatedAppEventHandler
AbstractResultSetResultsTask	AddressListBean	AggregatedEventHandler
AbstractRWBeanVisitor	AddressListFileTransform	AggregatedFolderEventHandler
AbstractRWBeanVisitor2	AddressListPrintTransform	AggregatedReportSavePropertiesEventHandler
AbstractRWDataVisualizationTransform	AddressListTransform	AggregatedWebFeatures
AbstractRWTransform	AddVisualizationTask	AggregatingTaskFactory
AbstractSaveReportPropertiesAddOn	AddWidgetTask	AggregatingTaskFactoryConfig
AbstractSaveTask	AdminBean	AlertsEditorBean
AbstractScheduleListTransform	AdminBeanEventHandler	AlertsEditorTransform
AbstractServerPropertiesTransform	AdminBeanFactory	AlertsObjectExplorerTransform
AbstractShortcut	AdminContextTag	AlertsSubscriptionMobileWidgetImpl
AbstractShortcutList	AdminContextTagHelper	AllObjectBrowserBean
AbstractSimplifiedRWWMLTransform	AdminController	AllObjectBrowserBeanImpl
AbstractStyleCatalogElement	AdminDatasetTableTagHelper	AllObjectBrowserHTMLTransform
AbstractStyleCatalogList	AdminErrorsTreeObject	AllObjectBrowserTransform
AbstractSubscriptionEditTransform	AdminFolderDelete	AllowablePropertiesWizardPage
AbstractSubscriptionFolderBeanTransform	AdminFolderEdit	AllowableValueConfig
AbstractSubscriptionListTransform	AdminFolderSearchResultsTransform	AllowableValuesConfig
AbstractSubscriptionsEditorTransform	AdminFolderViewTransform	AllowableValueWizard
AbstractTabTransform	AdminGenericTabTransform	AllowContentTypes
AbstractTemplateElementsBean	AdminInitParamsTag	AllowDomains
	AdminInitParamsTagHelper	AnchorTag
	AdminNavBarTag	AnchorTagImpl

AnchorWebTag	ArrayOfMWSElementFormInstance	ArrayOfMWSObjectInfoProperty.Factory
AnnotationBean	ArrayOfMWSElementFormInstance	ArrayOfMWSObjectInfoPropertyBridge
AnnotationsTransform	ArrayOfMWSElementFormInstance.Factory	ArrayOfMWSObjectInfoPropertySet
AnswerPromptsTask	ArrayOfMWSElementFormInstanceJBridge	ArrayOfMWSObjectInfoPropertySet
AppAddOns	ArrayOfMWSElementRestriction	ArrayOfMWSObjectInfoPropertySet.Factory
AppBeanError	ArrayOfMWSElementRestriction	ArrayOfMWSObjectInfoPropertySetGroup
AppBeanFactory	ArrayOfMWSElementRestriction.Factory	ArrayOfMWSObjectInfoPropertySetGroup
AppComponent	ArrayOfMWSElementRestrictionJBridge	ArrayOfMWSObjectInfoPropertySetGroup.Factory
AppContext	ArrayOfMWSExecutelInfo	ArrayOfMWSObjectInfoPropertySetGroupJBridge
AppContextImpl	ArrayOfMWSExecutelInfo	ArrayOfMWSObjectInfoPropertySetBridge
AppController	ArrayOfMWSExecutelInfo.Factory	ArrayOfMWSObjectInfoPropertySetSpec
AppControllerImpl	ArrayOfMWSExecutelInfoJBridge	ArrayOfMWSObjectInfoPropertySetSpec
AppControllerServlet	ArrayOfMWSExportSetting	ArrayOfMWSObjectInfoPropertySetSpec.Factory
AppEventHandlerFactory	ArrayOfMWSExportSetting	ArrayOfMWSObjectInfoPropertySetSpecBridge
AppGlobalContext	ArrayOfMWSExportSetting.Factory	ArrayOfMWSPageByElements
AppGlobalContextImpl	ArrayOfMWSExportSettingJBridge	ArrayOfMWSPageByElements
AppLayoutParser	ArrayOfMWSFunctionalityProperty	ArrayOfMWSPageByElements.Factory
AppLayoutSourceFile	ArrayOfMWSFunctionalityProperty	ArrayOfMWSPageByElementsJBridge
ApplicationParameters	ArrayOfMWSFunctionalityProperty.Factory	ArrayOfMWSPageByInfo
ApplicationPropertiesPage	ArrayOfMWSFunctionalityPropertyJBridge	ArrayOfMWSPageByInfo
AppLoginTask	ArrayOfMWSHierarchyAttributeFormInfo	ArrayOfMWSPageByInfo.Factory
AppTaskFactory	ArrayOfMWSHierarchyAttributeFormInfo	ArrayOfMWSPageByInfoBridge
AppTaskRequestContext	ArrayOfMWSHierarchyAttributeFormInfo.Factory	ArrayOfMWSProjectFolderID
AppTransform	ArrayOfMWSHierarchyAttributeFormInfoJBridge	ArrayOfMWSProjectFolderObj
AppTransformKeyMapper	ArrayOfMWSHierarchyAttributeInfo	ArrayOfMWSProjectFolderObj
ArgumentInfo	ArrayOfMWSHierarchyAttributeInfo	ArrayOfMWSProjectFolderObj.Factory
ArgumentTag	ArrayOfMWSHierarchyAttributeInfo.Factory	ArrayOfMWSProjectFolderObjJBridge
ArgumentUnresolvedException	ArrayOfMWSHierarchyAttributeInfoJBridge	ArrayOfMWSProjectInfo
ArrayOfEnumMWSObjectType	ArrayOfMWSHierarchyElementFilter	ArrayOfMWSProjectInfo
ArrayOfEnumMWSObjectType	ArrayOfMWSHierarchyElementFilter	ArrayOfMWSProjectInfo.Factory
ArrayOfEnumMWSObjectType.Factory	ArrayOfMWSHierarchyElementFilter.Factory	ArrayOfMWSProjectInfoBridge
ArrayOfEnumMWSObjectTypeJBridge	ArrayOfMWSHierarchyElementFilterJBridge	ArrayOfMWSProjectSessionProperty
ArrayOfEnumMWSObjSubType	ArrayOfMWSHierarchyElementInfo	ArrayOfMWSProjectSessionProperty
ArrayOfEnumMWSObjSubType	ArrayOfMWSHierarchyElementInfo	ArrayOfMWSProjectSessionProperty.Factory
ArrayOfEnumMWSObjSubType.Factory	ArrayOfMWSHierarchyElementInfo.Factory	ArrayOfMWSProjectSessionPropertyBridge
ArrayOfEnumMWSObjSubTypeJBridge	ArrayOfMWSHierarchyElementInfoJBridge	ArrayOfMWSProjectSetting
ArrayOfMWSAttachmentInfo	ArrayOfMWSHistoryInfo	ArrayOfMWSProjectSetting
ArrayOfMWSAttachmentInfo	ArrayOfMWSHistoryInfo	ArrayOfMWSProjectSetting.Factory
ArrayOfMWSAttachmentInfo.Factory	ArrayOfMWSHistoryInfo.Factory	ArrayOfMWSProjectSettingJBridge
ArrayOfMWSAttachmentInfoJBridge	ArrayOfMWSHistoryInfoJBridge	ArrayOfMWSProjectSourceInfo
ArrayOfMWSConnectProperty	ArrayOfMWSInfoProperty	ArrayOfMWSProjectSourceInfo
ArrayOfMWSConnectProperty	ArrayOfMWSInfoProperty	ArrayOfMWSProjectSourceInfo.Factory
ArrayOfMWSConnectProperty.Factory	ArrayOfMWSInfoProperty.Factory	ArrayOfMWSProjectSourceInfoJBridge
ArrayOfMWSConnectProperty2	ArrayOfMWSInfoPropertyJBridge	ArrayOfMWSProjectWellKnownObj
ArrayOfMWSConnectProperty2	ArrayOfMWSMetricInfo	ArrayOfMWSProjectWellKnownObj
ArrayOfMWSConnectProperty2.Factory	ArrayOfMWSMetricInfo	ArrayOfMWSProjectWellKnownObj.Factory
ArrayOfMWSConnectProperty2JBridge	ArrayOfMWSMetricInfo.Factory	ArrayOfMWSProjectWellKnownObjBridge
ArrayOfMWSConnectPropertyJBridge	ArrayOfMWSMetricInfoJBridge	ArrayOfMWSPromptSuggestions
ArrayOfMWSCurrentElement	ArrayOfMWSObjectInfo	ArrayOfMWSPromptSuggestions
ArrayOfMWSCurrentElement	ArrayOfMWSObjectInfo	ArrayOfMWSPromptSuggestions.Factory
ArrayOfMWSCurrentElement.Factory	ArrayOfMWSObjectInfo.Factory	ArrayOfMWSPromptSuggestionsJBridge
ArrayOfMWSCurrentElementJBridge	ArrayOfMWSObjectInfoJBridge	ArrayOfMWSProperty
ArrayOfMWSDiscoverColumn	ArrayOfMWSObjectInfoProperty	ArrayOfMWSProperty
ArrayOfMWSDiscoverRow	ArrayOfMWSObjectInfoProperty	ArrayOfMWSProperty.Factory

ArrayOfMWSPropertyJBridge	AttributeBean	BeanTaskConfig
ArrayOfMWSReportHeader	AttributeElementsWidgetTransform	BeanTaskConstants
ArrayOfMWSReportHeader	AttributeEventHandler	BeanTaskFactory
ArrayOfMWSReportHeader.Factory	AttributeFormsEditorBean	BeanTaskRequestContext
ArrayOfMWSReportHeaderJBridge	AttributeFormsEditorTransform	BeanTreeObject
ArrayOfMWSResourceRequestInfo	AttributeFormsElement	BeanValueTag
ArrayOfMWSResourceRequestInfo	AttributeFormsQualificationBean	BeanValueTagHelper
ArrayOfMWSResourceRequestInfo.Factory	AttributeFormsQualificationBeanImpl	BerInputStream
ArrayOfMWSResourceRequestInfoJBridge	AttributeFormsQualificationTransform	BinaryXMLAPITask
ArrayOfMWSResourceReturnInfo	AttributesXt	Block
ArrayOfMWSResourceReturnInfo	AttrTag	BlockColorPickerTransform
ArrayOfMWSResourceReturnInfo.Factory	AuthenticationHelper	BlockConfig
ArrayOfMWSResourceReturnInfoJBridge	AuthFormatter	BlockContext
ArrayOfMWSRestriction	Axis2ServantTargetInfo	BlockElementsReorderPage
ArrayOfMWSRestriction	AxisServantRuntimeContext	BlockElementsReorderWizard
ArrayOfMWSRestriction.Factory	BalloonWindow	BlockFactory
ArrayOfMWSRestrictionJBridge	Base16	BlockFrameWorkConfig
ArrayOfMWSRSExportSetting	Base64	BlockInputValidator
ArrayOfMWSRSExportSetting	Base64Harder	BlockLibraryConfig
ArrayOfMWSRSExportSetting.Factory	Base64Harder.InputStream	BlockList
ArrayOfMWSRSExportSettingJBridge	Base64Harder.OutputStream	BlockListConfig
ArrayOfMWSRSLayoutInfo	BaseObjectInfo	BlockListElement
ArrayOfMWSRSLayoutInfo	BaseTag	BlockListElement.ListAction
ArrayOfMWSRSLayoutInfo.Factory	BaseTag	BlockLoaderTask
ArrayOfMWSRSLayoutInfoJBridge	BaseTagHelper	BlockMenuAddElementWizard
ArrayOfMWSSyncCommunicationSetting	BaseTagImpl	BlockMenuDependencyAnalyzer
ArrayOfMWSSyncCommunicationSetting	BaseTaskRequestContext	BlockMenuItemEditEventPage
ArrayOfMWSSyncCommunicationSetting.Factory	BaseUserEntityBeanTask	BlockMenuItemEditIndividualSelectionPage
ArrayOfMWSSyncCommunicationSettingJBridge	BasicPageProperties	BlockMenuItemEditIndividualWizard
ArrayOfMWSUserPreference	BasicStyleProperties	BlockMenuItemEditPropertiesPage
ArrayOfMWSUserPreference	BasicTaskPropertiesPage	BlockMenuItemEditWizard
ArrayOfMWSUserPreference.Factory	BatchAuditResults	BlockMenuItemEventPage
ArrayOfMWSUserPreferenceJBridge	BatchLicenseAudit	BlockMenuItemPropertiesPage
ArrayOfMWSUserPrivilege	BeanContext	BlockMenuItemResultsPage
ArrayOfMWSUserPrivilege	BeanContextImpl	BlockMenuItemSelectionPage
ArrayOfMWSUserPrivilege.Factory	BeanDefn	BlockMenuItemStartPage
ArrayOfMWSUserPrivilegeJBridge	BeanDefnImpl	BlockMenuItemWizard
ArrayOfMWSVisualization	BeanDefns	BlockMenuItemElementEditWizard
ArrayOfMWSVisualization	BeanDefnsImpl	BlockMenuItemItemsPage
ArrayOfMWSVisualization.Factory	BeanFactory	BlockMenuItemTreeObject
ArrayOfMWSVisualizationJBridge	BeanFactory.BeanFactoryInfo	BlockMenuResultsPage
ArrayOfMWSVisualizationList	BeanFactory.BeanMappingInfo	BlockMenuSelectionPage
ArrayOfMWSVisualizationList	BeanFactory.BeanMappingInfoList	BlockMenuTreeObject
ArrayOfMWSVisualizationList.Factory	BeanFactory.BeanPackageInfo	BlockMenuWizard
ArrayOfMWSVisualizationListJBridge	BeanFactory.BeanPackageInfoList	BlockProperty
ArrayOfString	BeanGeneratedOutputTransform	BlockPropertyAnnotation
ArrayOfString	BeanHelper	BlockPropertyAnnotationGroup
ArrayOfString.Factory	BeanInfo	BlockPropertyAnnotationGroups
ASN1Constants	BeanManipulationTask	BlockPropertyMacro
ASN1Exception	BeanNotFoundException	BlockPropertyMacros
AspsModule	BeanProxy	BlockRegistry
Assignments	BeanStyleParametersCartDialog	BlocksFeatureShoppingCartDialog
AttachFilterToUser	BeanStylesTab	BlockSubMenuElementEditPropertiesPage
AttachUserToGroup	BeanTask	BlockSubMenuElementPropertiesPage

BlockTarget	Caches	CertificateServlet
BlockTarget.PathInfo	CacheSource	ChangePasswordBean
BlockToolbarAddElementWizard	CachesUpdateTimeTask	ChangePasswordGeneralTransform
BlockToolbarElementEditEventPage	CancelBrowse	ChangePasswordPreferencesTransform
BlockToolbarElementEditIndividualSelectionPage	CancelBrowse.Factory	ChangePasswordTask
BlockToolbarElementEditIndividualWizard	CancelBrowseResponse	ChangeUserPassword
BlockToolbarElementEditPropertiesPage	CancelBrowseResponse.Factory	ChangeUserPassword.Factory
BlockToolbarElementEditSelectionPage	CancelDocument	ChangeUserPasswordResponse
BlockToolbarElementEditWizard	CancelDocument.Factory	ChangeUserPasswordResponse.Factory
BlockToolbarElementEventPage	CancelDocumentResponse	CharsetEncoderWithSpecialChar
BlockToolbarElementPropertiesPage	CancelDocumentResponse.Factory	CharsetTranscoder
BlockToolbarElementResultsPage	CancelReport	CheckBoxComposite
BlockToolbarElementSelectionPage	CancelReport.Factory	CheckLatencyTask
BlockToolbarElementWizard	CancelReportResponse	CheckUserPrivilegesTask
BlockToolbarItemElementEditWizard	CancelReportResponse.Factory	ClassFilter
BlockToolbarItemsPage	CancelRSDocument	ClassFinder
BlockToolbarItemTreeObject	CancelRSDocument.Factory	ClassMethodTracingFilter
BlockToolbarResultsPage	CancelRSDocumentResponse	CleanUpModule
BlockToolbarSelectionPage	CancelRSDocumentResponse.Factory	ClientLog
BlockToolbarTreeObject	Cart	ClientSideDescriptorTag
BlockToolbarWizard	CartAttFormImpl	ClientSideDescriptorTagHelper
BlockTransform	CartDualAxisDistribution	ClientSideHelper
BlockTransformContext	CartElement	ClientSideLayout
BlockTransformExecutionFailure	CartElementAttribute	ClientSideLayoutCacheHint
BlockTransformFactory	CartElementBasic	ClientSideLayouts
BlockTransformScript	CartElementFolder	CloseSessionsTask
BlockTransformScriptInvalid	CartElementGeneric	CodeTaskBasicPage
BlockVisitor	CartElementPicker	ColHeader
BPMControllerParameter	CartElementPickerImpl	ColorGradientEditorBean
BPMDescriptor	CartElementRWDataField	ColorGradientEditorBeanImpl
BPMPreference	CartElements	ColorGradientEditorTransform
BridgeUtils	CartElementTree	ColorPickerEditorBean
BrowserNavigationListener	CartElementTU	ColorPickerEditorTransform
BrowserSettingDefinition	CartElementWebAttributeForm	ColTitle
BrowserSettingDefinitionList	CartElementWebElement	ComboAutoComplete
BrowserSettings	CartElementWebObjectInfo	ComboFieldEditor
BrowserSettingsListProperties	CartExpression	Compare
BrowserSettingsTreeObject	CartExpressionImpl	CompareGUI
Bundle	CartFontStyles	ComparisonUnresolvedException
BundleDescriptorTag	CartFontStylesImpl	ComponentBlock
BundleDescriptorTagHelper	CartImpl	ComponentInitializationContext
BundleEntry	CartObjectBrowser	ComponentInitializationContextImpl
BundleFileNameFilter	CartObjectBrowserImpl	CompoundFormatter
ByteArray	CaselnsensitiveRequestKeys	Compression
Cache	Categories	ConfigChangeObserver
Cache	Category	ConfigException
CacheBase	CellTag	ConfigFileManager
CacheDetails	CertificateAdminController	ConfigFileParser
CacheException	CertificateCollection	ConfigFileParser.ConfigObject
CacheHint	CertificateContainer	ConfigFilesConflictLogAction
CacheManipulationFailure	CertificateController	ConfigFilesConflictLogDialog
CacheManipulator	CertificateControllerRequestState	ConfigFilesUpgradeAction
CacheRegistry	CertificateProvider	Configurable
CacheResults	CertificateReference	ConfigurationElement

ConfigurationElementList	ContextTreeObject	CSSModule
ConfigurationFile	ControllerHelper	CSSNewPropertyDialog
ConfigurationFilesCache	ControllerHelper.BlockLibraryInfo	CSSPropertiesSearch
ConfigurationFilesCache.ConfigurationFileCacheHint	ControllerInfo	CSSPropertiesSearch.CSSRuleComparator
ConfigurationSource	ControllerInfoList	CSSProperty
ConfigurationUpgradePage	ControllerInputValidator	CSSPropertyDialog
ConfigurationUpgradeWizard	ControlToolbarBean	CSSRule
ConfigurationValue	ControlToolbarBeanImpl	CSSRuleDialog
ConfirmAddDialog	ControlToolbarTransform	CSSTreeObject
ConfirmOverwriteDialogBean	CookieBrowserSettings	CSSTreeViewListener
ConfirmOverwriteDialogTransform	CopyFilesModule	CSSView
Connect	CoreTagHelper	CSSViewContentProvider
Connect.Factory	Counter	CSSViewFilter
ConnectedServersTransform	CounterData	CSV2ADOResultsetConverter
ConnectionValueTag	CounterDataList	CustomCredentialMapper
ConnectionValueTagHelper	CountInfo	CustomCredentialMapper
ConnectResponse	CountSettings	CustomCredentialMapper
ConnectResponse.Factory	CountSummary	CustomCredentialMapper
ConnectToProject	CPULicenseDetails	CustomCredentialMapper
ConnectToProject.Factory	CreateAddonClassPropertiesPage	CustomDirectoryFieldEditor
ConnectToProjectResponse	CreateAddonResultsPage	CustomDocumentHandler
ConnectToProjectResponse.Factory	CreateAddonWizard	CustomEdit
ConsoleHandler	CreateConnectionMap	CustomEventHandler
ConstantPromptBean	CreateEditMapConditionPage	CustomGroupBean
ConstantPromptEditorTransform	CreateEditMapConditionWizard	CustomGroupCreationSample
ContactCartElement	CreateEventHandlerPropertiesPage	CustomizationInfo
ContactsBrowserBean	CreateEventHandlerResultsPage	CustomizationPlugin
ContactsCart	CreateFolderTask	Customizations
ContainerException	CreateGroup	CustomizationsAwareCache
ContainerServices	CreateGroupTask	CustomizationsListener
ContainerServicesContext	CreateMobileConfigurationTask	CustomListElementKeyConfig
ContainerServicesNamespace	CreateSecurityFilterWithWebObjects	CustomLoginESM
ContainerStringCollection	CreateSelectorControl	CustomLoginForm
ContentEncoder	CreateStyleFromCheatSheetAction	CustomLoginValidator
ContentEncoderChain	CreateTransformMethodsPage	CustomPromptsHandler
ContentTag	CreateTransformPropertiesPage	CustomReportHandler
ContentTagHelper	CreateTransformResultsPage	CustomReportHandlerForExtendedProps
ContentType	CreateTransformWizard	CustomReportHandlerForMaxRowsColsOnly
ContentTypes	CreateUser	CustomReportHandlerForPromptsOnly
ContentTypes.ContentType	CreateUserTask	CustomURLParameterMapper
ContentTypeTag	CreateXMLBasedJavaClassPage	CustomURLParameterMapper
ContentTypeTagHelper	CredentialAccess	CustomURLParameterMapper
ContextInfo	CredentialManager	CustomURLParameterMapper
ContextMenu	CredentialMapper	CustomWizardDialog
ContextMenu.DefaultContextMenuBuilder	CreditsBean	DAG
ContextMenuInfo	CryptoHash	DAG.AnnotatableObject
MenuItem	CryptoHTML	DAG.Edge
MenuItemBuilder	CryptoSessionKey	DAG.Vertex
MenuItemList	CSLayoutFileInputValidator	DataExplorerBean
ContextMenuManager	CSSAlphabetViewFilter	DataExplorerElementsWidgetTransform
MenusTag	CSSAutoCompletionProcessor	DataExplorerFolderWidgetTransform
MenusTagHelper	CSSFileTreeObject	DataExplorerTransform
	CssHTMLMappingHelper	DataExplorerWidgetTransform

DataHTMLTransform	DeployWidgetTask	DocumentDetailsTransform
DataProviderCartRenderer	DerInputStream	DocumentDetailsXHTMLTransform
DataSetTransform	DerivedElementBean	DocumentDetailsXHTMLTransform
DatasetObjectExplorerBean	DerivedElementBean.Host	DocumentExportTransform
DatasetObjectExplorerBeanImpl	DerivedElementBeanImpl	DocumentFastExportTransform
DatasetObjectExplorerEventHandler	DerivedElementCalculationTransform	DocumentFrameBean
DatasetObjectExplorerTransform	DerivedElementEventHandler	DocumentFrameElement
DataSetParserBean	DerivedElementFilterTransform	DocumentGridToolbarBlockLoaderTransform
DataSetTable	DerivedElementFormatEditorBean	DocumentImageServlet
DatasetTable.DatasetCell	DerivedElementFormatEditorTransform	DocumentInsertToolbarBlockLoaderTransform
DatasetTable.DatasetColumn	DerivedElementListTransform	DocumentPageComponentImpl
DatasetTable.DatasetRow	DerivedElementsBrowserBean	DocumentSetFlagsAddOn
DataSource	DerivedElementsBrowserBeanImpl	DocumentSimpleTransform
DataSourceEnumerator	DerivedElementsBrowserEventHandler	DocumentToolbarTransform
DatedFileHandler	DerivedElementsBrowserTransform	DocumentXMLResultsTask
DateTimeConverter	DerivedElementsEditorBean	Domain
DBConnection	DerivedElementsEditorTransform	DOMSupport
DBConnectionManipulator	DerivedElementTransform	DragAndDropHelper
DBConnectionResults	DesktopPageComponentImpl	DragAndDropItem
DBConnectionSource	DesktopPathTransform	DrillDocumentPromptElement
DBDisconnectionFailure	DiagnosticJsonRenderer	DrillDocumentPromptElement.Factory
Debug	DiagnosticsBean	DrillDocumentPromptElementResponse
DebugFlagsAddOn	DiagnosticsBeanEventHandler	DrillDocumentPromptElementResponse.Factory
DebugFlagsTag	DiagnosticsConfiguration	DrillEditorBean
DebugFlagsTagHelper	DiagnosticsDispatcher	DrillFilterEditorBean
DebugFormatter	DiagnosticsLogger	DrillFilterEditorTransform
DebugMonitorHandler	DiagnosticsLoggers	DrillReportPromptElement
DebugUser	DiagnosticsPropertiesTransform	DrillReportPromptElement.Factory
DebugUtils	DiagnosticsStatisticsTransform	DrillReportPromptElementResponse
DefaultBlockVisitor	DiagnosticsViewerTransform	DrillReportPromptElementResponse.Factory
DefaultCertificateProvider	DiagnosticsViewerTransformHelper	DrillRSDocumentPromptElement
DefaultContentHandler	DialogPathTransform	DrillRSDocumentPromptElement.Factory
DefaultCredentialAccess	Dict	DrillRSDocumentPromptElementResponse
DefaultErrorHandler	DimtyPromptBean	DrillRSDocumentPromptElementResponse.Factory
DefaultExternalSecurity	DirectExportBean	DSSXMLClassFactory
DefaultHiddenInputBuilderImpl	DirectExportTransform	DuplicateMobileConfigurationTask
DefaultMessageRouter	Disconnect	DynamicMenus
DefaultPageRefreshDialog	Disconnect.Factory	DynamicTag
DefaultPortletParmsAccess	DisconnectedContainerServices	DynTableCellTag
DefaultSharedPortletSpaceAccess	DisconnectedContainerServicesImpl	DynTableCellTagHelper
DefaultStyleRequestContext	DisconnectResponse	DynTableTag
DefaultSubscriptionWidget	DisconnectResponse.Factory	DynTableTagHelper
DefaultURIBuilderImpl	DisplayBeanTag	Edit_jsp
DefaultURIBuilderImpl	DisplayBeanTagHelper	EditableBean
DefaultViewBeanFeaturesImpl	DisplayCookiesTag	EditableFilter
Definition	DisplayCookiesTagHelper	EditableFormatter
DeleteElementDialog	DisplayGuiComponentTag	EditableHandler
DeleteMessageTask	DisplayGuiComponentTagHelper	EditableLogger
DeleteMobileConfigurationTask	DisplayTag	EditableObject
DeltaProperties	DisplayTagImpl	EditableObjectAppBean
DeltaProperties.Visitor	DisplayUnitsVisitor	EditableObjectBean
DemoManager	DivBasedCartImpl	Editor
DenyDomains	DocumentBean	EditorImpl
DeployDescLoader	DocumentBinaryResultsTask	EditorToolbarTransform

EditPromptPageComponentImpl	EnumDiagnosticsBeanEvents
ElementComboModifyListener	EnumDimensionalityLevel
ElementPickerBean	EnumDocumentBeanEvents
ElementPickerBeanImpl	EnumDocumentFrameEvents
ElementPickerTransform	EnumDragAndDropProperties
ElementPromptBean	EnumDrillEditorEvents
ElementsBrowseTask	EnumDrillFilterEditorEvents
ElementsPromptEditorTransform	EnumDSSourceManipulationSchema
ElseTag	EnumDSSXML3DConnGrpMinorTypes
EmptyListException	EnumDSSXML3DConnSeriesMinorTypes
EmptyNamespaceEncoderImpl	EnumDSSXML3DFloatMinorTypes
EmptyNumericStringPreferenceDefinition	EnumDSSXML3DRiserMinorTypes
EncodedMarkupOutput	EnumDSSXML3DScatterMinorTypes
EncodeSpecialStreamingContext	EnumDSSXML3DSurfaceMinorTypes
Encryptor	EnumDSSXMLAccessEntryType
EndPointXMLConstants	EnumDSSXMLAccessRightFlags
EndPointXMLFactory	EnumDSSXMLAEGgregation
EnumAddressListBeanEvents	EnumDSSMLAggregation
EnumAdminBeanEvents	EnumDSSMLApplicationType
EnumAnnotationBeanEvents	EnumDSSMLAreaShapes
EnumAppBeanViewModes	EnumDSSMLAttributeDirection
EnumAppComponentEvents	EnumDSSMLAttributeLockType
EnumAppWebFeatures	EnumDSSMLAuditorTypes
EnumAttributeBeanEvents	EnumDSSMLAuditUserFilter
EnumAttributeFormsEditorEvents	EnumDSSMLAuthModes
EnumBeanErrorCodes	EnumDSSMLAuthModes
EnumBeanOutputFormat	EnumDSSMLAutoTextType
EnumBeanPersistMode	EnumDSSMLAxesBitMap
EnumBlockPropertyTypes	EnumDSSMLAxisName
EnumBrowserType	EnumDSSMLBaseFormType
EnumCartProperties	EnumDSSMLBinaryContextDetailFlags
EnumCertificateErrorCodes	EnumDSSMLBinaryContextFlags
EnumChangePasswordEvents	EnumDSSMLBoxPlotMinorTypes
EnumCipher	EnumDSSMLBubbleMinorTypes
EnumColorPickerBeanEvents	EnumDSSMLCacheAdminAction
EnumConfigFileChangeEvent	EnumDSSMLCacheExpCommand
EnumConstantPromptRestrictionTypes	EnumDSSMLCacheExpOptions
EnumContactBrowserBeanEvents	EnumDSSMLCacheInfo
EnumContactsCartProperties	EnumDSSMLCacheSetting
EnumContextMenuItems	EnumDSSMLCacheSwapPolicy
EnumCSSClasses	EnumDSSMLCacheType
EnumCustomDocumentBeanEvents	EnumDSSMLChannelType
EnumCustomGroupBeanEvents	EnumDSSMLCharacterEncoding
EnumCustomReportBeanEvents	EnumDSSMLClientTypes
EnumCustomReportBeanEvents	EnumDSSMLClusterMemberStatus
EnumCustomWizardStepBeanEvents	EnumDSSMLColumnClass
EnumDataExplorerBeanEvents	EnumDSSMLCombinationMinorTypes
EnumDatasetObjectExplorerBeanEvents	EnumDSSMLConnectionMode
EnumDefaultServerProperties	EnumDSSMLConnectionState
EnumDerivedElementBeanEvents	EnumDSSMLConnParam
EnumDerivedElementsBrowserEvents	EnumDSSMLCurrentElementAction
EnumDerivedElementsEditorEvents	EnumDSSMLCurrentElementStatus
EnumDerivedElementType	EnumDSSMLDatabaseType
EnumDeviceTypes	EnumDSSMLDatabaseVersion
	EnumDSSXMLDataLineType
	EnumDSSXMLDataSourceConnectionType
	EnumDSSXMLDataSourceFlags
	EnumDSSXMLDataSourceType
	EnumDSSXMLDataType
	EnumDSSXMLDayOfWeek
	EnumDSSXMLDayOfWeekBitEncoded
	EnumDSSMLDBConnectionCacheOption
	EnumDSSMLDBConnectionDriverMode
	EnumDSSMLDBConnectionExecutionMode
	EnumDSSMLDBConnectionInfo
	EnumDSSMLDBConnectionMultiProcessOption
	EnumDSSMLDBConnectionStatus
	EnumDSSMLDefaultFormats
	EnumDSSMLDerivedElementSaveAsFlags
	EnumDSSMLDimtyUnitType
	EnumDSSMLDisplayMode
	EnumDSSMLDocExecutionFlags
	EnumDSSMLDocResultFlags
	EnumDSSMLDocSaveAsFlags
	EnumDSSMLDrillImportance
	EnumDSSMLDrillOption
	EnumDSSMLDrillQual
	EnumDSSMLDrillType
	EnumDSSMLDrillUnitSource
	EnumDSSMLDynamicTime
	EnumDSSMLElementType
	EnumDSSMLERType
	EnumDSSMLExecutionFlags
	EnumDSSMLExportFormat
	EnumDSSMLExpressionType
	EnumDSSMLExternalSourceFlags
	EnumDSSMLExtnType
	EnumDSSMLFieldGroupDataLevel
	EnumDSSMLFiltering
	EnumDSSMLFilterType
	EnumDSSMLFolderNames
	EnumDSSMLFolderType
	EnumDSSMLFontAlignTypes
	EnumDSSMLFontAntiAlias
	EnumDSSMLFontStyles
	EnumDSSMLFunction
	EnumDSSMLFunctionsFlags
	EnumDSSMLFunnelMinorTypes
	EnumDSSMLGanttMinorTypes
	EnumDSSMLGaugeMinorTypes
	EnumDSSMLGetClusterMembershipFlags
	EnumDSSMLGetScheduleFlags
	EnumDSSMLGraphAttributeID
	EnumDSSMLGraphGaugeBorderStyle
	EnumDSSMLGraphGaugeNeedleStyle
	EnumDSSMLGraphGaugeStyle
	EnumDSSMLGraphImgTypes
	EnumDSSMLGraphMarkerShape

EnumDSSXMLGraphObjectID	EnumDSSXMLNodeDimty	EnumDSSXMLRWFieldType
EnumDSSXMLGraphRiserEmphasisType	EnumDSSXMLNodeState	EnumDSSXMLRWGroupByUnitType
EnumDSSXMLAreaMinorTypes	EnumDSSXMLNodeType	EnumDSSXMLRWLinkType
EnumDSSXMLBarMinorTypes	EnumDSSXMLNotificationDataID	EnumDSSXMLRWManipulationMethod
EnumDSSXMLHiloStockMinorTypes	EnumDSSXMLNotificationModuleID	EnumDSSXMLRNNodeTransactionChangeType
EnumDSSXMLHistogramMinorTypes	EnumDSSXMLObjectFlags	EnumDSSXMLRWNodeType
EnumDSSXMLHLineMinorTypes	EnumDSSXMLObjectLockCommands	EnumDSSXMLRWOptions
EnumDSSXMLIDType	EnumDSSXMLObjectLockFlags	EnumDSSXMLRWPageByStyle
EnumDSSXMLInboxDeleteFlags	EnumDSSXMLObjectSearchStatus	EnumDSSXMLRWPreLoadPanelState
EnumDSSXMLInboxFlags	EnumDSSXMLObjectState	EnumDSSXMLWSectionType
EnumDSSXMLInitRepositoryModes	EnumDSSXMLObjectSubTypes	EnumDSSXMLWSelectorUnsetStatus
EnumDSSXMLJobInfo	EnumDSSXMLObjectTypes	EnumDSSXMLWTemplateNodeOrigin
EnumDSSXMLJobStatus	EnumDSSXMLOrder	EnumDSSXMLRWTreeType
EnumDSSXMLJoinType	EnumDSSXMLOrderParentFirst	EnumDSSXMLScaleTypes
EnumDSSXMLKeepIndependent	EnumDSSXMLParetoMinorTypes	EnumDSSXMLScatterMinorTypes
EnumDSSXMLKeepSort	EnumDSSXMLPartialType	EnumDSSXMLScheduleFilterFlags
EnumDSSXMLLDAPCallConvention	EnumDSSXMLPerformanceMonitorFlags	EnumDSSXMLScheduleTriggerType
EnumDSSXMLLDAPImport	EnumDSSXMLPieMinorTypes	EnumDSSXMLSearchDomain
EnumDSSXMLLDAPPPlatform	EnumDSSXMLPolarMinorTypes	EnumDSSXMLSearchFlags
EnumDSSXMLLDAPServerSecureConnection	EnumDSSXMLPrivilegeCategoryTypes	EnumDSSXMLSearchScope
EnumDSSXMLLDAPStringCoding	EnumDSSXMLPrivilegeTypes	EnumDSSXMLSearchUnit
EnumDSSXMLLDAPSync	EnumDSSXMLPrivilegeUpdateTypes	EnumDSSXMLSectionElementType
EnumDSSXMLLDAPVendor	EnumDSSXMLProductLicenseStatus	EnumDSSXMLSectionTreeType
EnumDSSXMLLDAPVerification	EnumDSSXMLProjectActions	EnumDSSXMLServerDefManipulationMethod
EnumDSSXMLLDAPVersion	EnumDSSXMLProjectConfigurationSettingType	EnumDSSXMLServerProjectReportCacheDB-MatchingFlag
EnumDSSXMLLegendLock	EnumDSSXMLProjectLoadOption	EnumDSSXMLServerProjectSettingID
EnumDSSXMLLevelFlags	EnumDSSXMLProjectStatus	EnumDSSXMLServerSettingID
EnumDSSXMLLicenseComplianceCategory	EnumDSSXMLPromptAnswerReuse	EnumDSSXMLSessionFlags
EnumDSSXMLLicenseScheme	EnumDSSXMLPromptReuse	EnumDSSXMLSessionFlags
EnumDSSXMLLicenseStatus	EnumDSSXMLPromptType	EnumDSSXMLShapeType
EnumDSSXMLLicenseType	EnumDSSXMLPropertyFlags	EnumDSSXMLShowAxis
EnumDSSXMLLinePattern	EnumDSSXMLPropertyGroupTypes	EnumDSSXMLSortType
EnumDSSXMLLockQuery	EnumDSSXMLPropertyXmlFlags	EnumDSSXMLSourceFeatures
EnumDSSXMLLockState	EnumDSSXMLPurgeFlag	EnumDSSXMLSourceManipulation-Commands
EnumDSSXMLMajorGraphTypes	EnumDSSXMLRadarMinorTypes	EnumDSSMLSQLType
EnumDSSXMLManipulationType	EnumDSSXMLReportCacheDisableOptions	EnumDSSXMLStatisticDataID
EnumDSSXMLMDUpdateCommands	EnumDSSXMLReportCacheStatus	EnumDSSXMLStatisticModuleID
EnumDSSXMLMDUpdateFlags	EnumDSSXMLReportCacheType	EnumDSSXMLStatisticTableID
EnumDSSXMLMDVersion	EnumDSSXMLReportFilter	EnumDSSXMLStatus
EnumDSSXMLMessageDuplicationMode	EnumDSSXMLReportManipulationMethod	EnumDSSXMLStepTypes
EnumDSSXMLMessageStatus	EnumDSSXMLReportObjects	EnumDSSXMLSubscriptionContactType
EnumDSSXMLMessageTypes	EnumDSSXMLReportSaveAsFlags	EnumDSSXMLSubscriptionDeliveryType
EnumDSSXMLMetricFormulaType	EnumDssXMLReportState	EnumDSSXMLSubscriptionMobileClientType
EnumDSSXMLMetricType	EnumDSSXMLReportTypes	EnumDSSXMLSubtotalsPosition
EnumDSSXMLMonitorFilterOperator	EnumDSSXMLRequestTypes	EnumDSSXMLSubtotalStyle
EnumDSSXMLMonitorType	EnumDSSXMLResult2Flags	EnumDSSXMLSymbol
EnumDSSXMLMonth	EnumDSSXMLResult3Flags	EnumDSSXMLSystemLinks
EnumDSSMLNCSAllowChangeMaskType	EnumDSSXMLResultFlags	EnumDSSXMLTableExtraInformation
EnumDSSMLNCSContactSourceField	EnumDSSXMLRWControlActionType	EnumDSSMLTemplateSubtotalType
EnumDSSMLNCEmailDestinationType	EnumDSSXMLRWControlSubsequentAction	EnumDSSMLTemplateUnitType
EnumDSSMLNCSInstanceSourceField	EnumDSSXMLRWControlType	EnumDSSMLThresholdScope
EnumDSSMLNCSErrorByContactField	EnumDSSXMLRWDCacheOptions	EnumDSSMLTimeOccurrenceType
EnumDSSMLNCSSubscriptionType	EnumDSSXMLRWEaseCreationFlags	EnumDSSMLTriggerOptions
EnumDSSMLNetProtocol	EnumDSSXMLRWExportFlags	EnumDSSMLTrustWebServerStatus

EnumDSSXMLUnitTransactionFlags	EnumGraphFormatToolbarEvents	EnumMWSAttributeFormsSetting
EnumDSSXMLUpdateTransactionDataType	EnumGridAutoStylesEvents	EnumMWSAttributeFormsSetting.Factory
EnumDSSXMLUserAccountServiceMethod	EnumGridCellLevels	EnumMWSAttributeFormsSettingJBridge
EnumDSSXMLUserAnswerCommands	EnumGridCellMapAttributes	EnumMWSAttributeLockType
EnumDSSXMLUserConnectionInfo	EnumGridFormatEditorEvents	EnumMWSAttributeLockType
EnumDSSXMLUserFilter	EnumGridFormatEditorTabs	EnumMWSAttributeLockType.Factory
EnumDSSXMLValidationLevel	EnumGridFormatEvents	EnumMWSAttributeLockTypeJBridge
EnumDSSXMLVAreaMinorTypes	EnumGridFormatGenericValues	EnumMWSAuthMode
EnumDSSXMLVBarMinorTypes	EnumGridFormatNumberCategory	EnumMWSAuthMode
EnumDSSXMLViewMedia	EnumGridFormatToolbarEvents	EnumMWSAuthMode.Factory
EnumDSSXMLVLineMinorTypes	EnumGridFormatUnit	EnumMWSAuthModeJBridge
EnumDSSXMLWidthScenario	EnumGridHorizontalAlignmentType	EnumMWSBrowseStatus
EnumDSSXMLXDAType	EnumGridNegativeNumbersFormat	EnumMWSBrowseStatus
EnumEditableObjectAppBeanEvents	EnumGridNumberCurrencyPosition	EnumMWSBrowseStatus.Factory
EnumEditableObjectBeanEvents	EnumGridTextDirectionType	EnumMWSBrowseStatusJBridge
EnumEditorDisplayFlags	EnumGridVerticalAlignmentType	EnumMWSConnectProperties
EnumEditorElements	EnumGroupBeanEvents	EnumMWSConnectProperties
EnumEditorSectionTypes	EnumGroupbyPropertiesEditorEvents	EnumMWSConnectProperties.Factory
EnumEnablePushNotification	EnumGroupPathEvents	EnumMWSConnectPropertiesJBridge
EnumEventElement	EnumGuiElements	EnumMWSConnectProperty
EnumEventHandlerTypes	EnumHTMLAttributes	EnumMWSConnectProperty
EnumExcelVersion	EnumHTMLTags	EnumMWSConnectProperty.Factory
EnumExecutionMode	EnumImportWizardBeanEvents	EnumMWSConnectPropertyJBridge
EnumExecutionScope	EnumInboxBeanEvents	EnumMWSDataType
EnumExportFormats	EnumIncrementalFetchProperties	EnumMWSDataType
EnumExportPlaintextDelimiters	EnumJSONAttributes	EnumMWSDataType.Factory
EnumExportSection	EnumLiferayPortletParameterNames	EnumMWSDataTypeJBridge
EnumExpressionBeanEvents	EnumLinkAnswerMode	EnumMWSElementRestriction
EnumExpressionBeanType	EnumLocaleResolution	EnumMWSElementRestriction
EnumExpressionCartTransformProperties	EnumLocalizationType	EnumMWSElementRestriction.Factory
EnumExpressionEditMode	EnumLogDestinationType	EnumMWSElementRestrictionJBridge
EnumExpressionPromptBeanTypes	EnumLoggingLevels	EnumMWSElementSourceFlags
EnumFillColorEffects	EnumMenulImages	EnumMWSElementSourceFlags
EnumFilterBeanEvents	EnumMobileControllerErrorSource	EnumMWSElementSourceFlags.Factory
EnumFilterEditorErrorCodes	EnumMSTRCredentialType	EnumMWSElementSourceFlags_type0
EnumFilterEditorEvents	EnumMSTRPortletType	EnumMWSElementSourceFlags_type0.Factory
EnumFilterElementEvents	EnumMWSAccessRights	EnumMWSElementSourceFlagsJBridge
EnumFilterTypes	EnumMWSAccessRights	EnumMWSExecutionFlags
EnumFolderBeanEvents	EnumMWSAccessRights.Factory	EnumMWSExecutionFlags
EnumFolderFrameBeanEvents	EnumMWSAccessRights_type0	EnumMWSExecutionFlags.Factory
EnumFormatBorderFlags	EnumMWSAccessRights_type0.Factory	EnumMWSExecutionFlags_type0
EnumFormatOutputTypes	EnumMWSAccessRightsJBridge	EnumMWSExecutionFlags_type0.Factory
EnumFormatTabManagerEvents	EnumMWSAttachmentType	EnumMWSExecutionFlagsJBridge
EnumFormatToolbarEvents	EnumMWSAttachmentType	EnumMWSExecutionObjType
EnumFormulaBarEvents	EnumMWSAttachmentType.Factory	EnumMWSExecutionObjType
EnumFrameEditorEvents	EnumMWSAttachmentTypeJBridge	EnumMWSExecutionObjType.Factory
EnumFrameEvents	EnumMWSAttributeFormCategories	EnumMWSExecutionObjTypeJBridge
EnumGenericEvents	EnumMWSAttributeFormCategories	EnumMWSExportSetting
EnumGradientShadingStyles	EnumMWSAttributeFormCategories.Factory	EnumMWSExportSetting
EnumGradientTransitionTypes	EnumMWSAttributeFormCategories_type0	EnumMWSExportSetting.Factory
EnumGraphFormatAxisParameters	EnumMWSAttributeFormCategories_type0.	EnumMWSExportSettingJBridge
EnumGraphFormatEditorEvents	Factory	EnumMWSFunctionalityProperty
EnumGraphFormatEditorTabs	EnumMWSAttributeFormCategoriesJBridge	EnumMWSFunctionalityProperty
EnumGraphFormatTextParameters	EnumMWSAttributeFormsSetting	EnumMWSFunctionalityProperty.Factory

EnumMWSFunctionalityPropertyJBridge	EnumMWSResultFlags	EnumMWSVisualizationScope_type0
EnumMWSGraphType	EnumMWSResultFlags.Factory	EnumMWSVisualizationScope_type0.Factory
EnumMWSGraphType	EnumMWSResultFlags_type0	EnumMWSVisualizationScopeJBridge
EnumMWSGraphType.Factory	EnumMWSResultFlags_type0.Factory	EnumObjectBeanEvents
EnumMWSGraphTypeJBridge	EnumMWSResultFlagsJBridge	EnumObjectBrowserBeanEvents
EnumMWSHistoryListAction	EnumMWSRSExecutionFlags	EnumObjectBrowserContexts
EnumMWSHistoryListAction	EnumMWSRSExecutionFlags	EnumObjectBrowserManagerBeanEvents
EnumMWSHistoryListAction.Factory	EnumMWSRSExecutionFlags.Factory	EnumObjectBrowserTabs
EnumMWSHistoryListActionJBridge	EnumMWSRSExecutionFlags_type0	EnumObjectBrowserTargets
EnumMWSHistoryMessageStatus	EnumMWSRSExecutionFlags_type0.Factory	EnumObjectExplorerBeanEvents
EnumMWSHistoryMessageStatus	EnumMWSRSExecutionFlagsJBridge	EnumObjectManipulationDialogEvents
EnumMWSHistoryMessageStatus.Factory	EnumMWSRSExportSetting	EnumObjectPromptBeanEvents
EnumMWSHistoryMessageStatus_type0	EnumMWSRSExportSetting	EnumObjectSelectionBeanEvents
EnumMWSHistoryMessageStatus_type0.Factory	EnumMWSRSExportSetting.Factory	EnumOMDCreateFolderEvents
EnumMWSHistoryMessageStatusJBridge	EnumMWSRSExportSettingJBridge	EnumOraclePortletParameterNames
EnumMWSHistoryRequestStatus	EnumMWSRSResultFlags	EnumPageByBeanEvents
EnumMWSHistoryRequestStatus	EnumMWSRSResultFlags	EnumPageEvents
EnumMWSHistoryRequestStatus.Factory	EnumMWSRSResultFlags.Factory	EnumPageInfoDirection
EnumMWSHistoryRequestStatusJBridge	EnumMWSRSResultFlags_type0	EnumPageResourceManagerModes
EnumMWSInfoProperty	EnumMWSRSResultFlags_type0.Factory	EnumParameterType
EnumMWSInfoProperty	EnumMWSRSResultFlagsJBridge	EnumPerformanceLogLevel
EnumMWSInfoProperty.Factory	EnumMWSRSSimpleResultFlag	EnumPortletContentIdentifier
EnumMWSInfoPropertyJBridge	EnumMWSRSSimpleResultFlag.Factory	EnumPortletContentType
EnumMWSObjectType	EnumMWSRunStatusType	EnumPortletDescriptorNames
EnumMWSObjectType	EnumMWSRunStatusType	EnumPortletLocaleID
EnumMWSObjectType.Factory	EnumMWSRunStatusType.Factory	EnumPortletParameterNames
EnumMWSObjectTypeJBridge	EnumMWSRunStatusTypeJBridge	EnumPortletParms
EnumMWSObjSubType	EnumMWSSearchFlags	EnumPreferenceLevels
EnumMWSObjSubType	EnumMWSSearchFlags	EnumPreferencesEvents
EnumMWSObjSubType.Factory	EnumMWSSearchFlags.Factory	EnumPrintEvents
EnumMWSObjSubTypeJBridge	EnumMWSSearchFlags_type0	EnumProductLicenseStatuses
EnumMWSProjectFolder	EnumMWSSearchFlags_type0.Factory	EnumProducts
EnumMWSProjectFolder	EnumMWSSearchFlagsJBridge	EnumProjectBrowserEvents
EnumMWSProjectFolder.Factory	EnumMWSSearchRestriction	EnumPromptAnswerBeanEvents
EnumMWSProjectFolderJBridge	EnumMWSSearchRestriction	EnumPromptAnswerFormat
EnumMWSProjectSessionProperty	EnumMWSSearchRestriction.Factory	EnumPromptAnswerOptions
EnumMWSProjectSessionProperty	EnumMWSSearchRestrictionJBridge	EnumPromptBeanEvents
EnumMWSProjectSessionProperty.Factory	EnumMWSSyncCommunicationSetting	EnumPromptDefinitionAppBeanEvents
EnumMWSProjectSessionPropertyJBridge	EnumMWSSyncCommunicationSetting	EnumPromptDefinitionBeanEvents
EnumMWSProjectSetting	EnumMWSSyncCommunicationSetting.Factory	EnumPromptExpressionBeanEvents
EnumMWSProjectSetting	EnumMWSSyncCommunicationSettingJBridge	EnumPromptLevelFlags
EnumMWSProjectSetting.Factory	EnumMWSUserPrivilege	EnumPromptObjectSelectionMode
EnumMWSProjectSettingJBridge	EnumMWSUserPrivilege	EnumPromptPropertyNameAndValues
EnumMWSProjectWellKnownObj	EnumMWSUserPrivilege.Factory	EnumPromptsBeanEvents
EnumMWSProjectWellKnownObj	EnumMWSUserPrivilegeJBridge	EnumPromptsBeanTypes
EnumMWSProjectWellKnownObj.Factory	EnumMWSViewMedia	EnumPromptsSourceTypes
EnumMWSProjectWellKnownObjJBridge	EnumMWSViewMedia	EnumQuickSymbol
EnumMWSProperties	EnumMWSViewMedia.Factory	EnumReportBeanEvents
EnumMWSProperties	EnumMWSViewMedia_type0	EnumReportDesignMode
EnumMWSProperties.Factory	EnumMWSViewMedia_type0.Factory	EnumReportFrameEvents
EnumMWSPropertiesJBridge	EnumMWSViewMediaJBridge	EnumReportGridDisplayCellTypes
EnumMWSRequestType	EnumMWSVisualizationScope	EnumReportPageType
EnumMWSResultFlags	EnumMWSVisualizationScope.Factory	EnumReportThresholdSymbols

EnumReportViewModes	EnumThresholdTypes	EnumWebReportExecutionModes
EnumRequestStatus	EnumToolbarBeanEvents	EnumWebReportExportModes
EnumRibbonEvents	EnumToolbarSetBeanEvents	EnumWebReportSourceType
EnumRWBeanEvents	EnumTriStateIndicator	EnumWebReportViewMode
EnumRWControlStyle	EnumUserBeanEvents	EnumWebRowValueType
EnumRWExecutionModes	EnumUserContactBrowserBeanEvents	EnumWebScheduleSort
EnumRWExportModes	EnumUserEntitiesBeanEvents	EnumWebServicesConstants
EnumRWExportRanges	EnumUserEntitiesBrowserEvents	EnumWebSessionType
EnumRWFormatTypes	EnumUserEntitiesSelectorEvents	EnumWebStateFlags
EnumRWFrameEvents	EnumUserEntitiesSourceTypes	EnumWebStateLevel
EnumRWGraphDropZoneEditorEvents	EnumUserEntityBeanEvents	EnumWebStatementType
EnumRWGroupByLevels	EnumUserSearchBeanEvents	EnumWebSubscriptionContentCompressionStatus
EnumRWGroupByTypes	EnumValueFormatter	EnumWebSubscriptionContentFormatModes
EnumRWHeightWidthMode	EnumVariantTypes	EnumWebSubscriptionContentFormatTypes
EnumRWHTMLFieldType	EnumViewBeanEvents	EnumWebSubscriptionContentTypes
EnumRWItemWidthMode	EnumWebAppDebugFlags	EnumWebSubscriptionDeliveryMode
EnumRWMouseOverAction	EnumWebAppErrorCodes	EnumWebSubscriptionDeviceSubtype
EnumRWOrientation	EnumWebAttributeFormNamesDisplay	EnumWebSubscriptionEditModes
EnumRWProjectBrowserBeanEvents	EnumWebBrowserSettings	EnumWebSubscriptionObjectTypes
EnumRWResultFilterModes	EnumWebConfigBeanType	EnumWebToggleFunction
EnumRWUnitTypes	EnumWebControlDataFilterModes	EnumWebTransformableType
EnumRWViewModes	EnumWebControllers	EnumWebVisualizationViewModes
EnumSaveAsEvents	EnumWebControlSubType	EnumWebWorkingSetSortByField
EnumSaveObjectAsDialogEvents	EnumWebCustomGroupDisplayOptions	EnumWizardStepBeanEvents
EnumScheduleBeanEvents	EnumWebCustomGroupReportFilterInteraction	EnumWriteBackInfoEventsAndArgs
EnumSearchTypeConstants	EnumWebDependentObjectType	EnumWSFObjectType
EnumSearchWildcards	EnumWebDimensionDisplayStyle	EnumWSFObjSubType
EnumSecurityFilterBeanEvents	EnumWebDisplayedForms	EnumWSFSearchFlags
EnumSecurityFilterFrameBeanEvents	EnumWebDocumentViewMode	EnumWSFSearchRestriction
EnumSecurityPluginProviders	EnumWebDrillProperties	EnumXDADataExplorerBeanEvents
EnumSecurityRoleEvents	EnumWebElementSourceType	EnumXmlPriorityType
EnumSecurityRoleUserSelectorEvents	EnumWebFeatures	EnumXmlStatePhase
EnumServantTarget	EnumWebFolderDisplayStyle	ErrorCodeTag
EnumServerState	EnumWebFormatType	ErrorCodeTagHelper
EnumServerTypes	EnumWebFunctionType	ErrorDialog
EnumServiceMode	EnumWebGraphLabelType	ErrorInfo
EnumServletEvents	EnumWebHeaderType	ErrorInfo.ErrorMessage
EnumSortBy	EnumWebLimitSummaryFlags	ErrorInfo.ErrorTitle
EnumSortEditorEvents	EnumWebMessageSortByField	ErrorInfoList
EnumSubscriptionBeanEvents	EnumWebMonitorStatistics	ErrorMessageHandler
EnumSubscriptionContactBrowserBeanEvents	EnumWebMonitorType	ErrorMessageTag
EnumSubscriptionEditorEvents	EnumWebMRPFunction	ErrorMessageTagHelper
EnumSubscriptionFolderBeanEvents	EnumWebNamedUsers	ErrorMsgLib
EnumSubsetUnitLimitElementEvents	EnumWebObjectsFeatures	ErrorMsgLib
EnumSubtotalCheckboxState	EnumWebObjectSort	ErrorsListProperties
EnumSubtotalInstanceType	EnumWebODBCVersion	ErrorsTreeObject
EnumSubtotalsEditorEvents	EnumWebPages	ErrorValueTag
EnumSubtotalState	EnumWebParameterReuseSessionValues	ErrorValueTagHelper
EnumSysBeanNames	EnumWebParameters	ESRIVisualizationPropertiesTransform
EnumSysBeanTypes	EnumWebPersistableState	EstablishTrust
EnumTabManagerBeanEvents	EnumWebPreferences	EstablishTrust.Factory
EnumTemplateFilterExecEvents	EnumWebPromptAnswerMode	EstablishTrustResponse
EnumTextBeanEventsAndArgs	EnumWebPromptType	EstablishTrustResponse.Factory
EnumThresholdEditorEvents	EnumWebPropertySource	

EventArgsInfo	ExportFormatsListProperties	FileUtils
EventArgsInfoList	ExportFormatsTreeObject	Filter
EventArgsTag	ExportOptionsEditorBean	FilterAllObjectBrowserTransform
EventFlagList	ExportOptionsEditorTransform	FilterAnd
EventFlagListSet	ExportPlaintextDelimiter	FilterAndNot
EventHandlerInfo	ExportPlaintextDelimiterList	FilterAppBean
EventHandlerInfo	ExportReportAddOn	FilterBean
EventHandlerList	ExportReportPreferencePropertyListMapper	FilterDetailsEditorTransform
EventHandlersCatalog	ExportReportTask	FilterDetailsFormatting
EventHandlerTreeObject	ExportSaveReportPropertiesAddOn	FilterDetailsFormattingEditorBean
EventHelper	ExportSetContentTypeTag	FilterDetailsFormattingEditorBeanImpl
EventInfo	ExportSetContentTypeTagHelper	FilterDetailsFormattingEditorTransform
EventInfo	ExportTransform	FilterEditorBean
EventInfoList	ExpressionBean	FilterElementBean
EventList	ExpressionBeanContext	FilterElementRWTransform
EventListSet	ExpressionDisplayBean	FilterElementTransform
EventManager	ExpressionFilterTransform	FilterEventHandler
EventMap	ExpressionGenericTransform	FilterExpressionTransform
EventMap.EventMapArgument	ExpressionHelper	FilterFrameBean
EventMapArgsPage	ExpressionMessages	FilterObjectBrowserAddOn
EventMapList	ExpressionObject	FilterObjectBrowserTransform
EventMapResultsPage	ExpressionParts	FilterOr
EventMapSelectionPage	ExpressionPartsImpl	FilterOrNot
EventMapSelectionPage.	ExpressionPlainTextTransform	FilterTransform
EventMapIdComparator	ExpressionPromptBean	FlagInfo
EventMapWizard	ExpressionPromptEditorTransform	FlashTabBean
EventModifyListener	ExpressionPromptEventHandlerImpl	FlatFetchSettings
EventSettingsProperties	ExpressionStrings	FlatStateDelimiters
EventsModule	ExpressionStringsImpl	FlatStateSerializer
EventTag	ExtendedFormatter	FlatStateTokenizer
EventTagHelper	ExtendedFormatter2	FlatStateUtil
EventTreeObject	ExtendedFormatterEx	FlattenedEnumerationBase
ExecuteDocument	ExtensionMapper	FlattenedTreeNode
ExecuteDocument.Factory	ExternalPromptsSource	FlattenedTreeNodeImpl
ExecuteDocumentResponse	ExternalSecurity	FloatingEditorBlockTransform
ExecuteDocumentResponse.Factory	FailureDialog	FolderAddDataSetExplorerTransform
ExecuteIncrementalBrowse	FastExportAddOn	FolderAddDataSetTransform
ExecuteIncrementalBrowse.Factory	FastReportPDFExportAddOn	FolderAdminBrowserTransform
ExecuteIncrementalBrowseResponse	FeatureArray	FolderBean
ExecuteIncrementalBrowseResponse.	FeaturesHelper	FolderBrowseTask
Factory	FeatureShoppingCartDialog	FolderBrowsingSample
ExecuteReport	FieldPickerDialog	FolderBulletReportsTransform
ExecuteReport.Factory	FileCacheBase	FolderBulletRWTransform
ExecuteReportResponse	FileContentsObjectList	FolderBulletTransform
ExecuteReportResponse.Factory	FileContentsObjectWrapper	FolderCopy
ExecuteRSDocument	FileDispenser	FolderCreateAnalysisFromReport
ExecuteRSDocument.Factory	FileFinder	FolderCreateDocumentFromAnalysis
ExecuteRSDocumentResponse	FileFinderEntityResolver	FolderCreateDocumentFromReport
ExecuteRSDocumentResponse.Factory	FileHandler	FolderCreateDocumentFromReport
ExportBean	FileLoader	FolderCreateDocumentTransform
ExportBeanHelper	FileLoaderEntityResolver	FolderCreateReportTransform
ExportDocumentTask	FileLogDestination	FolderCreateReportXHTMLTransform
ExportFormat	FileOpenDialog	FolderCreateShortcut
ExportFormatList	FileUtil	FolderCreationSample

FolderDelete	FolderSearchResultsXHTMLTransform	GenericEventHandler
FolderDeleteObjectTransform	FolderSearchResultsXHTMLTransform	GenericEventHandler.GenericEventKeys
FolderDetailsTransform	FolderSetFlagsAddOn	GenericFileModule
FolderDisplayProperties	FolderSimpleAction	GenericFileModule.PatternInfo
FolderEdit	FolderSmallIconTransform	GenericFormEditor
FolderEdit	FolderSummaryCreateDocumentTransform	GenericFormEditor.ProgressMonitor
FolderExportLink	FolderSummaryCreateReportTransform	GenericFormPage
FolderFrameAddOn	FolderSummaryTransform	GenericJsonGenerator
FolderFrameBean	FolderSummaryViewTransform	GenericMultipleEvents
FolderFrameEventHandler	FolderTreeExportLink	GenericRequestKeys
FolderFrameTreeTransform	FolderTreeTransform	GenericTreeNode
FolderIconRadioButtonTransform	FolderTreeTransform	GenericTreeNodeImpl
FolderIconTransform	FolderViewTransform	GenericTreeObject
FolderInlineRename	FolderViewTransform	GenericWebAppController
FolderLastVisitedAddOn	FolderViewTransform.ClientEvent	GenericWebEvent
FolderLink	FolderViewWidgetTransform	GetAttributeChildren
FolderLinksListProperties	FolderViewWidgetTransform.Tree	GetAttributeChildren.Factory
FolderListTransform	FolderViewXMLTransform	GetAttributeChildrenResponse
FolderMapping	FormalParameter	GetAttributeChildrenResponse.Factory
FolderMappingList	FormalParameterAnnotation	GetAttributeElements
FolderMappingListProperties	FormalParameterImpl	GetAttributeElements.Factory
FolderMappingTreeObject	FormalParameters	GetAttributeElementsResponse
FolderMove	FormatBeanHelper	GetAttributeElementsResponse.Factory
FolderMoveSample	FormatColorLinesTabBean	GetAttributeParents
FolderNameHelper	FormatEditorTabManagerTransform	GetAttributeParents.Factory
FolderNew	FormatNumberTabBean	GetAttributeParentsResponse
FolderNewFilter	FormatPromptNumberTransform	GetAttributeParentsResponse.Factory
FolderNewPrompt	FormatTabAlignmentTransform	GetContents
FolderObjectBrowserTransform	FormatTabBean	GetContents.Factory
FolderObjectExplorerBean	FormatTabColorLinesTransform	GetContentsResponse
FolderObjectExplorerBeanImpl	FormatTabEffectsTransform	GetContentsResponse.Factory
FolderObjectExplorerEventHandler	FormatTabFontTransform	GetCubeBinaryResultsTask
FolderObjectExplorerTransform	FormatTabManagerBean	GetDBInstancesTask
FolderOMDBrowserTransform	FormatTabNumberTransform	GetDBMSTask
FolderOpen	Formatter	GetDescriptorsTask
FolderOpen	FormatToolbarBean	GetDMXMetricFunctionProperties
FolderPathRenderHelper	FormatToolbarTransform	GetDMXMetricFunctionProperties.Factory
FolderProjectBrowserAddOn	FormatUtils	GetDMXMetricFunctionPropertiesResponse
FolderProjectBrowserAddOnBase	FormEditorInput	GetDMXMetricFunctionPropertiesResponse.Factory
FolderProjectBrowserTransform	FormEventTag	GetDocumentPromptElements
FolderQuickSearchTransform	FormEventTagHelper	GetDocumentPromptElements.Factory
FolderRadioButtonTransform	FormGuiToolkit	GetDocumentPromptElementsResponse
FolderRename	FormulaBarBean	GetDocumentPromptElementsResponse.Factory
FolderReportProjectBrowserAddOn	FormulaBarTransform	GetDocumentResults
FolderRun	FrameBean	GetDocumentResults.Factory
FolderRun	FrameElement	GetDocumentResultsResponse
FolderRunAs	FrameGenericTransform	GetDocumentResultsResponse.Factory
FolderRunViewMedia	FreeFormSqlWriter	GetDocumentResultsResponse.Factory
FolderRunViewMedia	FunctionEditorBean	GetESRIConfigurationTask
FolderSaveAsBrowserTransform	FunctionEditorTransform	GetFlexSettingsTask
FolderSaveAsTransform	GenericBrowserSettings	GetGeoShapesTask
FolderSaveAsXHTMLTransform	GenericCookie	GetGoogleConfigurationTask
FolderSearchResultsTransform	GenericCSSTreeObject	GetGridFormatTask
	GenericEditorTransform	

GetHierarchyAttributes	GetProjects	GetVisualizationListsResponse.Factory
GetHierarchyAttributes.Factory	GetProjects.Factory	GGCalendar
GetHierarchyAttributesResponse	GetProjectSources	GGCalendarImpl
GetHierarchyAttributesResponse.Factory	GetProjectSources.Factory	GlobalPropertiesPage
GetHistory	GetProjectSourcesResponse	GlobClassMethodTracingFilter
GetHistory.Factory	GetProjectSourcesResponse.Factory	GrantAllPrivilegesToUser
GetHistoryResponse	GetProjectsResponse	GraphAxisInformation
GetHistoryResponse.Factory	GetProjectsResponse.Factory	GraphBean
GetIncrementalBrowseResults	GetProjectsTask	GraphColorGradientEditorBean
GetIncrementalBrowseResults.Factory	GetReportAnnotations	GraphColorGradientEditorBeanImpl
GetIncrementalBrowseResultsResponse	GetReportAnnotations.Factory	GraphColorPicker
GetIncrementalBrowseResultsResponse.Factory	GetReportAnnotationsResponse	GraphFormatEditorBean
GetMobileConfigurationBlockTask	GetReportAnnotationsResponse.Factory	GraphFormatHelper
GetMobileConfigurationIndexTask	GetReportGraphPropertiesTask	GraphFormatInfoHelper
GetMobileConfigurationTask	GetReportPromptElements	GraphFormatInfoHelper.Attribute
GetMobileConfigurationVersionTask	GetReportPromptElements.Factory	GraphFormatInfoHelper.Exception
GetMobileReportsTask	GetReportPromptElementsResponse	GraphFormatInfoHelper.Exceptions
GetMobileSubscriptionTask	GetReportPromptElementsResponse.Factory	GraphFormatInfoHelper.GraphAttribute
GetMWSFunctionalityProperties	GetReportResults	GraphFormatInfoHelper.GraphAttributes
GetMWSFunctionalityProperties.Factory	GetReportResults.Factory	GraphFormatInfoHelper.GraphFormatInfo
GetMWSFunctionalityPropertiesResponse	GetReportResultsResponse	GraphFormatInfoHelper.GraphMajorType
GetMWSFunctionalityPropertiesResponse.Factory	GetReportResultsResponse.Factory	GraphFormatInfoHelper.GraphMajorTypes
GetMWSIdentity	GetResources	GraphFormatInfoHelper.GraphMinorType
GetMWSIdentity.Factory	GetResources.Factory	GraphFormatInfoHelper.GraphMinorTypes
GetMWSIdentityResponse	GetResourcesResponse	GraphFormatInfoHelper.GraphObject
GetMWSIdentityResponse.Factory	GetResourcesResponse.Factory	GraphFormatInfoHelper.GraphObjects
GetMWSInfoProperties	GetRSDocumentAnnotations	GraphFormatInfoHelper.IncludeType
GetMWSInfoProperties.Factory	GetRSDocumentAnnotations.Factory	GraphFormatInfoHelper.IncludeTypes
GetMWSInfoPropertiesResponse	GetRSDocumentAnnotationsResponse	GraphFormatToolbarBean
GetMWSInfoPropertiesResponse.Factory	GetRSDocumentAnnotationsResponse.Factory	GraphImageTransform
GetMWSOfficeInstallInfo	GetRSDocumentPromptElements	GraphSeriesInformation
GetMWSOfficeInstallInfo.Factory	GetRSDocumentPromptElements.Factory	GraphSeriesTargetInfo
GetMWSOfficeInstallInfoResponse	GetRSDocumentPromptElementsResponse	GraphTitleInformation
GetMWSOfficeInstallInfoResponse.Factory	GetRSDocumentPromptElementsResponse.Factory	GraphTitleTabBean
GetMWSProperties2	GetRSDocumentResults	GraphTitleTabBeanImpl
GetMWSProperties2.Factory	GetRSDocumentResults.Factory	GridAutoStylesBean
GetMWSProperties2Response	GetRSDocumentResultsResponse	GridAutoStylesHelper
GetMWSProperties2Response.Factory	GetRSDocumentResultsResponse.Factory	GridAutoStylesTransform
GetNamedFolderID	GetRWGraphPropertiesTask	GridColorGradientEditorBean
GetNewMobileConfigurationTask	GetRWGridFormatTask	GridColorGradientEditorBeanImpl
GetObjectDetailsTask	GetRWTransactions	GridFormatBean
GetObjectProperties	GetSessionStateTask	GridFormatEditorBean
GetObjectProperties.Factory	GetTransactionReportDefinition	GridFormatHelper
GetObjectPropertiesResponse	GetTriggersTask	GridFormatToolbarBean
GetObjectPropertiesResponse.Factory	GetUserPreferences	GridLinkedDrillTask
GetPageByTreeTask	GetUserPreferences.Factory	GridWebFormatHelper
GetProjectFolder	GetUserPreferencesResponse	GridWidget
GetProjectFolder.Factory	GetUserPreferencesResponse.Factory	GridWidgetList
GetProjectFolderResponse	GetUserPrivilegeXmlTask	GroupbyPropertiesEditorBean
GetProjectFolderResponse.Factory	GetVisualizationLists	GroupbyPropertiesEditorTransform
	GetVisualizationLists.Factory	GroupPathBean
	GetVisualizationListsResponse	GroupPathTransform

GuiButtonTextClassSelectionAdapter	IfBeanValueTagHelper	InitTemplateTab
GuiButtonTextFolderSelectionAdapter	IfConnectionValueTag	InMemoryHandler
GuiButtonTextSelectionAdapter	IfConnectionValueTagHelper	InputTag
GuiComponent	IfDisplayMoreTag	InsertLayoutBlockTransform
GUID_Generator	IfDisplayMoreTagHelper	InsertLayoutFolderTransform
GUID_Generator	IfErrorValueTag	InsertMenuBlockTransform
GuiDataModifyListener	IfErrorValueTagHelper	Instance
GuiElement	IfFeatureTag	InstanceStatus
GuiElementFactory	IfFeatureTagHelper	IntArrayList
GuiElementsBlockTransform	IfFieldComparator	IntegerPool
GUIOperator	IFlushable	IntelligenceServerInstance
GuiSelectionChangeListener	IfPlainTextExportTag	IntelligenceServerInstances
GZIPHelper	IfPlainTextExportTagHelper	IntelligenceServerInstanceState
Handler	IFrameOutputTag	IntWrapper
HashList	IFrameUpdateFilterAddon	IPhoneGetReportResultsTask
HelpHelper	IFShowAdminPageTag	IPhoneGetRWResultsTask
HelpOnCBTabManagerTransform	IFShowAdminPageTagHelper	IphoneGraphResultsTask
HighlightedObject	ITag	IPhoneGridVisualizationPropertiesTransform
HighlightedObjects	IfTemplateValueTag	IPhoneImageMapPropertiesTransform
HistoryListPageComponentImpl	IfTemplateValueTagHelper	IPhoneMapVisualizationPropertiesTransform
HTMLAttributesJsGenerator	IfUseiframeTag	IPhoneMarkerMashupVisualizationPropertiesTransform
HTMLEncoder	IfUseiframeTagHelper	IPhoneMobileSubscribeTask
HTMLHelper	ImageMapVisualizationDataTransform	IPhoneMobileUnsubscribeTask
HttpBrowserSettings	ImageTag	IPhonePreLoadRWResultsTask
HTTPHelper	ImportDerivedElementsBean	IPhoneReportDataXMLTransform
HttpRequestKeys	ImportDerivedElementsEventHandler	IPhoneReportGridCellColHeaderImpl
HyperLinkObjectExplorerTransform	ImportDerivedElementsTransform	IPhoneReportGridCellMetricValueImpl
ICharsetModifiable	ImportFileTask	IPhoneReportGridCellRowHeaderImpl
ICharsetTranscoder	ImportWizardBean	IPhoneReportGridTransformImpl
IContentConsumer	ImportWizardFlashTransform	IPhoneReportInteractiveGridXMLTransform
IDSSXMLAdmin	InboxBean	IPhoneReportMapXMLTransform
IDSSXMLCluster	InboxClearTransform	IPhoneReportResultsTask
IDSSXMLClusterAdmin	InboxClearTransform	IPhoneReportXMLTransform
IDSSXMLClusterNode	InboxDelete	IPhoneRWResultsTask
IDSSXMLClusters	InboxDetails	IPhoneRWTransform
IDSSXMLConfigServer	InboxDetailsTransform	IPhoneTimeSeriesEditorTransform
IDSSXMLDocumentServer	InboxExport	IPhoneUtils
IDSSXMLElementServer	InboxExtrasTransform	IPortlet
IDSSXMLEncryptor	InboxListTransform	IPortletHelper
IDSSXMLFileLoader	InboxListXHTMLTransform	IResultsAsStream
IDSSXMLInbox	InboxListXHTMLTransform	ISelectionChangedListener
IDSSXMLMDXCubeServer	InboxPDF	IsSessionValid
IDSSXMLObjectServer	InboxRename	IsSessionValid.Factory
IDSSXMLPerfCounterServer	InboxToolbarTransform	IsSessionValidResponse
IDSSXMLReportServer	InboxToolbarTransform	IsSessionValidResponse.Factory
IDSSXMLScheduleServer	InboxUnread	IWebConfigModelListener
IDSSXMLServerAdmin	IncludeBlock	JavaClassInfo
IDSSXMLServerSession	IncludeDir	JavaConsole
IDSSXMLSubscriptionServer	IncludeTag	JavaConsoleHandler
IDSSXMLUserPrivilege	IncludeTagImpl	JavaConstructorInfo
IDSSXMLUserProjectPrivilege	IncrementalFetch	JavaMethodInfo
IExternalPortlet	IncrementalFetch.IncrementalFetchDetails	JavaParameterInfo
IExternalPortletHelper	IncrementalFetchImpl	JavaScriptBundles
IfBeanValueTag	IncrementalFetchSettings	JavaScriptBundles.JavaScriptBundle

JavaScriptBundles.	LayoutsImpl	LoginInfo
JavaScriptBundlesContentHandler	LayoutSource	LoginTask
JavaScriptBundles.JavaScriptFile	LayoutSourceDefn	LoginTransform
JavaScriptDependencyAnalyzer	LayoutSourceDefnImpl	LoginXHTMLTransform
JavaScriptFeaturesPage	LayoutSourceDefs	LogManager
JavaScriptHelper	LayoutSourceDefnsImpl	LogoffAction
JavaScriptInfo	LayoutSourceFile	LogoutBean
JavaScriptInfoList	LayoutSourceString	LogoutTask
JavaScriptPropertiesPage	LayoutsPage	LogoutTransform
JavaScriptResultsPage	LayoutTag	LogRecord
JavaScriptTag	LayoutTransform	LRUHashList
JavaScriptTagHelper	Level	LZW
JavaScriptWizard	LicensedUsers	MachineCPUInfo
Job	LicenseSource	Macro
JobDeletionFailure	LicensingException	MacroAlreadyExists
JobDetails	LicensingHelper	MacroDoesNotExist
JobManipulator	LicensingInfoTag	MacroHelper
JobResults	LicensingInfoTagHelper	Macros
JobSource	LicensingSupport	MainServlet
JsonEncoder	LimitedDict	MainView
JsonGenerator	LimitElementTransform	MainView.EditorHashMap
JsonPrettyPrinter	LimitExpressionBean	MalformedMethodException
JSONPTag	ListElementKeyConfig	ManipulateUserAnswer
JSONPTagHelper	ListObjectInfo	ManipulateUserAnswer.Factory
JsonRenderer	ListTag	ManipulateUserAnswerResponse
JspModule	LocalBeanFactory	ManipulateUserAnswerResponse.Factory
JuilElementTag	LocaleInfo	MapperDefn
JuilElementTagHelper	LocaleInfo	MapperDefnImpl
JUILLayoutsTagImpl	LocalesModule	MapperDefns
JUILLayoutsTagImpl	LocalizableException	MapperDefnsImpl
KeepSessionAliveTask	LocalizableMessage	MarkupOutput
KeyAlreadyExistsException	Log	MasterEdit_jsp
KeyDoesNotExistException	Log	MasterMSTRExternalPortlet
KeyedList	Log	MasterMSTRPortlet
KeyedObject	Log	MasterMSTRPortlet
LaunchDeveloperLibraryAction	Log	MasterMSTRPortlet
LaunchWebAPIReferenceAction	Log	MasterMSTRPortlet
LaunchWebBrowserCheatSheetAction	LogCategories	MasterPortletFolderSetFlagsAddOn
Layout	LogCategory	MasterPromptPortletAddon
LayoutContext	LogCounter	MDXReportLimitElementTransform
LayoutFileSelectionListener	LogDestination	Menu
LayoutFilesModule	LogDestinations	MenuBlockTransform
LayoutFilesTreeObject	LogFormatter	MenuElementEventPage
LayoutFileUtils	Logger	MenuElementItemsPage
LayoutImpl	Logger	MenuElementPropertiesPage
LayoutParser	LoggerConfigObserver	MenuElementResultsPage
LayoutParser.MessageListener	LoggerConfigurator	MenuElementWizard
LayoutParserAware	LoggerHelper	MenuImpl
LayoutParserDefn	LoggingManagementBean	MenuItemPage
LayoutParserDefnImpl	LoggingTag	MenuResultsPage
LayoutParserDefns	LoggingTagHelper	MenuSelectionPage
LayoutParserDefnsImpl	LoginBean	MenuWizard
LayoutParsingException	LoginContinueTransform	MergeDelta
Layouts	LoginForm	MergeDeltaManager

MessageInfo	MobilePollStatusTask	MSTRExternalPortlet
MessageImpl	MobilePreambleXMLTransform	MSTRExternalPortletHelper
MessageRouter	MobilePromptsXmlTask	MSTRExternalPortletHelper
Messages	MobilePushRegisterTask	MstrExternalProvider
MessagesManager	MobilePushUnregisterTask	MstrHelpAboutAction
MetaContentType	MobileRefreshSubscriptionTask	MstrHelpAboutDialog
MetaContentTypeTag	MobileRWPDFResultsTask	MstrInputDialog
MetaContentTypeTagHelper	MobileRWResultsTask	MSTRLayoutParser
MetadataPromptsSource	MobileSavePropertiesTask	MSTRLayoutParser.ParserContentHandler
MetadataSearchTask	MobileSaveSubscriptionTask	MSTRLayoutParser.ParserErrorHandler
MetricQualLevelEditorBean	MobileServlet	MSTRLiferayPortletHelper
MetricQualLevelEditorTransform	MobileSubscriptionTask	MSTRLog
MetricValue	MobileTaskUtils	MSTRLoginForm
MicroStrategyPropertiesPage	MobileUpdateSubscriptionTask	MstrMasterExternalPortlet
MicroStrategySettingsEditor	MobileValidateMetricValueTask	MSTRMessage
MiscellaneousPropertiesPage	ModifyChildrenTask	MstrMessageDialog
MissingKeyException	ModifyHistoryListMessages	MstrNamedObject
MobileAppLoginTask	ModifyHistoryListMessages.Factory	MstrPageTagHelper
MobileChunkedApplicationContext	ModifyHistoryListMessagesResponse	MSTRPortlet
MobileChunkedApplicationContext.	ModifyHistoryListMessagesResponse.Factory	MSTRPortlet
MobileElementlDMap	ModifyPrivilegesTask	MSTRPortlet
MobileChunkedApplicationContext.	ModifySecurityFilterTask	MSTRPortlet
MobileObjectRefIDMap	ModifySecurityRolesTask	MSTRPortletException
MobileChunkedApplicationContext.	ModifyTaskClassPage	MSTRPortletHelper
MobilePageIndexMap	ModifyTaskClassWizard	MSTRPortletRegistry
MobileChunkedApplicationContext.	ModifyUserTask	MSTRPortletRegistryEntry
MobileStyleManager	MojoAttributeTransform	MSTRPortletTypeB
MobileChunkedGraphDataTransform	MojoCustomGroupTransform	MstrSettingsTreeObject
MobileChunkedGraphResultsTask	MojoFolderTransform	MstrTreeViewer
MobileChunkedGridResultsTask	MojoMapTransform	MSTRUncheckedException
MobileChunkedRWResultsTask	MojoRWDTransform	MSTRWebAPIException
MobileChunkedRWTransform	MojoRWDTransform.ScriptTagExtractor	MSTRWebController
MobileChunkingHelper	MojoRWDTransform.TransformStringUtil	MSTRWebRequest
MobileConfigurationException	MojoVisualizationDataTransform	MSTRWSJMessageReceiverInOut
MobileConfigurationManager	MojoVisualizationSimplifiedDataTransform	MSTRWSJMSTRWSSoap
MobileController	MojoXtabTransformImpl	MSTRWSJMSTRWSSoap12
MobileControllerRequestState	MonitorFactory	MSTRWSJMSTRWSSoap12Stub
MobileDataXMLTransform	MonitorFilter	MSTRWSJServlet
MobileDeleteSubscriptionTask	MonitorFilterCondition	MSTRWSJSkeleton
MobileDuplicateMessageTask	MonitoringUseCases	MultipartRequest
MobileExecuteReportDSTask	MonitorManipulationException	MultipartRequest.File
MobileExecuteReportTask	MonitorManipulationFailure	MultipleObjectSelectorTransform
MobileExecuteRWDSTask	MonitorManipulator	MutableTreeNode
MobileExecuteRWTask	MonitorResults	MutableTreeNodeImpl
MobileGetDevicesTask	MonitorSort	MWSAttachmentInfo
MobileGetInboxMessageTime	MonitorSortCondition	MWSAttachmentInfo
MobileGetPushRegistrationStatusTask	MonitorSource	MWSAttachmentInfo.Factory
MobileGraphResultsTask	MSTRCheckedException	MWSAttachmentInfoBridge
MobileGraphXMLTask	MSTRCheckedException.	MWSAttributeElementBrowser
MobileGraphXMLTransform	ExceptionXMLHandler	MWSAttributeLockInfo
MobileGridResultsTask	MSTRCredentials	MWSAttributeLockInfo
MobileHTMLGridResultsTask	MstrCssLoader	MWSAttributeLockInfo.Factory
MobileHTMLGridTransform	MSTRDocumentException	MWSAttributeLockInfoBridge
MobileLoadSubscriptionTask	MSTRExternalPortlet	
MobileLoginTask		

MWSAttributeObjectInfo	MWSElementRestriction.Factory	MWSHistoryInfo
MWSAttributeObjectInfo	MWSElementRestrictionJBridge	MWSHistoryInfo
MWSAttributeObjectInfo.Factory	MWSEnumHelper	MWSHistoryInfo.Factory
MWSAttributeObjectInfoJBridge	MWSErrorCodes	MWSHistoryInfoJBridge
MWSAttributeShortcutInfo	MWSErrorInfo	MWSInfoPropertiesHandler
MWSAttributeShortcutInfo	MWSEception	MWSInfoProperty
MWSAttributeShortcutInfo.Factory	MWSExecuteInfo	MWSInfoProperty
MWSAttributeShortcutInfoJBridge	MWSExecuteInfo	MWSInfoProperty.Factory
MWSBrowseInfo	MWSExecuteInfo.Factory	MWSInfoPropertyJBridge
MWSBrowseInfo	MWSExecuteInfoJBridge	MWSInitializer
MWSBrowseInfo.Factory	MWSExecutionObjSpec	MWSMetricInfo
MWSBrowseInfoJBridge	MWSExecutionObjSpec	MWSMetricInfo
MWSBrowser	MWSExecutionObjSpec.Factory	MWSMetricInfo.Factory
MWSBrowserUtils	MWSExecutionObjSpecJBridge	MWSMetricInfoJBridge
MWSCompoundParameter	MWSExecutionSettings	MWSObjectAnnotationsHandler
MWSConfigurationProperties	MWSExecutionSettings	MWSObjectExecuteInfo
MWSConnectInfo	MWSExecutionSettings.Factory	MWSObjectExecuteInfo
MWSConnectInfo	MWSExecutionSettingsJBridge	MWSObjectExecuteInfo.Factory
MWSConnectInfo.Factory	MWSExecutionUtils	MWSObjectExecuteInfoJBridge
MWSConnectInfoJBridge	MWSExportSetting	MWSObjectInfo
MWSConnection	MWSExportSetting	MWSObjectInfo
MWSConnectProperty	MWSExportSetting.Factory	MWSObjectInfo.Factory
MWSConnectProperty	MWSExportSettingJBridge	MWSObjectInfoJBridge
MWSConnectProperty.Factory	MWSExportSettings	MWSObjectInfoProperty
MWSConnectProperty2	MWSExportSettings	MWSObjectInfoProperty
MWSConnectProperty2	MWSExportSettings.Factory	MWSObjectInfoProperty.Factory
MWSConnectProperty2.Factory	MWSExportSettingsJBridge	MWSObjectInfoPropertyJBridge
MWSConnectProperty2JBridge	MWSFindInfo	MWSObjectInfoPropertySet
MWSConnectPropertyJBridge	MWSFunctionalityPropertiesHandler	MWSObjectInfoPropertySet
MWSCurrentElement	MWSFunctionalityProperty	MWSObjectInfoPropertySet.Factory
MWSCurrentElement	MWSFunctionalityProperty	MWSObjectInfoPropertySetGroup
MWSCurrentElement.Factory	MWSFunctionalityProperty.Factory	MWSObjectInfoPropertySetGroup
MWSCurrentElementJBridge	MWSFunctionalityPropertyJBridge	MWSObjectInfoPropertySetGroup.Factory
MWSDiscoverColumn	MWSHierarchyAttributeFormInfo	MWSObjectInfoPropertySetGroupJBridge
MWSDiscoverRow	MWSHierarchyAttributeFormInfo	MWSObjectInfoPropertySetJBridge
MWSDocumentExecution	MWSHierarchyAttributeFormInfo.Factory	MWSObjectInfoPropertySetSpec
MWSDocumentObjectInfo	MWSHierarchyAttributeFormInfoJBridge	MWSObjectInfoPropertySetSpec
MWSDocumentObjectInfo	MWSHierarchyAttributInfo	MWSObjectInfoPropertySetSpec.Factory
MWSDocumentObjectInfo.Factory	MWSHierarchyAttributInfo	MWSObjectInfoPropertySetSpecBridge
MWSDocumentObjectInfoJBridge	MWSHierarchyAttributInfo.Factory	MWSObjectPropertyHandler
MWSDocumentShortcutInfo	MWSHierarchyAttributInfoJBridge	MWSOfficeInstallInfo
MWSDocumentShortcutInfo	MWSHierarchyBrower	MWSOfficeInstallInfo
MWSDocumentShortcutInfo.Factory	MWSHierarchyElementCollectionInfo	MWSOfficeInstallInfo.Factory
MWSDocumentShortcutInfoJBridge	MWSHierarchyElementCollectionInfo	MWSOfficeInstallInfoHandler
MWSElementBrowseInfo	MWSHierarchyElementCollectionInfo.Factory	MWSOfficeInstallInfoJBridge
MWSElementBrowseInfo	MWSHierarchyElementCollectionInfoJBridge	MWSPageByElements
MWSElementBrowseInfo.Factory	MWSHierarchyElementFilter	MWSPageByElements
MWSElementBrowseInfoJBridge	MWSHierarchyElementFilter	MWSPageByElements.Factory
MWSElementFormInstance	MWSHierarchyElementFilter.Factory	MWSPageByElementsJBridge
MWSElementFormInstance	MWSHierarchyElementFilterJBridge	MWSPageByInfo
MWSElementFormInstance.Factory	MWSHierarchyElementInfo	MWSPageByInfo
MWSElementFormInstanceJBridge	MWSHierarchyElementInfo	MWSPageByInfo.Factory
MWSElementRestriction	MWSHierarchyElementInfo.Factory	MWSPageByInfoJBridge
MWSElementRestriction	MWSHierarchyElementInfoJBridge	MWSProjectFolderID

MWSProjectFolderObj	MWSResourceRequestInfo	MWSUAPIBridge
MWSProjectFolderObj	MWSResourceRequestInfo.Factory	MWSUserPreference
MWSProjectFolderObj.Factory	MWSResourceRequestInfoJBridge	MWSUserPreference
MWSProjectFolderObjJBridge	MWSResourceReturnInfo	MWSUserPreference.Factory
MWSProjectInfo	MWSResourceReturnInfo	MWSUserPreferenceHandler
MWSProjectInfo	MWSResourceReturnInfo.Factory	MWSUserPreferenceJBridge
MWSProjectInfo.Factory	MWSResourceReturnInfoJBridge	MWSUserPrivilege
MWSProjectInfoJBridge	MWSRestriction	MWSUserPrivilege
MWSProjectSessionInfo	MWSRestriction	MWSUserPrivilege.Factory
MWSProjectSessionInfo	MWSRestriction.Factory	MWSUserPrivilegeJBridge
MWSProjectSessionInfo.Factory	MWSRestrictionJBridge	MWSUtils
MWSProjectSessionInfoJBridge	MWSResultsSizeInfo	MWSVisualization
MWSProjectSessionProperty	MWSResultsSizeInfo	MWSVisualization
MWSProjectSessionProperty	MWSResultsSizeInfo.Factory	MWSVisualization.Factory
MWSProjectSessionProperty.Factory	MWSResultsSizeInfoJBridge	MWSVisualizationJBridge
MWSProjectSessionPropertyJBridge	MWSResultsWindow	MWSVisualizationList
MWSProjectSetting	MWSResultsWindow	MWSVisualizationList
MWSProjectSetting	MWSResultsWindow.Factory	MWSVisualizationList.Factory
MWSProjectSetting.Factory	MWSResultsWindowJBridge	MWSVisualizationListHandler
MWSProjectSettingJBridge	MWSRSDocShortcutInfo	MWSVisualizationListJBridge
MWSProjectSourceConnectList	MWSRSDocShortcutInfo	MWSWebAdmin
MWSProjectSourceInfo	MWSRSDocShortcutInfo.Factory	MWSSDLVersion
MWSProjectSourceInfo	MWSRSDocShortcutInfoJBridge	MyCustomExternalSecurity
MWSProjectSourceInfo.Factory	MWSRSDocumentObjectInfo	MySubscriptionsPageComponentImpl
MWSProjectSourceInfoJBridge	MWSRSDocumentObjectInfo	NamedUserLicense
MWSProjectSources	MWSRSDocumentObjectInfo.Factory	NamespaceEncoder
MWSProjectWellKnownObj	MWSRSDocumentObjectInfoJBridge	NativeUtils
MWSProjectWellKnownObj	MWSRSExecution	NCSRemoveRecipient
MWSProjectWellKnownObj.Factory	MWSRSExecutionSettings	NETServantRuntimeContext
MWSProjectWellKnownObjJBridge	MWSRSExecutionSettings	NetServantTargetInfo
MWSPromptSuggestions	MWSRSExecutionSettings.Factory	NewAddonParameterWizardPage
MWSPromptSuggestions	MWSRSExecutionSettingsJBridge	NewAddonParamPropertiesPage
MWSPromptSuggestions.Factory	MWSRSExportSetting	NewAddonResultsPage
MWSPromptSuggestionsJBridge	MWSRSExportSetting	NewAddonWizard
MWSProperties	MWSRSExportSetting.Factory	NewAddonWizardPage
MWSProperty	MWSRSExportSettingJBridge	NewConfigElementDialog
MWSProperty	MWSRSLayoutInfo	NewCSSElementDialog
MWSProperty.Factory	MWSRSLayoutInfo.Factory	NewEnumCustomEventHandlerTypes
MWSPropertyJBridge	MWSRSLayoutInfoJBridge	NewEventHandlerFactory
MWSReportExecution	MWSServletListener	NewFolderLinkDialog
MWSReportHeader	MWSShortcutObjectInfo	NewMicroStrategyProjectBasicPage
MWSReportHeader	MWSShortcutObjectInfo	NewMicroStrategyProjectWizard
MWSReportHeader.Factory	MWSShortcutObjectInfo.Factory	NewMstrProjectCheatSheetAction
MWSReportHeaderJBridge	MWSShortcutObjectInfoJBridge	NewMstrProjectWizardAction
MWSReportObjectInfo	MWSSoapHeader	NewPageInfoBasicPropertiesPage
MWSReportObjectInfo	MWSSoapHeader.Factory	NewPageInfoMenusPage
MWSReportObjectInfo.Factory	MWSSoapHeader0	NewPageInfoResultsPage
MWSReportObjectInfoJBridge	MWSSoapHeader0.Factory	NewPageInfoSectionsPage
MWSReportShortcutInfo	MWSSyncCommunicationSetting	NewPageInfoShortcutsPage
MWSReportShortcutInfo	MWSSyncCommunicationSetting	NewPageInfoStartPage
MWSReportShortcutInfo.Factory	MWSSyncCommunicationSetting.Factory	NewPageInfoTemplateStatusPage
MWSReportShortcutInfoJBridge	MWSSyncCommunicationSettingJBridge	NewPageInfoToolbarsPage
MWSReportSpec	MWSUAPI	NewPageWizard
MWSResourceRequestInfo		

NewParameterWizardPage	ObjectPathTagHelper	PageConfigContainerServices
NewStyle	ObjectPathTransform	PageConfigsModule
NewStyleLayoutPage	ObjectPathTransform	PageEventAndScriptProperties
NewStyleMapperWizard	ObjectPathXHTMLTransform	PageFullScreenModeAddOn
NewStyleMapWizard	ObjectPromptBean	PageGuiModifyListener
NewStyleMapWizardPage	ObjectRegistry	PageHistoryItem
NewStyleParameterWizardPage	ObjectReportPathTransform	PageHistoryList
NewStyleParamPropertiesPage	ObjectReportXHTMLPathTransform	PageInfo
NewStyleResultsPage	ObjectSelectionBean	PageInfoList
NewStyleWizard	ObjectSelectorProjectBrowserTransform	PageInputValidator
NewStyleWizardPage	ObjectSelectorTransform	PageManager
NewTaskStartPage	ObjectSiblingsTransform	PageResourceManager
NewTaskWizard	ObjectsPromptEditorTransform	PageSelectionListener
NewTemplateDialog	OfficeTransform	PageSetupHTMLEditorBean
NewTypeCustomHandler	OIVMToolbarTransform	PageSetupHTMLEditorBeanImpl
NewWebBeanBasicPropertiesPage	OldLinksController	PageSetupHTMLEditorEventHandler
NewWebBeanEventHandlerPage	OldLinksRequest	PageSetupHTMLEditorTransform
NewWebBeanFeaturesPage	OldLinksServlet	PageStateTag
NewWebBeanInitPropertiesPage	OMDCreateFolderBean	PageStateTagHelper
NewWebBeanRequestPropertiesPage	OMDCreateFolderTransform	PageSubscriptionsEdit
NewWebBeanResultsPage	OnDemandContextMenuBuilder	PageTemplateProperties
NewWebBeanWizard	OnDemandContextMenuBuilderFactory	PageTreeObject
NewWindowPageComponentImpl	OnDemandContextMenusTransform	Panel_CloseInfoTag
NewWizardTemplatePage	OnExceptionTag	Panel_ContentTag
NewXMLTaskResultsPage	OpenFormEditorAction	Panel_OpenInfo
NextTag	OpenSimpleFormEditorAction	Panel_PanelTag
NoErrorFilter	OpenStyle	Panel_TitleTag
NormalizeXML	OpenTransformType	PanelTagHelper
NumberParser	OptimizedReportGraphTransform	PaperSizes
ObjBrowserFolderLinks	OptimizedReportTransform	PaperSizesList
ObjectArray	OptimizedTabBeanImpl	PaperSizesListProperties
ObjectBean	OptionsAddOn	PaperSizesTreeObject
ObjectBrowserBean	OptionsBean	Parameter
ObjectBrowserManagerBean	OptionsBeanImpl	ParameterBuilder
ObjectBrowserManagerBeanFeaturesImpl	OptionsHelper	ParameterBuilder
ObjectBrowserManagerTransform	OutlineExpansionState	ParameterBuilderConfig
ObjectBrowserTabBean	OutputFormatStyleMapper	ParameterInfo
ObjectBrowserTabOMDTransform	PageAddOnsProperties	ParameterInfo.Description
ObjectBrowserTabProjectBrowserTransform	PageBeanProperties	ParameterInfoList
ObjectBrowserTabReportAllObjectsTransform	PageByBean	Parameters
ObjectBrowserTabReportWorkingSetTransform	PageByDelete	ParametersCartDialog
ObjectBrowserTabRWWorkingSetTransform	PageByElement	ParameterValidationException
ObjectExplorerBean	PageByElementList	ParserParameters
ObjectExplorerBlockTransform	PageByInfo	PatchXML
ObjectInfoTask	PageByPivot	PathBean
ObjectInfoTransform	PageByProperties	PathBeanImpl
ObjectInfoWidget	PageByRemove	PathTransform
ObjectLock	PageByRename	PDFBean
ObjectLockSource	PageBySort	PDFReportPreferencePropertyListMapper
ObjectLockSourceImpl	PageByTransform	PDFSaveReportPropertiesAddOn
ObjectManipulationDialogBean	PageByUnit	PDFSetFlagsAddOn
ObjectManipulationDialogTransform	PageByUnitReportFaked	PDFTransform
ObjectPathHelper	PageComponent	PerformanceLoggingConfiguration
ObjectPathTag	PageComponentFeaturesImpl	PerformanceMonitorBean

PerformanceMonitorBeanManager	Preferences	ProjectBrowserEventHandler
PerformanceMonitorInstance	PreferencesBean	ProjectBrowserTransform
PerformanceMonitorManager	PreferencesBean.GroupInfo	ProjectCacheBase
PerformanceStringTag	PreferencesBeanImpl	ProjectInfo
PerformanceStringTagHelper	PreferencesConfig	ProjectInformation
PerformanceTimerTag	PreferencesDocumentTransform	ProjectsBean
PerformanceTimerTagHelper	PreferencesDrillTransform	ProjectsTask
Persistent	PreferencesExcelHeaderFooterTransform	ProjectStatusBean
PersistentAdapter	PreferencesException	ProjectStatusTransform
PIKController	PreferencesExportAdminHeaderFooterTransform	ProjectsTransform
PIKRequest	PreferencesExportTransform	PromptAnswerAddon
PIKServlet	PreferencesFolderTransform	PromptAnswerBean
PlotGraph	PreferencesGeneralTransform	PromptAnswerTransform
PlugInDirectoryDialog	PreferencesGenericTransform	PromptAppBean
PluginsHelper	PreferencesGraphTransform	PromptAppBean.StyleController
PollStatusTask	PreferencesGridProjectLevelTransform	PromptBean
PortalAdminForm	PreferencesGridTransform	PromptCartTransform
PortalFeatures	PreferencesInboxTransform	PromptCartXHTMLTransform
PortalHelper	PreferencesLogoutTransform	PromptCheckboxTransform
PortalKitController	PreferencesMgr	PromptConstantAppBean
PortalKitException	PreferencesNCTransform	PromptConstantTransform
PortalKitRequest	PreferencesOfficeTransform	PromptConstantWidgetTransform
PortalKitServlet	PreferencesPageComponentImpl	PromptDefinitionAppBean
PortalUser	PreferencesPDFReportHeaderFooterTransform	PromptDefinitionBean
PortalUsers	PreferencesPDFTransform	PromptDetailsBean
PortletBacking	PreferencesPrintHeaderFooterTransform	PromptDetailsBeanImpl
PortletLogHandler	PreferencesPrintHeaderFooterTransform.	PromptDetailsTransform
PortletLogHandler	ImageButtonInformation	PromptDimtyAppBean
PortletLogHandler	PreferencesPrintTransform	PromptDimtyWidgetTransform
PortletLogHandler	PreferencesProjectTransform	PromptEditorJavaScriptBoneTransform
PortletLogHandler	PreferencesPromptsLayoutTransform	PromptEditorTabManagerTransform
PortletParmsAccess	PreferencesPromptsTransform	PromptEditorTabTransform
PortletSendToOptions	PreferencesSecurityTransform	PromptElementsAppBean
PortletSessionHelper	PreferencesToolbarTransform	PromptElementsWidgetTransform
PortletSessionManager	PreferencesUserGroupsTransform	PromptExpressionAppBean
PortletSettingBean	PreferenceTreeObject	PromptExpressionCartTransform
PortletSettingBean	PreferenceValidationForDefaultLoginPage	PromptExpressionCartWithPickerTransform
PortletSettingBean	PreferenceValidationForMargins	PromptExpressionCartXHTMLTransform
PortletSettingBean	PreferenceValidationForPrintRowsColsPerPage	PromptExpressionListTransform
PortletSettingBean	PreferenceValidationForStrings	PromptExpressionNonCartTransform
PortletSettingBeanTypeB	PreferenceValueTag	PromptExpressionTextboxTransform
PostedFile	PreferenceValueTagHelper	PromptExpressionTransform
PreferenceAllowableValue	PreorderEnumeration	PromptExpressionWidgetTransform
PreferenceBrowserSettings	PrettyPrintTransform	PromptHierarchicalCartTransform
PreferenceConfig	PrintBean	PromptHierarchicalCartXHTMLTransform
PreferenceDefinition	PrintBeanHelper	PromptHierarchicalTreeTransform
PreferenceDefinitionBase	PrintReportPreferencePropertyListMapper	PromptHierarchicalTreeXHTMLTransform
PreferenceDefinitionImpl	PrintSaveReportPropertiesAddOn	PromptHierarchyAppBean
PreferenceDefinitionLocale	PrintTransform	PromptHierarchyWidgetTransform
PreferenceDefinitionMinMax	ProcessStreamGobbler	PromptListboxTransform
PreferenceLevel	ProductLicenseInfo	PromptObject
PreferenceLevels	ProgrammingBasedMethodsTaskPage	PromptObjectAppBean
PreferenceProperties	ProjectBrowser	PromptObjectBrowsingTransform
PreferenceReaderWriter	ProjectBrowserBeanImpl	PromptObjectBrowsingXHTMLTransform

PromptObjectWidgetTransform	RegisteredTaskWizardPropertiesPage	ReportCellMetricFilterEdit
PromptPathTransform	RenameElementDialog	ReportCellMove
PromptRadioTransform	RenameHistoryListMessage	ReportCellMoveDown
PromptsBean	RenameHistoryListMessage.Factory	ReportCellMoveLeft
PromptsClassicTransform	RenameHistoryListMessageResponse	ReportCellMoveRight
PromptsClassicTransformForPortlet	RenameHistoryListMessageResponse.Factory	ReportCellMoveToColumns
PromptsContainerBean	Renderer	ReportCellMoveToPageBy
PromptsContainerFrameBean	RenderTag	ReportCellMoveToRows
PromptsContainerGUILayoutMapper	ReplaceTag	ReportCellMoveUp
PromptsContainerTransform	ReplaceTagImpl	ReportCellPercentForEachMetric
PromptsContainerTransformForPortlet	Report2RWSettings	ReportCellPercentForEachMetric.
PromptsCreateReportContainerTransform	ReportAjaxMojoVisualizationTransform	AbstractInsertMetricContextMenuBuilder
PromptsCreateReportTransform	ReportAjaxVisualizationTransform	ReportCellPercentForEachMetric.
PromptsIFrameTransform	ReportBean	InsertMetricContextMenuBuilder
PromptsSource	ReportBinaryResultsTask	ReportCellPercentForEachMetric.
PromptsSubscriptionTransform	ReportCellActionControlEditor	InvalidInsertMetricContextMenuBuilder
PromptsSummaryTransform	ReportCellAdvancedThresholds	ReportCellPercentGrandTotalMetric
PromptXSLTransform	ReportCellAlert	ReportCellPercentOverColumnsMetric
PropertiesCache	ReportCellAlertEmail	ReportCellPercentOverPagesMetric
PropertiesCartDialog	ReportCellAlertMobile	ReportCellPercentOverRowsMetric
PropertiesComparator	ReportCellAttributeFormEditor	ReportCellRankMetric
PropertiesDescriptorModifyDialog	ReportCellAttributeForms	ReportCellRankMetric.
PropertiesModule	ReportCellAttributeForms.	AbstractInsertMetricContextMenuBuilder
PropertiesParser	AbstractAttributeFormsContextMenuBuilder	ReportCellRankMetric.
PropertiesTab	ReportCellAttributeForms.	InsertMetricContextMenuBuilder
Property	AttributeFormsContextMenuBuilder	ReportCellRankMetric.
Property	ReportCellAttributeForms.	InvalidInsertMetricContextMenuBuilder
PropertyFilesUtils	InvalidAttributeFormsContextMenuBuilder	ReportCellRemoveFromGrid
PropertyFileTreeObject	ReportCellClearDerivedElements	ReportCellRemoveFromReport
PropertySet	ReportCellCustomSort	ReportCellRename
PropertySetHelper	ReportCellDefaultMenu	ReportCellSelectorControlEditor
PropertySets	ReportCellDEQuickGroup	ReportCellSort
PropertyValue	ReportCellDEQuickGroupOperator	ReportCellSortAscending
PurgeCachesTag	ReportCellDerivedElements	ReportCellSortColAscending
PurgeCachesTagHelper	ReportCellDrill	ReportCellSortColDescending
QuickLinks	ReportCellDrill.AbstractDrillContextMenuBuilder	ReportCellSortDescending
QuickLinksImpl	ReportCellDrill.DrillContextMenuBuilder	ReportCellThresholds
QuickSearchTag	ReportCellDrill.HeaderDrillContextMenuBuilder	ReportCellTransformationMetric
QuickSearchTagHelper	ReportCellDrill.InvalidDrillContextMenuBuilder	ReportCellTransformationMetric.
ReadUserPropertySets	ReportCellDrill.RowStyleDrillContextMenuBuilder	AbstractInsertMetricContextMenuBuilder
ReadWriteLock	ReportCellDrill.TitleDrillContextMenuBuilder	ReportCellTransformationMetric.
ReadWriteLockFactory	ReportCellFilterOn	InsertMetricContextMenuBuilder
ReadWriteLockLogWrapper	ReportCellFilterOnSelections	ReportCellTransformationMetric.
Redirectable	ReportCellFormat	InvalidInsertMetricContextMenuBuilder
RedirectLoginException	ReportCellHyperLink	ReportCellVisualThresholds
ReentrantWriterPreferenceReadWriteLock	ReportCellHyperLinkNavigation	ReportDataServiceTask
ReflectionHelper	ReportCellHyperLinkNavigation.	ReportDataTransform
Reflector	AbstractHyperLinkNavigationContextMenuBuilder	ReportDesignModeTransformImpl
RegionalOptions	ReportCellHyperLinkNavigation.	ReportDetailsBean
RegionalOptionsHelper	HyperLinkNavigationContextMenuBuilder	ReportDetailsBeanImpl
RegionalOptionsVisitor	ReportCellHyperLinkNavigation.	ReportDetailsPanelTransform
RegionalOptionsVisitorHandler	InvalidHyperLinkNavigationContextMenuBuilder	ReportDetailsSetFlagsAddOn
RegionalValues	ReportCellInsertMetric	ReportDetailsTransform
RegisteredTaskWizard	ReportCellInsertPercentToTotalMetric	ReportDetailsXHTMLTransform
	ReportCellMetricFilter	ReportDetailsXHTMLTransform
		ReportDisplayCellsFactory

ReportErrorVisualizationTransform	ReportGridExcelCellColHeaderImpl	RequestTemplateTab
ReportExecuteTaskWithWriteback	ReportGridExcelCellColTitleImpl	ResizeEditorBean
ReportExecutionSample	ReportGridExcelCellMetricValueImpl	ResizeEditorTransform
ReportExecutionTask	ReportGridExcelCellRowHeaderImpl	Resource
ReportExportExcelFormattingTransform	ReportGridExcelCellRowTitleImpl	ResourceBundleCache
ReportExportExcelFormatTransform	ReportGridExcelTransform	ResourceCache
ReportExportHTMLFormatTransform	ReportGridGraphTransformImpl	ResourceCacheHint
ReportExportHTMLTransform	ReportGridHTMLTransform	ResourceCacheManager
ReportExportIServerTransform	ReportGridPrintTransform	ResourceFeed
ReportExportPlainTextTransform	ReportGridToolbarBlockLoaderTransform	ResourceFeedConfig
ReportFlashTransform	ReportGridTransformImpl	ResourceFeedController
ReportFrameBean	ReportGridWritebackTransform	ResourceFeedNameTag
ReportFrameBeanImpl	ReportHomeExportMenuBlockTransform	ResourceFeedNameTagHelper
ReportFrameClassicTransform	ReportHTMLFormatTransform	ResourceFeedRequestState
ReportFrameElement	ReportIncrementalFetchTransform	ResourceMgrTag
ReportFrameGenericTransform	ReportLastUpdateTransform	ResourceMgrTagHelper
ReportGoogleGraphVisualizationTransform	ReportMoveSample	ResponseContextTag
ReportGraphDesignTransform	ReportObjectsTabBean	ResponseContextTagHelper
ReportGraphDrill	ReportOptionsTransform	RestrictedFolderObjectExplorerTransform
ReportGraphDrill.	ReportOutlineModeExcelTransform	ResultSetBean
GraphAreaContextMenuBuilder	ReportOutlineModeHTMLTransform	ResultSetBeanManipulationTask
ReportGraphDrillTask	ReportOutlineModeTransformImpl	ResultSetFrameBean
ReportGraphFill	ReportPageByCellDrill	ResultSetFrameTransform
ReportGraphFont	ReportPageByCellDrill.TitleDrillContextMenuBuilder	ResultSetPageComponentImpl
ReportGraphFormat	ReportPageByInfoTransform	ResultSetStatusTransform
ReportGraphImageTransform	ReportPageByTransform	RetrieveChildrenTask
ReportGraphLine	ReportPageComponentImpl	ReturnToPathTag
ReportGraphTransformImpl	ReportPageEventHandler	ReturnToPathTagHelper
ReportGridCell	ReportPageInfo	ReturnToTag
ReportGridCellColHeader	ReportPathTransform	ReturnToTagHelper
ReportGridCellColHeaderImpl	ReportPortalComponent	ReverseGeoCode
ReportGridCellColHeaderImpl.	ReportPreferencePropertyListMapper	RibbonBean
ColHeaderPivotButtonInfo	ReportQuickLinksTransform	RibbonBeanImpl
ReportGridCellColTitle	ReportSavePropertiesTransform	RibbonEventHandlerImpl
ReportGridCellColTitleImpl	ReportSaveTask	RibbonInfo
ReportGridCellHeader	ReportSearchTask	RibbonInputValidator
ReportGridCellImpl	ReportSetFlagsAddOn	RibbonList
ReportGridCellMetricValue	ReportSetResultWindowAddOn	RibbonListInputValidator
ReportGridCellMetricValueImpl	ReportTimelineVisualizationTransform	RibbonListProperties
ReportGridCellRowHeader	ReportTimelineXMLTransform	RibbonListSet
ReportGridCellRowHeaderImpl	ReportToolbarTransform	RibbonListTreeObject
ReportGridCellRowTitle	ReportTransformHelper	RibbonProperties
ReportGridCellRowTitleImpl	ReportViewerTransform	RibbonToolbarTransform
ReportGridCellsFactory	ReportWorkingSetBrowserTabBean	RibbonTreeObject
ReportGridCellTitle	ReportWorkingSetBrowserTabBeanFeaturesImpl	RootBlockTagImpl
ReportGridDisplayCell	ReportWSDelete	RowHeader
ReportGridDisplayCell.Context	ReportXMLTransform	RowTag
ReportGridDisplayCellColHeader	RepositorySettings	RowTitle
ReportGridDisplayCellColTitle	RepositorySource	RunTimeDiagnosticsTag
ReportGridDisplayCellImpl	RequestKeys	RunTimeDiagnosticsTagHelper
ReportGridDisplayCellMetricValue	RequestPersistable	RWActionControlEditorBean
ReportGridDisplayCellRowHeader	RequestState	RWActionControlEditorTransform
ReportGridDisplayCellRowTitle	RequestStringTag	RWActionSelector
ReportGridDrillTask	RequestStringTagHelper	RWBean

RWBeanVisitor	RWGridTransform	RWSelectorControlObject
RWBeanVisitor2	RWGroupBy	RWSetFlagsAddOn
RWClipboard	RWGroupByElement	RWSetOIVMFlagsAddOn
RWConditionSelector	RWGroupByElements	RWSettings
RWContainerFeatures	RWGroupBys	RWShapeDef
RWControl	RWGroupByUnit	RWSource
RWControlDef	RWHTMLDef	RWSubsectionDef
RWControlGroupBy	RWHyperlink	RWSubtotal
RWControlGroupBys	RWHyperlinkContainer	RWSubtotals
RWData	RWHyperLinkObjectLinks	RWTextDef
RWDataServiceTask	RWHyperlinks	RWTextObject
RWDataSet	RWHyperlinksImpl	RWThreshold
RWDataSetDetails	RWImageDef	RWThresholdText
RWDataSets	RWIncrementalFetchJsonGenerator	RWTransactable
RWDataSettings	RWIncrementalFetchSettings	RWTransformHelper
RWDColorGradientEditorBean	RWIncrementalFetchTransform	RWUnit
RWDColorGradientEditorBeanImpl	RWInstance	RWUnitDef
RWDColorGradientEditorTransform	RWIterateSettings	RWUnitVisualizationSettingsImpl
RWDefinition	RWLastUpdateTransform	RWUserMetric
RWDesignModeAddOn	RWLayoutSectionDef	RWUserMetrics
RWDetails	RWLayoutTabStripTransform	RWViewBean
RWDetailsFormatter	RWLayoutTransform	RWViewExpressionBean
RWDetailsSetFlagsAddOn	RWLineDef	RWViewLimitHelper
RWDetailsTransform	RWLinkedDrillTask	RWVisitor
RWDetailsTransform	RWManipulation	RWVisitor2
RWDFilterAddOn	RWManipulationTask	RWWorkingSetBrowserTabBean
RWDocPropertiesEditorTransform	RWManipulationWithoutResultTask	RWXMLTransform
RWDocumentDetails	RWMessage	SampleCustomBean
RWDrl	RWMetricSelector	SampleCustomEventHandler
RWDrl2ReportSettings	RWObject	SampleCustomTransform
RWDrlTask	RWOIVMFeatures	SampleExternalSecurity
RWDWizardAddOn	RWOIVMPageComponent	SampleTransform
RWElementSelector	RWOnDemandDrillTask	SaveAndPublishCubeTask
RWExcelExportSettings	RWOutlineViewTransform	SaveAsBean
RWExecutionTask	RWPageComponentFeaturesImpl	SaveAsFolderFlagsAddOn
RWExportAddOn	RWPageComponentImpl	SaveAsReportFlagsAddOn
RWExportSettings	RWPanelsStackControl	SaveAsReportTransform
RWExportTransform	RWPanelsStackDef	SaveAsRWTransform
RWFieldDef	RWPathTransform	SaveAsTransform
RWFieldGroupDef	RWPDFTransform	SaveCustomGroupTask
RWFlashDashboardTransform	RWProjectBrowserTabBean	SaveDBInstanceTask
RWFlashTransform	RWPromptsPageComponent	SaveObjectACLTTask
RWFormatProperty	RWPropertiesEditorTransform	SaveObjectAsDialogBean
RWFrameBean	RWReportGraphDesignDropZoneTransformImpl	SaveObjectAsDialogTransform
RWFrameGenericTransform	RWReportGraphDropZoneTransformImpl	SaveRWTasks
RWGraphDropZonesEditorBean	RWSaveAsPageComponentImpl	SaveSubscriptionTask
RWGraphDropZonesEditorBeanImpl	RWSavePropertiesTransform	SAXHelper
RWGraphDropZonesEditorEventHandlerImpl	RWSaveTask	SAXParsingException
RWGraphDropZonesEditorTransform	RWSecondaryDataProviderHelper	SAXSupport
RWGridGraphDef	RWSecondaryDataProviderHelperImpl	SAXSupportException
RWGridGraphObject	RWSection	SaxToDomHandler
RWGridGraphSettings	RWSectionDef	SAXXMLStringDumper
RWGridIFRenderer	RWSectionsVisitor	ScheduleBean
RWGridIfSettings	RWSelectorControl	ScheduleEditTransform

ScheduleOverwriteTransform	SerializeSampleBean	ShoppingCartComposite
SchedulesBulletTransform	ServantGenerator	ShoppingCartDialog
SchedulesFilteredListTransform	ServantGenerator.JavaGenerator	Shortcut
SchedulesFilteredListXHTMLTransform	ServantRuntimeContext	ShortcutArgument
SchedulesIconTransform	ServantRuntimeSupport	ShortcutBean
SchedulesIconXHTMLTransform	ServantTargetInfo	ShortcutButton
SchedulesListTransform	ServantTargetInfoFactory	ShortcutButtonLabel
SchedulesListXHTMLTransform	ServerAdminController	ShortcutCaption
ScheduleWidget	ServerAdminFeatures	ShortcutColorPickerPane
ScriptletTag	ServerAdminSessionManager	ShortcutCombo
ScriptletTagHelper	ServerCacheBase	ShortcutDefault
ScriptPageStateTag	ServerClusteringStartupMembershipBuilder	ShortcutElement
ScriptPageStateTagHelper	ServerConfigException	ShortcutElementTag
ScriptTagImpl	ServerConnectionTask	ShortcutEvent
ScriptVariableJSONEncoder	ServerControlException	ShortcutEventDialog
Scrollable	ServerControlSource	ShortCutEventEditDialog
ScrolledPropertiesBlock	ServerEncoding	ShortcutGridInfo
ScrolledPropertiesPage	ServerInfo	ShortcutGridLevels
Scroller	ServerInstance	ShortcutHelper
ScrollerImpl	ServerInstances	ShortcutInput
SearchBean	ServerLicense	ShortcutInputValidator
SearchConfigObjectsAction	ServerMachine	ShortcutLabel
SearchConfigObjectsDialog	ServerPropertiesTransform	ShortcutList
SearchInText	ServersTask	ShortcutListElement
SearchPageComponentImpl	ServerStatisticsBean	ShortcutListPicker
SearchResultRange	ServerStatisticsBeanImpl	ShortcutListPickerElement
SearchSimpleTransform	ServerVersionInfo	ShortcutListProperties
SearchSimpleTransform	ServiceConfiguration	ShortcutListSet
SearchSupportedTypesAddon	ServletContainerServices	ShortcutListSet.EventActionMapping
SearchTab	ServletEvent	ShortcutListSetInputValidator
SecondaryDataSourcesEditorBean	ServletWebComponent	ShortcutListSetProperties
SecondaryDataSourcesEditorBeanImpl	SessionCacheBase	ShortcutListSetTreeObject
SecondaryDataSourcesEditorTransform	SessionHelper	ShortcutListStyles
SectionInfo	SessionHelperException	ShortcutListTreeObject
SectionInfoList	SessionManagementSample	ShortcutOptionsTag
SecurityFilterBean	SessionManager	ShortcutPickerOption
SecurityFilterEventHandler	SetCubeMappingTask	ShortcutProperties
SecurityFilterExpressionTransform	SetDocSelectorElementsTask	ShortcutPropertyTag
SecurityFilterFrameBean	SetDocumentZoomTask	ShortcutPulldown
SecurityFilterFrameTransform	SetPreferenceTask	ShortcutSeparatorTag
SecurityFilterTransform	SetReportAnnotations	ShortcutTagHelper
SecurityRoleBean	SetReportAnnotations.Factory	ShortcutTextcombo
SecurityRoleEditorTransform	SetReportAnnotationsResponse	ShortcutTreeObject
SecurityRoleEventHandler	SetReportAnnotationsResponse.Factory	ShowBundleDescriptorTag
SecurityRolePathTransform	SetReportWritebackSQL	ShowBundleDescriptorTagHelper
SecurityRoleUserEntitiesSelectorBean	SetRSDocumentAnnotations	ShowDebugInfoTag
SecurityRoleUserEntitiesSelectorBeanImpl	SetRSDocumentAnnotations.Factory	ShowDebugInfoTagHelper
SecurityRoleUserEntitiesSelectorTransform	SetRSDocumentAnnotationsResponse	ShowJsonBlocksTag
SecurityTransform	SetRSDocumentAnnotationsResponse.Factory	ShowJsonBlocksTagHelper
SelectControlTargets	Factory	SimpleBrowserSettingsEditor
SelectionChangedListener	SetRWUnitPropertiesTask	SimpleCartElement
SelectTag	SettingBean	SimpleCSSSearchEditor
SendNowWidget	SetVisualizationPropertiesTask	SimpleDebugger
SerializerParameters	SharedPortletSpaceAccess	SimpleEditorBean

SimpleEditorTransform	SlotTagImpl	StyleMapper
SimpleErrorHandler	SmartProperties	StyleMapperResult
SimpleErrorsEditor	SortDefinition	StyleMapperResultImpl
SimpleEventEditor	SortEditorBean	StyleMapperWizardPage
SimpleExecuteReport	SortEditorTransform	StyleMapProperties
SimpleExecuteReport.Factory	SortGeneralTransform	StyleMaps
SimpleExecuteReportResponse	SortTabBeanTransform	StyleMapsImpl
SimpleExecuteReportResponse.Factory	SoucreConfig	StyleMapTreeObject
SimpleExecuteRSDocument	SourceLookup	StyleRequestContext
SimpleExecuteRSDocument.Factory	SourcesConfig	Styles
SimpleExecuteRSDocumentResponse	SparklinesTransform	Styles
SimpleExecuteRSDocumentResponse.Factory	SpnegoFilter	Styles
SimpleExportFormatsEditor	SpnegoWSFilter	StylesImpl
SimpleFileHandler	SQLWritebackEditorBean	StylesSelectionCartDialog
SimpleFolderLinksEditor	SQLWritebackEditorTransform	StyleTreeObject
SimpleFolderMappingEditor	StandardOutWriter	SubscriptionBean
SimpleFormatter	StaticSession	SubscriptionConfirmSendNowTransform
SimpleGetReportResults	StaticSessionList	SubscriptionConfirmTransform
SimpleGetReportResults.Factory	StatsHandler	SubscriptionContactsBrowserBean
SimpleGetReportResultsResponse	StatsHandlerSupport	SubscriptionContactsBrowserTransform
SimpleGetReportResultsResponse.Factory	StatsHandlerSupport.Data	SubscriptionEditFileTransform
SimpleGetRSDocumentResults	StatsLog	SubscriptionEditInboxTransform
SimpleGetRSDocumentResults.Factory	StatusBar	SubscriptionEditMobileTransform
SimpleGetRSDocumentResultsResponse	StatusCodeTag	SubscriptionEditorBean
SimpleGetRSDocumentResultsResponse.Factory	StatusCodeTagHelper	SubscriptionEditPrintTransform
SimpleHint	StreamHandler	SubscriptionEditTransform
SimpleList	StreamingContext	SubscriptionEditTransform.
SimpleListObj	StringKey	NCSRecipientContext
SimpleModel	StringRequestKeys	SubscriptionFileDialog
SimplePageComponentImpl	StringStore	SubscriptionFileDialogImpl
SimplePageEditor	StringTag	SubscriptionFolderBean
SimplePaperSizesEditor	StringTagHelper	SubscriptionFrameBean
SimplePreferenceEditor	StringTokenizerWithEscape	SubscriptionFrameTransform
SimpleRibbonEditor	StringUtils	SubscriptionMobileWidgetImpl
SimpleRibbonListEditor	StringUtils	SubscriptionParserTags
SimpleSessionESM	StringWrapper	SubscriptionPrintWidget
SimpleShortcutEditor	Style	SubscriptionPrintWidgetImpl
SimpleShortcutListEditor	StyleAutoCompletionProcessor	SubscriptionsBulletTransform
SimpleShortcutListSetEditor	StyleCatalog	SubscriptionsEditorHistoryListTransform
SimpleStyleEditor	StyleCatalogFactory	SubscriptionsEditorMobileTransform
SimpleStyleMapEditor	StyleCatalogImpl	SubscriptionsEditorNCFileTransform
SimpleTaskEditor	StyleCatalogModule	SubscriptionsEditorNCPrintTransform
SimpleTaskFactoryEditor	StyleFilter	SubscriptionsEditorNCSendNowTransform
SimpleThresholdsEditorTransform	StyleFormEditorInput	SubscriptionsEditorNCTransform
SimpleTransformDefinitionsEditor	StyleImpl	SubscriptionsEditorTransform
SimpleVisualizationEditor	StyleInfo	SubscriptionSendNowTransform
SimpleWidgetsEditor	StyleInputValidator	SubscriptionsFilteredListTransform
SingleObjectSelectorTransform	StyleMap	SubscriptionsFilteredListXHTMLTransform
SiteDesigner	StyleMapClause	SubscriptionsIconTransform
SJISModifier	StyleMapCondition	SubscriptionsIconXHTMLTransform
SlavePromptPortletAddon	StyleMapConditionImpl	SubscriptionsListTransform
SlotTag	StyleMapDefault	SubscriptionsListXHTMLTransform
	StyleMapDefaultImpl	SubscriptionWidget
	StyleMapImpl	SubsetUnitLimitBean

SubsetUnitLimitElementBean	TaskEventsArgumentsParametersPage	TemplateShortcutsSelectionPage
SubsetViewLimitElementDialogTransform	TaskException	TemplateShortcutsWizard
SubsetViewLimitElementTransform	TaskFactory	TemplateTab
SubtotalDefinition	TaskFactory.TaskCategory	TestBodyContentTag
SubtotalsDHTMLTransform	TaskFactoryInfo	TestFileHandler
SubtotalsEditorBean	TaskFactoryList	TestFormatter
SubtotalsEditorTabManagerTransform	TaskFactorySettingsListProperties	TestService
SubtotalsEditorTransform	TaskFactorySettingsTreeObject	TestService.Factory
Sync	TaskInfo	TestServiceResponse
SyncLogWrapper	TaskInputValidator	TestServiceResponse.Factory
SysDefaultPrefType	TaskInternalException	TestServiceServlet
SysDimensionCache	TaskInvoker	TextBeanEventHandler
SysEventHandler	TaskInvoker.Context	TextDisplayBean
SysPrefType	TaskInvoker.TaskInfo	TextDisplayTransform
SystemPickerList	TaskMetadata	TextHyperLinkNavigation
TabBean	TaskMetadataImpl	ThenTag
TabBeanTransform	TaskNotFoundException	ThreadLocalDebugHandler
TabControlHelper	TaskNotRequestedException	ThreadLocalSession
TabDocumentAdvancedTransform	TaskOutput	ThreadRegistry
TabDocumentExportTransform	TaskParameterMetadata	ThreadSessionMap
TabDocumentOtherTransform	TaskParameterMetadataImpl	ThresholdEditorBean
TabDocumentTableOfContentsTransform	TaskProcessor	ThresholdEditorBeanFeaturesImpl
TabDocumentWatermarkTransform	TaskProcessor.TaskInvoker	ThresholdExpressionBean
TabGraphAdvancedTransform	TaskProcessorAdminContext	ThresholdExpressionBeanContext
TabGraphAxesTransform	TaskProcessorController	ThresholdExpressionBeanImpl
TabGraphGeneralTransform	TaskProcessorController.TaskEnvelope	ThresholdExpressionTransform
TabGraphNumberTransform	TaskProcessorNameTag	ThresholdsEditorTransform
TabGraphOptionsTransform	TaskProcessorNameTagHelper	ThresholdsFormatEditorTransform
TabGraphTitlesTransform	TaskProcessorRequestState	TimeSeriesVisualizationDataTransform
TabLayoutTransform	TaskProcessorResponseContext	Toolbar
TableTag	TaskProcessorServlet	ToolbarBean
TabManagerBean	TaskRequestContext	ToolbarElementEventPage
TabManagerTransform	TaskRequestMalformedException	ToolbarElementPropertiesPage
TabPageMarginTransform	TaskSettingsTreeObject	ToolbarElementResultsPage
TabPageOutputTransform	TaskShoppingCartComposite	ToolbarElementSelectionPage
TabPageSetupTransform	TasksList	ToolbarElementWizard
TabPropertiesAdvancedTransform	TaskStatusCodes	ToolbarInfoList
TabPropertiesFlashTransform	TaskStyleParametersPage	ToolbarItemsPage
TabPropertiesGeneralTransform	TaskUnauthorizedException	ToolbarResultsPage
TabPropertiesGridTransform	TaskViewerController	ToolbarSelectionPage
TabPropertiesLayoutTransform	TaskViewerController.OptionInfo	ToolbarSetBean
TabPropertiesLineTransform	TaskViewerServlet	ToolbarSetTransform
TabPropertiesOtherGridTransform	TemplateElement	ToolbarTransform
TabPropertiesPictureTransform	TemplateElementsBean	ToolbarWizard
TabPropertiesSelectorTransform	TemplateFilterExecBean	TransactionAttribute
TabSectionsTransform	TemplateFilterExecTransform	TransactionAttributeForm
Tag	TemplateFilterExecXHTMLTransform	TransactionElement
TagImpl	TemplateInfo	TransactionMetric
TagsFactory	TemplateInfoList	TranscodingStreamingContext
Task	TemplateNewShortcutPage	Transform
TaskAdminController	TemplateParser	Transformable
TaskAdminServlet	TemplateShortcutsBasicPage	TransformContext
TaskConfigurationException	TemplateShortcutsRearrangePage	TransformCreationException
TaskEventDialog	TemplateShortcutsResultsPage	

TransformDefinitionPropertiesPage	UploadImageTask	ViewFilterElementTransform
TransformDefinitionsProperties	UrlEventTag	ViewFolderAction
TransformDefinitionTreeObject	UrlEventTagHelper	ViewFolderReportAction
TransformDefinitionWizard	URLParameterMapper	ViewFrameReportAction
TransformDefn	UserAddressListTransform	ViewFrameReportGraphAction
TransformDefnImpl	UserBean	ViewFrameReportGraphLoaderAction
TransformDefsns	UserConnection	ViewLabelProvider
TransformDefnsImpl	UserConnectionManipulator	ViewMultipleReportsAction
TransformDoesNotExistException	UserConnectionResults	ViewReportAction
TransformInstance	UserConnectionSource	ViewReportGraphAction
TransformKeyMapper	UserContactsBrowserBean	ViewReportGraphLoaderAction
Transforms	UserContactsBrowserCartTransform	ViewReportObjectExplorerEventHandler
Transition	UserDisconnectionFailure	Visualization
Transitions	UserEntitiesBean	VisualizationList
Tree	UserEntitiesBrowserBean	VisualizationListPropertiesWizardPage
TreeBox	UserEntitiesSelectorBean	VisualizationListWizard
TreeBoxImpl	UserEntitiesSelectorTreeCartTransform	VisualizationPropertiesEditorBean
TreeCart	UserEntitiesTransform	VisualizationPropertiesEditorBeanImpl
TreeCartExpression	UserEntitiesTreeCartImpl	VisualizationPropertiesEditorTransform
TreeCartExpressionImpl	UserEntityBean	VisualizationPropertiesTransform
TreeCartImpl	UserEntityEditorTransform	Visualizations
TreeNode	UserEntityInfoTask	VisualizationsEditorBean
TreeObject	UserEntityPrivilegesTransform	VisualizationsEditorBeanImpl
TreeParent	UserGroupBean	VisualizationsEditorTransform
TreeTabManagerTransform	UserGroupTransform	VisualizationSelectionListener
TreeView	UserLicenseAudit	VisualizationSettingsListProperties
TreeViewImpl	UserLicenseInfo	VisualizationSettingsTreeObject
TrustRelationshipExists	UserMgrDelete	VisualizationsHelper
TrustRelationshipExists.Factory	UserMgrEdit	VisualizationStyleSelectionComboBoxDialog
TrustRelationshipExistsResponse	UserMgrNew	WaitTag
TrustRelationshipExistsResponse.Factory	UserMgrNewGroup	WaitTagHelper
TrustRelationshipTransform	UserMgrNewGroupStandalone	WebAccessControlEntry
TypeFP	UserMgrNewUser	WebAccessControlList
TypeFPDetailsPage	UserMgrNewUserStandalone	WebACLParse
TypeUnsupportedException	UserParameterHelper	WebAnnotatable
UnconnectedServersTransform	UserParameterTemplate	WebAPIErrorCodes
UnmodifiableException	UserSearchBean	WebAppBeanErrorImpl
UnsynchronizedStack	UserServicesTask	WebAppConfigurationRuntimeException
UpdateFilterInfo	ValidationHelper	WebAppException
UpdateManagerHelper	ValuefulTag	WebAppFactory
UpdateManagerTag	ValueTag	WebAppRuntimeException
UpdateManagerTagHelper	ValueTagHelper	WebAppSessionManager
UpdateMobileConfigurationPropertiesTask	Version	WebAppSimpleThreshold
UpdateMobileConfigurationTask	ViewBean	WebAttribute
UpdateXML	ViewBeanParserBean	WebAttribute
UpgradeException	ViewBeanTransformHelper	WebAttributeForm
UpgradeInfo	ViewContentProvider	WebAttributeForms
UpgradeManager	ViewDisplayModeHelper	WebAttributeHeader
UpgradeModule	ViewFilterEditorRWTransform	WebAttributeList
UpgradePreviousVersionsCheatSheetAction	ViewFilterElementPanelBean	WebAxis
UpgradeWizardConclusionDialog	ViewFilterElementPanelBeanImpl	WebAxisSubtotalInstance
UpgradeWizardIntroductionPage	ViewFilterElementPanelEventHandler	WebBean
UploadFileTask	ViewFilterElementPanelTransform	WebBeanError
UploadFileTask.UploadFileParseOptions	ViewFilterElementRWTransform	WebBeanException

WebBeanFactory	WebConfigFileStorage	WebDocumentMessage
WebBeanInfo	WebConfigFileStorageEditorInput	WebDocumentSource
WebBeanInfoList	WebConfigLocationDialog	WebDocumentStatement
WebBeanProperty	WebConfigPerspective	WebDrillAction
WebBeanPropertyList	WebConfigPlugin	WebDrillActions
WebBeanScalableException	WebConfigPluginException	WebDrillElements
WebBeanStyle	WebConfigPreferencePage	WebDrillInstance
WebBeanStyleList	WebConfigRadioFieldEditor	WebDrillMap
WebBeanUtils	WebConfigUtils	WebDrillPath
WebBeanUtils.SimpleRequestKeys	WebConfigUtils.	WebDSN
WebBigDecimalNode	AbstractConfigurationElementByKeyComparator	WebEditablePrivileges
WebBlockAppendListTagImpl	WebConfigUtils.IMethodComparator	WebElement
WebBlockContextTagImpl	WebConfigUtils.IntegerComparator	WebElementHelper
WebBlockDeletePropertyTagImpl	WebConfigUtils.StringComparator	WebElements
WebBlockEditBlockTagImpl	WebConnectionMap	WebElementSearchFilterBuilder
WebBlockEditListTagImpl	WebConnectionMapSource	WebElementSearchFilterBuilder.
WebBlockEditTagImpl	WebConstantNode	ParserException
WebBlockListTagImpl	WebConstantPrompt	WebElementsObjectNode
WebBlockSetPropertyParamsTagImpl	WebConstantPromptAnswer	WebElementSource
WebBlockTagImpl	WebContactsSource	WebElementsPrompt
WebBrowseHierarchyItem	WebContentStatement	WebElementsPromptAnswer
WebBrowsePath	WebCssFormatContainer	WebEvent
WebBrowsePath.	WebCSSHelper	WebEvent.Argument
WebBrowsePathObserver	WebCustomGroup	WebEventHandler
WebBrowserEditor	WebDatamart	WebEventHandlerFactory
WebBrowserEditorInput	WebDatamartTable	WebEvents
WebCacheStatement	WebDBConnection	WebEventTags
WebChannel	WebDBLogin	WebEventUtils
WebChannels	WebDBMS	WebException
WebCluster	WebDBMSImpl	WebExpression
WebClusterAdmin	WebDBRole	WebExpressionHelper
WebClusterMember	WebDefaultDisplaySettings	WebExpressionParser
WebClusterMembership	WebDefaultSort	WebExpressionParser.
WebClusterNode	WebDerivedElement	ExpressionXMLTypes
WebClusterProjectInfo	WebDerivedElements	WebExpressionPrompt
WebClusterProjectInfoCollection	WebDerivedElementsEditorObject	WebExpressionPromptAnswer
WebClusters	WebDerivedElementsHelper	WebFailedServer
WebClusterSource	WebDetailsFormatter	WebFailedServerException
WebColumn	WebDimension	WebFeatures
WebColumns	WebDimensionAttribute	WebFilter
WebComponent	WebDimensionHeader	WebFilterBase
WebComponentBasicPropertiesPage	WebDimensionLockProperties	WebFilterEditorException
WebComponentEventDialog	WebDimty	WebFolder
WebComponentEventsPage	WebDimtyPrompt	WebFormat
WebComponentFactory	WebDimtyPromptAnswer	WebFormatContainer
WebComponentFeaturesPage	WebDimtyUnit	WebFormatIterator
WebComponentModifyListener	WebDirectedAttribute	WebFormatIteratorBase
WebComponentResultsPage	WebDirectedAttributes	WebFormShortcutNode
WebComponentTask	WebDisplayHelper	WebFormSort
WebComponentWizard	WebDisplayUnit	WebFormSortConfig
WebConfigAutoCompletionProcessor	WebDisplayUnitEntry	WebFunction
WebConfigButton	WebDisplayUnits	WebFunctionFolders
WebConfigCatalog	WebDocReportInstances	WebFunctionParameter
WebConfigFileLoader	WebDocumentInstance	WebFunctionProperty

WebGraph	WebMDXDisplayUnit	WebProjectReferences
WebGraphArea	WebMDXExpressionHelper	WebProjectSetting
WebGraphAreaImpl	WebMDXReportLimitHelper	WebProjectSettings
WebGraphCoordinatesDrillAction	WebMDXSource	WebProjectSettingsGroup
WebGraphLabel	WebMDXSources	WebProjectSettingsModule
WebGraphLabels	WebMessage	WebProjectSource
WebGraphProperties	WebMessages	WebProjectStatus
WebGraphProperty	WebMessageStatusList	WebPrompt
WebGridData	WebMetric	WebPromptableNode
WebGridHeaders	WebMetricHeader	WebPromptAnswer
WebGridPageHeaders	WebMetricHierarchicalSort	WebPromptCustomStyle
WebGridRows	WebMetricHierarchicalSortConfig	WebPromptCustomStyleImpl
WebGridTitles	WebMetricHierarchicalSortConfigImpl	WebPromptCustomStyles
WebGridWidths	WebMetricSort	WebPromptCustomStylesImpl
WebGroupBySubtotalInstance	WebMetricSortConfig	WebPromptInstance
WebGraphNode	WebMimicLogFormatter	WebPromptInstances
WebGuiComponent	WebMonitor	WebPromptLocation
WebGuiComponentList	WebMonitorField	WebPrompts
WebHeader	WebMonitorProperty	WebPromptSite
WebHeaders	WebNewFormsDrillAction	WebProperties
WebHyperLink	WebNewObjectDrillAction	WebProperty
WebHyperLinkAnswer	WebNode	WebPropertyFilesEditor
WebHyperLinkAnswers	WebNTLoginInfo	WebPropertyFilesGuiPage
WebHyperLinkContainer	WebObjectCouple	WebPropertyGroup
WebHyperLinkEditorBean	WebObjectFilter	WebPropertySet
WebHyperLinkEditorTransform	WebObjectInfo	WebRankMetric
WebHyperLinks	WebObjectsAdminException	WebRDBMSLoginInfo
WeblnboxSource	WebObjectSecurity	WebRefNode
WeblServerSession	WebObjectsException	WebRelationship
WeblServerSessionList	WebObjectsFactory	WebRelationshipNode
WeblServerSessionProxy	WebObjectSource	WebRemoveMetricDrillAction
WeblServerSessionProxyFactory	WebObjectsPrompt	WebReportData
WeblServerSessionProxyFactory.Proxy	WebObjectsPromptAnswer	WebReportExcelExportSettings
WeblServerSubscriptionDevice	WebObjectsRuntimeException	WebReportExcelLayoutBuilder
WeblServerSubscriptionTrigger	WebOperatorNode	WebReportExecutionSettings
WebLDAPLoginInfo	WebOptimizedDrillPathSettings	WebReportExportSettings
WeblAPSettings	WebPDFSettings	WebReportGrid
WebLicenseDetails	WebPercentToTotalMetric	WebReportInstance
WebLicenseInfo	WebPerformanceMonitor	WebReportLayoutBuilder
WebLimitExpressionHelper	WebPrivilegeCategories	WebReportManipulation
WebLink	WebPrivilegeCategory	WebReportMessage
WebLinkConfigurationCache	WebPrivilegeEntry	WebReportPDFExportSettings
WebLinkItem	WebPrivilegeHelper	WebReportPlainTextExportSettings
WebLinkItems	WebPrivilegeOrigin	WebReportPlainTextLayoutBuilder
WebLocale	WebPrivileges	WebReportPropertiesHelper
WebLocaleObjectInfo	WebProductLicenseDetails	WebReportSource
WebLocalizationFactory	WebProject	WebReportStatement
WebLocalizationManager	WebProjectCreator	WebReportValidationException
WebLocalizedConnectionMap	WebProjectInstance	WebReportVisualizationSettingsImpl
WebMDSecurityFilter	WebProjectInstanceBase	WebRequest
WebMDUpdateSource	WebProjectInstances	WebResidueNode
WebMDXCatalog	WebProjectLookup	WebResultSetInstance
WebMDXCube	WebProjectManipulator	WebResultSetManipulation
WebMDXCubeSource	WebProjectReference	WebResultSettings

WebResultSettingsImpl	WebSubscriptionComponent	WebTime
WebResultWindow	WebSubscriptionContact	WebTimeNode
WebRow	WebSubscriptionContacts	WebTimePrompt
WebRowValue	WebSubscriptionContent	WebTitle
WebSchedule	WebSubscriptionContentDocument	WebTitleUnit
WebScheduleEvent	WebSubscriptionContentFormat	WebTools
WebSchedules	WebSubscriptionContentProperties	WebTransactableDataObject
WebScheduleSource	WebSubscriptionContentReport	WebTransactableUnit
WebScheduleTrigger	WebSubscriptionContentReportProperties	WebTransformationMetric
WebScheduleTriggerEvent	WebSubscriptionDeliveryModeEmailProperties	WebTransformException
WebScheduleTriggerTime	WebSubscriptionDeliveryModeEmailPropertiesImpl	WebTrustedServer
WebScheduleTriggerTimeDaily	WebSubscriptionDeliveryModeFileProperties	WebUnitSubtotalInstance
WebScheduleTriggerTimeMonthly	WebSubscriptionDeliveryModeHistoryListProperties	WebURLDecoder
WebScheduleTriggerTimeWeekly	WebSubscriptionDeliveryModeMobileProperties	WebURLEncoder
WebScheduleTriggerTimeYearly	WebSubscriptionDeliveryModePrintProperties	WebURLEncoder
WebSearch	WebSubscriptionDeliveryModeProperties	WebUser
WebSecurityRole	WebSubscriptionDevice	WebUserAnswers
WebSecurityRolePrivilegeCategoriesImpl	WebSubscriptionDeviceLocation	WebUserEntity
WebSecurityRoleUsers	WebSubscriptionHelper	WebUserGroup
WebServerDef	WebSubscriptionLocation	WebUserList
WebServerSetting	WebSubscriptionLocationAttributes	WebUserSearch
WebServerSettings	WebSubscriptionNotification	WebUserSecurityFilters
WebServiceAction	WebSubscriptionProperties	WebUserSecurityRoles
WebServiceClientStubDeployer	WebSubscriptionRecipient	WebUserServicesSource
WebServiceDeployer	WebSubscriptionRecipientEmail	WebUtilsException
WebServiceMethodObject	WebSubscriptionRecipientList	WebValueNode
WebServiceMethodObject.	WebSubscriptionsFilter	WebViewInstance
MethodParameter	WebSubscriptionsSource	WebViewMediaSettings
WebServiceObject	WebSubscriptionTrigger	WebVisualizationSettings
WebServicesClientSelectionPage	WebSubscriptionUserAddresses	WebVisualizationSettingsImpl
WebServicesDescriptionPage	WebSubsetViewLimitHelper	WebWorkingSet
WebServicesModifyPage	WebSubTitle	WebWrapperAttributeList
WebServicesPreferencePage	WebSubTitles	WebXMLBuilder
WebServicesResultsPage	WebSubtotalInstance	WebXMLModule
WebServicesTaskParamtersPage	WebTable	WebXmlParser
WebServicesTaskSelectionPage	WebTableSource	WFProperty
WebServiceWizard	WebTemplate	WFProperty.WFPropertySet
WebSessionInfo	WebTemplateAttribute	Widget
WebSessionInfoList	WebTemplateConsolidation	Widgets
WebShortcut	WebTemplateCustomGroup	WidgetSelectionListener
WebShortcutNode	WebTemplateDimension	WidgetSettingsListProperties
WebSimpleSecurityPluginLoginInfo	WebTemplateFormatContainer	WidgetSettingsTreeObject
WebSimpleSecurityPluginSettings	WebTemplateMetric	WidgetTransformHelper
WebSort	WebTemplateMetrics	WizardBean
WebSortConfig	WebTemplateMetricSubtotal	WizardStepBean
WebSorts	WebTemplateMetricSubtotals	WizardStepTransform
WebSourceManipulator	WebTemplateSubtotal	WizardTransform
WebStandardLoginInfo	WebTemplateSubtotals	WPSContainerServices
WebStatement	WebTemplateUnit	WPSContainerServicesImpl
WebStatsHelper	WebThreshold	WPSCredentialAccess
WebSubscription	WebThresholds	WPSNamespaceEncoderImpl
WebSubscriptionAddress	WebThresholdsContainer	WPSPortalServlet
WebSubscriptionAnswer	WebThresholdTemplateMetric	WPSPortletParmsAccess
WebSubscriptionComparatorHelper	WebThresholdText	WPSURIBuilderImpl

Write	XMLTaskEventsPage	AnalysisSubsectionViewer
WritebackBeanFactory	XMLTaskStyleParametersPage	AnalysisTableViewCell
WritebackControl	XMLTracingFormatter	AnswerPromptsTaskRequest
WritebackDefnCache	XMLTreePanel	ApplicationLauncherController
WritebackException	XMLTreePanel.XMLTreeCellRenderer	Asc
WritebackFeatures	XMLTreePanel.XMLTreeModel	MSynch::AtomicLong
WritebackFeaturesAddon	XMLUtils	Attribute
WritebackInfoBean	XMLValidator	AttributeElement
WritebackInfoBeanEventHandler	XSLTSupport	AttributeForm
WritebackInfoBeanTransform	MicroStrategy Mobile API Class List	
WritebackMacroException	ATL::_ATL_CACHEDATA	AuthenticationFormRequest
WritebackMetricInfo	ATL::_ATL_CHAINDATA	<AuthenticationPromptControllerCallback>
WritebackReportEventHandler	ATL::_ATL_INTMAP_ENTRY	AuthenticationPromptViewController
WritebackReportGridDisplayCellMetricValue	ATL::_ATL_MODULE	AutoreleaseLock
WritebackStyleButton	ATL::_ATL_OBJMAP_ENTRY	<BackgroundMaskDelegate>
WritebackStyleDropdown	ATL::_CComChainData< base, derived >	BackgroundMaskView
WritebackStyleFactory	_CodespaceRange	BandColor
WritebackStyleTextbox	ATL::_Copy< T >	BandColorTheme
WritebackUtils	ATL::_Copy< CONNECTDATA >	BarChartInfoWrapper
WritebackWriter	ATL::_Copy< LPOLESTR >	BarProps
WriteOnlyBeanTask	ATL::_Copy< VARIANT >	Base64Support
WriterPreferenceReadWriteLock	ATL::_CopyInterface< T >	BaseChartInfoWrapper
WSFException	_FILETIME	BaseColorTheme
WSFObjectInfo	_FontClassInfo	BaseEntity
WSFSearcher	_GeoMetaInfo	BaseEntityView
WSFSearchRestriction	_GUID	BaseMapWidgetHelper
XDADeveloperBean	ATL::_ICPLocator< piid >	BasePanel
XDADeveloperBeanImpl	_IDandAlias	MFormat::DecimalFormat
XDADeveloperEventHandler	_IPadClusterToolbarInfo	MFormat::DecimalFormatImpl
XDADeveloperTransform	_LARGE_INTEGER	BigDecimalNode
XmApiFileHandler	ATL::_NoAddRefReleaseOnCComPtr< T >	BookmarkRegistry
XMLAPITask	_SearchStatusInfo	BooleanPreference
XMLBuilder	_SECURITY_ATTRIBUTES	Border
XMLBuilderException	_ULARGE_INTEGER	BorderedImage
XMLDebugFormatter	_xtab_sort_error	BorderView
XMLFileModule	AboutController	BoxAnimator
XMLFolderSearchResultsTransform	AbstractReportDataService	<BoxAnimatorDelegate>
XMLFormatter	AdvancedSettingsController	BrightnessController
XMLFormatter2	AERawData	BrowseElementsService
XMLFormatterEx	MBase::Allocator< _Ty >	BrowseElementsTaskRequest
XMLMerge	MBase::Allocator< void >	BrowseHierarchyTaskRequest
XMLMergeUtils	AnalysisDocumentIterator	MBase::Buffer
XmlRenderer	AnalysisDocumentRenderer	MBase::BufferException
XmlRendererANF	AnalysisDropDownListController	BulletInfo
XmlRendererBlockVisitorANF	AnalysisDropDownListTextBox	BulletItem
XmlRendererBlockVisitorENF	<AnalysisDropdownListTextBoxDelegate>	BulletLegend
XMLStateSerializer	AnalysisPanelStackViewer	BulletProps
XMLSupportException	AnalysisPanelTitleBar	CacheHelper
XMLSupportRuntimeException	AnalysisPanelViewer	CachelDMap
XMLTags	AnalysisSelector	CachelInfo
XMLTaskAdditionalParametersPage	AnalysisSelectorDrop popover	CalculationContext
XMLTaskBasicPage	AnalysisSelectorTitleBar	CalendarArrowTip
XMLTaskBeanPropertiesPage	AnalysisSelectorViewer	CalendarArrowView
XMLTaskBeansParametersPage		CalendarBaseButtonBar

<CalendarButtonBarDelegate>	ATL::CComModule	CommonURLRequest
<CalendarCallback>	ATL::CComMultiThreadModel	Base::CompileTimeChecker< true >
CalendarCellHighlightView	ATL::CComMultiThreadModelNoCS	ComponentMemoryEntry
CalendarDayButtonBar	ATL::CComObject< Base >	MFormat::CondText
CalendarDayCell	ATL::CComObjectCachedClassFactory< Base > <ConfigurationAutoupdateCheckHandler>	ConfigurationAutoupdateCheckHandler
CalendarDayCellContainer	ATL::CComObjectLockT< ThreadModel >	ConfigurationAutoupdateCheckRequest
CalendarDayHeader	ATL::CComObjectRootBase	ConfigurationAutoupdateCheckService
CalendarDaySheet	ATL::CComObjectRootEx< ThreadModel >	ConfigurationLinkInfo
CalendarElement	ATL::CComObjectRootEx_NoLock< ThreadModel >	ConnectivitySerializableContext
CalendarEventItemRenderer	ATL::CComPtr< T >	Consolidation
CalendarGridView	ATL::CComQIPtr< T, piid >	ConsolidationElement
CalendarItemData	ATL::CComSingleThreadModel	ConsolidationHeader
CalendarLegend	CComVariant	ConstantNode
CalendarMonthButtonBar	CEBarChartRenderer	ConstantPrompt
CalendarMonthButtonItemrenderer	CEBaseChartRenderer	contained
CalendarMonthNavigator	CEChartRenderer	MBase::ContractManagerAllocException
CalendarMonthSheet	CEImageView	ControlCell
CalendarMonthView	<CEImageViewDelegate>	ControlProperty
CalendarPopoverController	CELineChartRenderer	<ControlPropertyDelegate>
CalendarPopupView	CellFieldProperties	CopyLogEntry
CalendarPropertiesStore	CellFormat	CoverSheetController
CalendarRoundCornerView	CertificateCreationViewController	<CreateCertificateViewControllerCallback>
CalendarearchBar	CertificateLoginFieldsService	CreateSessionWithProjectService
<CalendarSearchBarDelegate>	<CertificatePromptCallback>	Credentials
CalendarSeparatorView	CertificateServerRequest	CredentialsHelper
CalendarTile	CGlobalSubtotalMap	<CredentialsPromptCallback>
CalendarTooltipViewer	CGlobalToLocalSubtotalMap	CredentialsValidationService
CalendarTopLevelController	CGtoNSValueUtils	MSynch::CriticalSection
CalendarTopView	CGUtils	CrosstabData
CalendarView	ChangePasswordRequest	<CrosstabDataProtocol>
CalendarWeekButtonBar	ChangePasswordService	CryptoManager
CalendarWeekSheet	ChangePasswordViewController	CurrentDataPointView
CalendarWidgetViewer	<ChangePasswordViewControllerCallback>	CustAlertView
CalloutTooltipViewer	ChartDataTaskRequest	CustomCellBackgroundView
Caption	ChartInfos	CustomGroup
CatalogViewController	ChartInfoWrapper	CustomGroupElement
ATL::CComAggObject< contained >	ChunkedDataReader	CustomGroupHeader
ATL::CComAutoCriticalSection	CLocalSubtotalMap	CustomGroupItem
CComBSTR	CLocalToGlobalSubtotalMap	CustomGroupItemElement
ATL::CComClassFactory	CloseSessionsService	CustomGroupItemHeader
ATL::CComCoClass< T, pclsid >	CloseSessionsTaskRequest	CustomHomeScreenSettings
ATL::CComContainedObject< Base >	CMap	CustomSegmentedControl
ATL::CComCreator< T1 >	CMapParser	CustomTableView
ATL::CComCreator2< T1, T2 >	CMapSubstitution	CustomTextField
ATL::CComCriticalSection	CMPopTipView	CustScrollView
ATL::CComDynamicUnkArray	<CMPopTipViewDelegate>	<CustScrollViewDelegate>
ATL::CComEnum< Base, piid, T, Copy, ThreadModel >	ColorPane	CustSlider
ATL::CComEnumImpl< Base, piid, T, Copy >	MFormat::ColorText	DashboardServiceFactory
ATL::CComEnumOnSTL< Base, piid, T, Copy, CollType, ThreadModel >	ColumnHeaderItemRenderer	DataElement
ATL::CComFailCreator< hr >	ColumnHeaders	DataExplorerService
ATL::CComFakeCriticalSection	ColumnInfo	DataExplorerServiceResponse
ATL::CComEnum< T >	Command	<DataProviderProtocol>
	Commander	<DataRowProtocol>
		DatasetResolvedMacros

DataSourceCacheManager	DSSAEHelper	DSSConsolidation
DataSourceUtil	DSSAggMetric	DSSConsolidationElement
<DataViewer>	DSSAggregateGeneralFunc	DSSConsolidationElementProxy
DataViewerImpl	DSSAggregationFunction	DSSConsolidationSorter
MFormat::DateTimeFormat	DSSAllElementProxy	DSSConstantSlice
MFormat::DateTimeFormatImpl	DSSAODeltaXML	DSSCube
MFormat::DecimalFormatWrapper	DSSAODisplayMode	DSSCubeDataSlice
DefaultCredsController	DSSAOGroupBy	DSSCustomGroup
DefaultProjectProperties	DSSAOLayout	DSSCustomGroupSorter
DefaultProjPropService	DSSAOPanel	DSSDataColumn
DefaultProjPropTaskRequest	DSSAOSelection	DSSDataColumnSimple< T >
<DeferredExecutionHandler>	DSSAOSetObjectProperties	DSSDataColumnString< T >
DeferredExecutionManager	DSSAOSort	DSSDataConverter
<DeferredExecutionStatusListener>	DSSAOUnsetSelectorControl	DSSDataElement
DeferredExecutionStatusReport	DSSAttribute	DSSDataElements
MBase::DeleteArray< T >	DSSAttributeElementProxy	DSSDataModelBase
MBase::DeleteC< T >	DSSAttributeForm	DSSDataModelControlImpl
DeleteMessageTaskRequest	DSSAttributeFormSorter	DSSDataModelFieldGroupImpl
MBase::DeleteOperatorGeneric< T >	DSSAttributeFormWrapper	DSSDataModelGroupbyImpl
<DeprecatedUIApplicationMethods>	DSSAttributeFormWrappers	DSSDataModelTransaction
DerivedReportInfo	DSSAxis	DSSDataModelXTabImpl
Desc	DSSBandElementProxy	DSSDataSource
Description	DSSBaseElementProxy	MDatumType::DSSDate
MBase::Destroy< T >	DSSBaseElementsProxy	DSSDerivedElementSorter
DetachHelper	MDatumType::DSSBigDecimal	DSSDimension
DeviceFeatures	MDatumType::DSSBigInt	DSSDimMetricEvaluator
DevicePreferences	DSSBinaryDataReader	DSSDocumentDefinition
DeviceUnlockManager	DSSCalculationEvent	DSSDocumentInstance
Dimty	DSSCalculationEventEvaluator	DSSDocumentSubtotal
DisplayInfo	DSSCalculationHelper	DSSDocumentSubtotalRefs
DisplayObject	DSSCalculationPlan	DSSDocumentSubtotals
DocumentContext	MDatumType::DSSCellFmtData	DSSDrillPath
DocumentIterator	DSSCellHandle	DSSDrillPathRefs
<DocumentIteratorDelegate>	DSSCExpression	DSSDrillPaths
DocumentRenderer	DSSCMacroManipulation	DSSExpNode
DocumentRendererFactory	DSSCMutation	DSSExpNodeConstant
<DownloadModalControllerDelegate>	DSSCMutationCommit	DSSExpNodeElementsObject
DrilledReportInfo	DSSCMutationDataChange	DSSExpNodeFormShortcut
DrillInfo	DSSCMutationDrill	DSSExpNodeOperator
DrillPathItem	DSSCMutationMarkRow	DSSExpNodeShortcut
DrillRadarBackgroundView	DSSCMutationSetActionOnControl	DSSField
DrillRadarView	DSSCMutationSetControlProperties	DSSFields
<DrillRadarViewDelegate>	DSSCMutationSetCurrentPanel	DSSFilter
DrillService	DSSCMutationSetGroupByElement	DSSFilterElementProxy
DrillSet	DSSCMutationSetLayout	DSSFormat
DrillTree	DSSCMutationSetSelectionFromControl	DSSFormatProperty
DrillXMLCommandParameters	DSSCMutationSetSelectionFromTemplate	DSSFormatPropertySet
DropDownListController	DSSCMutationSetTemplateDisplayStyle	DSSFormatPropertySetMap
<DropDownListControllerDelegate>	DSSCMutationSetUnsetFromControl	DSSFormatPropertyDefinition
DropdownListTextBox	DSSCMutationSortTemplate	DSSFunctionObject
<DropdownListTextBoxDelegate>	DSSCMutationToggleCurrentSelectionOnControl	DSSFunctionServer
DSSAccessControlEntry	DSSColumnMappingInfo	DSSGraphProperties
DSSActionObject	DSSCompoundMetricEventEvaluator	DSSGroupByUnit
DSSActionObjectFactory	DSSConcreteElementProxy	DSSGroupByUnits

DSSImageCacheManager	DSSSubtotalEventEvaluator	EditElementSelectionsController
DSSImageCacheManagerUtils	DSSSumFunction	EditListController
DSSInputControlSetting	DSSTabularAttribute	Element
DSSJointConcreteElementProxy	DSSTabularConsolidation	ElementHeader
DSSLevelResolver	DSSTabularCustomGroup	ElementHelper
DSSMDSYSOBJECTINF	DSSTabularIndex	ElementIDParserOld
DSSMemoryGovernorProxy	DSSTabularIndexPool	ElementInfo
DSSMetric	DSSTabularMetrics	ElementListController
DSSMetricDefaultSorter	DSSTabularUnit	ElementRadioButton
DSSMetricElementProxy	DSSTemplate	<ElementRadioButtonDelegate>
DSSMetrics	DSSTemplateAttribute	ElementSelectionsController
DSSMetricSorter	DSSTemplateConsolidation	ElementSelectorItem
DSSModelPopulation	DSSTemplateCustomGroup	ElementsHandler
DSSNode	DSSTemplateDimension	ElementsObjectNode
DSSNodeElementProxy	DSSTemplateMetric	ElementSort< T >
DSSNullElementProxy	DSSTemplateMetrics	ElementsPrompt
DSSObjectContext	DSSTemplateUnit	ElementStateButton
DSSObjectInfo	DSSThreshold	<ElementStateButtonDelegate>
DSSOBJECTINST	DSSThresholds	Encoding
DSSPerformanceProxy	DSSTimeLogger	Entity
DSSProcessMetrics	DSSTimeLoggerEntry	EntityContainer
DSSProperty	DSSTransactionReport	EntityContainerView
DSSPropertyBaseClass	DSSUnitObject	EntityValues
DSSPropertySet	DSSUnitObjects	EntityView
DSSPropertySetBaseClass	DSSUserRuntime	<EntryDelegate>
DSSRWControl	DSSViewMap	EntryTracker
DSSRWControlIterator	DSSXTabHashTable	EPUBDownloadModalController
DSSRWControlNode	DSSXTabHeader	EPUBmanager
DSSRWControls	DSSXTabView	ErrorDetails
DSSRWDEngine	DuplicateMessageRequest	ErrorHandlingHelper
DSSRWDHelper	DynamicFieldFactory	ErrorInfo
DSSRWFieldGroupIterator	DynamicFieldViewerController	EventContext
DSSRWFieldGroupNode	EditableGridCellView	EventLoggerHandler
DSSRWGraphIterator	EditableLabel	EventSerializer
DSSRWGridIterator	EditableNumberGridCellView	MBase::ExceptionImpl< T >
DSSRWGroupbyIterator	EditablePopoverGridCellView	ExecuteDocumentTaskRequest
DSSRWIterator	EditableSliderGridCellView	ExecuteReportTaskRequest
DSSRWNode	EditableSwitchGridCellView	Expression
DSSRWSectionIterator	EditControl	ExpressionHelper
DSSRWSectionNode	EditControlDatePicker	ExpressionNode
DSSRWTemplateNode	<EditControlDelegate>	ExpressionPrompt
DSSSecondClassObject	EditControlElementList	DSSCellHandle::extend
DSSSimpleFunction	EditController	FieldGroupCell
DSSSimpleOperatorFunc	EditControlList	FieldGroupRow
DSSSliceCalculator	EditControlNumber	FieldGroupViewer
DSSSort	EditControlProperty	FieldLink
DSSSortConverter	EditControlSlider	FieldModel
DSSSortDefinition	EditControlSwitch	FieldRowController
DSSSorter	EditControlText	FieldRowModel
DSSSorts	EditControlTextArea	FieldTransactionDelegate
MBase::DSSStrongPtr< T , DSSDeleteOperator >	EditControlToggle	FileDownloadRequest
DSSSubExpressionElementProxy	EditControlViewer	FileDownloadService
DSSSubtotalElementProxy	<EditControlViewerDelegate>	FileUtils
	<EditControlWriteBack>	Fill

Filter	GMTreeLayerPos	GridWidgetInfo
FlagView	GMTreeScrollView	GridWidgetRowDisplayInfo
FlagViewTA	GMTreeScrollView_Size_Info	<GridWidgetViewDataSource>
Folder	GMZoomedInRenderer	<GridWidgetViewDelegate>
FolderBrowseTaskRequest	GMZoomScale	GridWidgetViewer
FolderBrowseTaskResponseHandler	Gradient	GroupbyCell
FolderItemView	GradientColor	<GroupbyDelegate>
FolderViewController	GradientColorTheme	GroupbyDropDownPopover
Font	GradientColumnView	GroupbyElement
FontMetric	GradientInfo	GroupbyElementPickerViewer
<Formatter>	GradientView	<GroupbyElementPickerViewerDelegate>
FormatUtilities	GradientViewer	GroupbyListViewer
FormsHelper	GraphDrillPath	GroupbyViewer
FormShortcutNode	GraphDrillPathCollection	GroupElement
FractionNumber	GraphDrillTaskRequest	GTMBase64
FullScreenImageView	GraphicUtils	MBase::GUID
DSSRVIterator::GBElement	GraphMatrixDynShowTaskParam	Handler
Generic	GraphMatrixPropertiesStore	Header
GenericDrillTaskRequest	GraphMatrixSliceTaskParam	HeaderElementInfo
GenericMSTRWebViewController	GraphMatrixWidgetViewer	HeaderValue
GenericSelectorItem	GraphMatrixWidgetViewer2	HeatMapControlsController
<GesturesHandler>	GraphOverlayLayer	HeatMapData
GetCacheUpdateTimes	GraphProperty	HeatMapDeletedItemsController
GetCertificateService	GraphTooltipContainerViewer	<HeatMapDeletedItemsDelegate>
GlossGradientView	GraphTooltipViewer	HeatMapDeletedItemsTableCell
GLOWLAYER	GraphViewer	HeatMapLegendView
GM2_ChartPosition	GridBorderWidths	HeatMapMenuController
GM2ChartInfoWrapper	GridCell	HeatMapPropertiesStore
GM2ChartRenderer	GridCellHelper	<HeatMapWidgetDelegate>
<GM2Delegate>	GridCellMeasurer	HeatMapWidgetViewer
GM2ItemLegendViewer	GridCellPropertySet	HelpController
GM2LegendContentViewer	GridCellView	<HelpControllerDelegate>
GM2PropertiesStore	<GridCellViewDelegate>	HighlightBar
GM2RangeLegendViewer	GridContainerView	HighlightLabel
GM2ScrollManager	<GridContainerViewDelegate>	HighlightLayer
GM2ScrollViewRenderTask	GridDrillTaskRequest	<HomeItemViewControllerDelegate>
GM2ScrollViewRenderTaskPool	GridGraphViewer	HomeScreenBackgroundSettings
GM2Utility	GridIterator	HomeScreenBackgroundView
GM_LEGEND_ITEM_INFO	GridManipulationService	HomeScreenButton
GM_SepLine_Info	GridManipulationTaskRequest	HomeScreenButtonActionSettings
GM_Subview_Position	GridTaskParam	HomeScreenButtonLayoutSettings
GM_Subview_Span	GridTransactionDelegate	HomeScreenButtonList
GM_TEMPLATE_UNIT	GridViewer	HomeScreenButtonSettings
GM_TEMPLATE_UNIT_Info	GridWidgetCell	HomeScreenController
GMDaDataPointCircleViewer	GridWidgetCellLabel	HomeScreenFolder
GMGirdScrollView	<GridWidgetCellLabelDelegate>	HomeScreenFormatSettings
GMHeaderTree	GridWidgetColumnDisplayInfo	HomeScreenItemButtonView
GMLayoutInfo	GridWidgetColumnDisplayInfoCollection	<HomeScreenItemButtonViewDelegate>
GMMetricInfo	GridWidgetColumnToggleView	HomeScreenItemView
GMPinchGestureRecognizer	GridWidgetDrillInfo	HomeScreenResultSet
GMScrollView	GridWidgetEditedIndicatorView	HomeScreenSettings
GMTextItemRenderer	<GridWidgetHeaderDelegate>	HomeScreenTitleBarSettings
GMToltipViewer	GridWidgetHeaderItem	HomeScreenViewSettings
GMTouchPointAndView	GridWidgetHeaderView	HorizontalSectionViewer

HTMLFieldModel	IPadDataControllerFactory	IPadWaitView
HTMLViewer	IPadDataControllerImpl	IPersist
IClassFactory	IPadEnforceOrientationController	IPersistMemory
Icon	IPadFolderController	IPersistStream
IConnectionPoint	<IPadFolderControllerDelegate>	IPersistStreamInit
IConnectionPointContainer	IPadFolderControllerState	IPhoneFolderPortraitView
ATL::IConnectionPointContainerImpl< T >	IPadFolderHelper	IPhoneInfoWindowViewController
ATL::IConnectionPointImpl< T, piid, CDV >	IPadFolderItem	IPhonePromptHelper
ICreateErrorInfo	IPadFolderItemArray	IRecordInfo
IDispatch	IPadFolderItemCustomHome	ISequentialStream
ATL::IDispatchImpl< T, piid, plibid >	<IPadFolderItemDelegate>	IServerCacheInfo
IEnumConnectionPoints	IPadFolderItemDetailsController	IServerCacheInfoMap
IEnumConnections	<IPadFolderItemDetailsControllerDelegate>	IsSessionAliveTaskRequest
ATL::IEnumOnSTLImpl< Base, piid, T, Copy, CollType >	IPadFolderItemDetailsView	IStatisticsInfo
IEnumVARIANT	IPadFolderItemFolder	IStream
IErrorInfo	IPadFolderItemPortraitView	ISupportErrorInfo
Image	<IPadFolderItemPortraitViewDelegate>	ATL::ISupportErrorInfoImpl< piid >
ImageCache	IPadFolderItemProject	ItemCircle
<ImageCacheCallback>	IPadFolderItemReport	<ItemRendererDelegate>
ImageCacheCallbackUnit	IPadFolderItemServerInteraction	Iterator
ImageFieldModel	IPadFolderItemSharedLibrary	ITypeComp
ImageFileInfo	IPadFolderItemSubscriptionHome	ITypeInfo
ImageItemRenderer	IPadFolderLandscapeView	ITypeLib
<ImageModelDelegate>	IPadFolderModel	IUnknown
ImageRequestImpl	<IPadFolderModelDelegate>	<IViewer>
ImageService	IPadFolderNavigationBarSettings	<IViewerContainer>
ImageUploaderWidgetView	IPadFolderPortraitView	<IWidgetProtocol>
ImageViewer	IPadFolderPortraitViewTableViewCell	IWImageview
ImageViewerDataManager	<IPadFolderPortraitViewTableViewCellDelegate>	KeyManager
ImageViewerHelper	IPadFolderTableView	<LatencyCallback>
ImageViewerWidget	IPadFolderView	LatencyManager
IMallocSpy	<IPadFolderViewDataSource>	LatencyNotificationView
IMarshal	<IPadFolderViewDelegate>	LatencyService
InboxMessageTimeTaskRequest	IPadLinkedDrillService	LatencyTaskRequest
InfoPListHelper	<IPadMapWidgetDataSource>	Layout
InfoWindow	<IPadMapWidgetDelegate>	LayoutContainerViewer
InfoWindowContainerViewer	IPadMapView	<LayoutDelegate>
<InfoWindowDelegate>	IPadMWAnnotation	LayoutIndicator
<InfoWindowHost>	IPadMWAnnotationView	LayoutInfo
InitializeApplicationRequest	IPadMWCalloutView	<LayoutInfoDelegate>
InitializeApplicationService	IPadMWNNetworkLayer	LayoutItem
InitialPageHelper	IPadMWView	LayoutPaginationControl
<InputFieldValueDelegate>	IPadMWVizPropInfo	LayoutSelectorItem
InputFieldView	IPadPageByElementSelectionController	LayoutTabControl
<InputFieldViewDelegate>	<IPadPageByElementSelectionControllerDelegate>	<LayoutTapControlDelegate>
IntegerPreference	IPadPromptHelper	LeftYearBar
IPadActionSelector	IPadPromptsService	LegalTextViewController
IPadAuthenticationPromptViewController	IPadReconcileService	LegendRenderer
IPadCacheCheckService	IPadRefreshService	LegendView
IPadDashboardViewerAppDelegate	IPadReportNavigationController	<LegendViewDelegate>
IPadDashboardViewerViewController	IPadRWDataController	LessOperator
<IPadDataController>	IPadRWDataService	LevelAttributes
	IPadSubscriptionCheckService	DSSCalculationPlan::LevelInfo
		LineChartInfoWrapper

LineDrawer	MenuBarSelectorItem	MSTRUWindowEventListerImpl
LineFieldModel	<MenuSelectorItem>	MSTRVCRetriever
LineViewer	MBase::Message	<MSTRVCRetrieverDelegate>
Link	MessageBasedTaskRequest	MSTRWebView
LinkAnswerHandler	MessageLock	MSTRWebViewController
LinkDrillingXMLHandler	Metric	<MSTRWebViewDelegate>
LinkDrillParameters	MetricExpression	MSTRWidgetHelper
LinkedDrillService	MetricHeader	MultiComponentSelector
LinkedDrillTaskRequest	MetricInfoWrapper	MultiDeleteButton
LinkHighlightView	MetricOperator	MultiSelectTableViewController
Links	MetricProps	MWAnnotation
LiveReportInfo	MetricSlider	MWAnnotationHiliteView
MBase::Message::Localizer	<MetricSliderDelegate>	MWAnnotationView
MSynch::LockOrderViolationException	<MetricsListControllerDelegate>	MWCalloutView
Log	MetricValue	MWDynamicBubbleView
LogController	MetricValueItemRenderer	MWMetricsDropDownListController
<LogEntry>	MIActionSheet	MWMetricToggleView
Logger	MIBadgeView	MWNNetworkElement
MSynch::Logger	MIBorderedImageView	MWNNetworkLayer
LoginAlertDialog	<MIBorderedImageViewDelegate>	MWView
LoginDialog	MicroChartFontInfo	<MWViewDataSource>
<LoginDialogCallback>	MicrochartsPropertiesStore	<MWViewDelegate>
LoginPromptViewController	MicroChartsSliceTaskParam	MWVizPropInfo
<LoginPromptViewControllerCallback>	<MicroChartsWidgetDelegate>	NamedFolderIDTaskRequest
LoginTaskRequest	MicroChartsWidgetViewer	<NetworkRequest>
LogLevelController	MILeftRightLayoutCell	NetworkRequestImpl
LogRecord	MISQLWritebackService	NetworkRequestImpl2
LogStore	MIViewController	NetworkRequestMock
LogStoreHandler	MoreButtonView	<NetworkResponseHandler>
MainLogger	<MoreButtonViewDelegate>	NetworkStatusCell
ManagedLogEntry	MSIAsymmetricCryptor	NetworkStatusController
MapWidgetController	MSICryptor	NGBSelections
MapWidgetControllerHelper	MDataType::MsiDateTime	MBase::NoCopy
MapWidgetHelper	MSIKeyStorage	NodeAndControls
MapWidgetInfo	<MSTRActionDelegate>	NonAntiAliasingLabel
MapWidgetViewer	MSTRAppContext	NormalGridCellView
<MapWidgetViewerDelegate>	<MSTRAppDelegate>	MFormat::NumberFormat
MapWidgetViewerHelper	MSTRCacheLRU	MFormat::NumberFormatImpl
MarkRowView	MSTRDoublyLinkedList	NumberFormatter
<MarkRowViewDelegate>	MSTRError	ObjectInfo
MaskView	MSTRException	ObjectInfoService
<MaskViewDelegate>	MSTRFolder	ObjectInfoTaskRequest
MasterChartView	MSTRFormatHelper	MBase::ObjectScopeGuardImpl0< Object,
MatrixLayoutInfo	MSTRFormatter	MemberFunction >
MBProgressHUD	<MSTRHandler>	MBase::ObjectScopeGuardImpl1< Object,
<MBProgressHUDDelegate>	MSTRHTMLForm	MemberFunction, Parameter1 >
MBRoundProgressView	MSTRLinkedHash	MBase::ObjectScopeGuardImpl2<
MBase::MemoryContract	MSTRLinkedList	Object, MemberFunction, Parameter1,
MemoryGoverningController	MSTRMobileAppDelegate	Parameter2 >
MemoryGovernor	MSTRMultipleError	ObjectService
MemoryLimitController	MSTRNavigationController	ObjectStore
<MemoryMeasurerDelegate>	MSTROBJECTViewController	ObjectUtils
MenuBar	MSTRUWindow	MDataType::ODBCDateTime
MenuBarItem	<MSTRUWindowEventLister>	OperationInfo
		Operator

<OperatorCallback>	PDFLink	PDFStack
OperatorNode	PDFMMType1Font	PDFTextExtractor
OptionsToolbar	PDFMultiPageDelegate	PDFTextState
<OptionsToolbarDelegate>	PDFNode	PDFThumbnailListViewController
Ordinal_System	PDFNodeReader	PDFTiledPageView
OutlinedUILabel	PDFObject	PDFTrueTypeFont
<PageBuilderDelegate>	PDFOpBeginText	PDFType0Font
PageByControl	PDFOpConcatenate	PDFType1Font
<PageByDelegate>	PDFOpEndText	PDFType3Font
PageByElementCell	PDFOperator	<PDFViewDelegate>
PageByElementPickerView	PDFOpGRestore	PDFXObjectReader
<PageByElementPickerViewDelegate>	PDFOpGSave	PDStream
PageByElementSelectionController	PDFOpMoveAndShow	PerformanceDiagnosis
<PageByElementSelectionListDelegate>	PDFOpMoveText	PerformanceTimeStamp
PageByInfo	PDFOpMoveTextSetLeading	PivotGridService
PageByItem	PDFOpNextLine	PivotGridTaskRequest
PagebyModalView	PDFOpSetCharSpacing	PivotTextItemRenderer
PageByService	PDFOpSetMatrix	MBase::Message::PlacementArrayPtr
PageByTaskRequest	PDFOpSetMoveAndShow	MBase::Message::PlacementPtr
PageElement	PDFOpSetTextFont	DSSCellHandle::pointer
PageHeader	PDFOpSetTextLeading	PointObject
PageIndicator	PDFOpSetWordSpacing	PollStatusTaskRequest
<PageIndicatorDelegate>	PDFOpShowText	PopoverContext
PanelPreview	PDFOpShowTextGlyph	<PopoverDelegate>
<PanelPreviewDelegate>	PDFOutlineItemReader	PopupAlertView
PanelStackViewer	PDFPage	<Preference>
PanelViewer	PDFPageBuilder	PreferencesStore
<PanelViewerDelegate>	<PDFPageDrawingListener>	PreloadDataManager
PDFCatalogReader	PDFPageInfo	PreloadDataService
PDFCIDFont	PDFPageReader	PreloadReportDataService
PDFCIDType0Font	PDFPageSnapViewController	PreloadRWDataService
PDFCIDType2Font	PDFPageThumbView	PreloadRWDataServiceBI
PDFColorSpaceReader	PDFPageView	PreloadRWDBinaryDataService
PDFContainer	<PDFPageViewDelegate>	PreloadRWTaskRequest
<PDFContainerDelegate>	PDFParserBase	<PreloadServiceHandler>
PDFContainerMultiPage	PDFReaderUtil	<PreloadStatusListener>
PDFContainerSinglePage	PDFResources	PreloadStatusReport
PDFCType1Font	PDFResourcesReader	PreSQLWritebackTaskRequest
PDFDelegate	PDFScrollViewController	<ProjectAuthModeCallback>
PDFDestination	PDFScrollView	ProjectAuthModeController
PDFDocument	<PDFScrollViewDataSource>	ProjectCell
PDFDocumentController	PDFSearchAttributedString	ProjectConfiguration
<PDFDocumentControllerDelegate>	PDFSearchResultCell	ProjectConfigurationRootFolderSettings
PDFExtGStateReader	PDFSearchStatusCell	ProjectConfigurationScheduleSettings
PDFFont	PDFSearchViewController	ProjectCredentials
<PDFFontDescriptor>	<PDFSearchViewControllerDelegate>	ProjectDefaultCredsController
PDFFontDescriptorDict	PDFSeeker	ProjectInfo
PDFFontDescriptorAFM	<PDFSeekerDelegate>	ProjectInfoController
PDFFontFactory	<PDFSelectionDelegate>	ProjectList
PDFFontReader	PDFSimpleFont	ProjectListController
PDFGraphicsState	PDFSinglePageDelegate	Prompt
PDFGraphInfo	PDFSlider	PromptAnswer
PDFLayer	PDFSnapHighlightView	PromptBarcodeCell
PDFLineDash	PDFSnapView	PromptBaseCell

PromptCalendarCell	ReportDataService	Parameter1 >
PromptControllersFactory	ReportDetailsCell	MBase::ScopeGuardImpl2< Function, Parameter1, Parameter2 >
PromptFormattableCell	ReportDisplayInfo	MBase::ScopeGuardImpl3< Function, Parameter1, Parameter2, Parameter3 >
PromptHelper	ReportExecuteXMLService	MBase::ScopeGuardImpl4< Function, Parameter1, Parameter2, Parameter3, Parameter4 >
<PromptHelperDelegate>	ReportGraphTaskRequest	MBase::ScopeGuardImplBase
PromptInformation	ReportGridTaskRequest	ScrollIndicator
PromptLocationCell	ReportInfo	SearchResultViewController
PromptPickerCell	ReportInfoHandle	<SearchResultViewControllerDelegate>
Prompts	ReportListTaskRequest	SectionHeaderViewer
PromptsAnswerHandler	ReportStore	SectionViewer
PromptsAnswerXMLHandler	RequestFactory	SegmentViewController
PromptSliderCell	RequestStatistic	<SegmentViewDelegate>
PromptsService	ResolvedLinkXMLHandler	SelectedElementListController
PromptsTaskRequest	MBase::ReturnPtr< T, DeleteOperator >	SelectedItems
PromptStepperCell	MBase::ReturnPtrl< ReferenceCountedT >	SelectionCoordinate
PromptStepperLabelView	MBase::ReturnString	SelectionHelper
PromptStepperOperatorBackgroundView	MBase::ReturnUString	Selector
PromptStepperOperatorSignView	MBase::ReturnWString	<SelectorDelegate>
PromptStepperOperatorView	RootViewerContainer	SelectorDropDown
PromptStepperView	RoundedRectView	<SelectorDropDownDelegate>
PromptSummaryScreen	RowHeaderItemRenderer	SelectorDropDownPopover
PromptSummaryScreenCell	RWDataService	<SelectorItem>
PromptSwitchCell	RWDataServiceResult	SelectorListbox
PromptTextCell	RWDBinaryDataCollector	SelectorMetricMenu
PromptTimePickerCell	RWDBinaryDataEntry	SelectorMetricSlider
PromptUtils	RWDBinaryDataService	SelectorRadioButton
Properties	RWDBinaryTaskRequest	SelectorSlider
PropertyGroup	RWDashboardDataViewer	SelectorSliderMulti
PSBorder	<RWDashboardDelegate>	SelectorStateButton
<PushNotificationListener>	RWDashboardView	SelectorStateButtonSingleHorizontal
Queue	RWDImageService	SelectorViewer
RadarLabel	RWDImageServiceTaskRequest	<Serializable>
<RadarLabelDelegate>	RWDManipulationTaskRequest	SeriesLabel
RangeSlider	RWDMemoryMeasurer	<Service>
<RangeSliderDelegate>	RWDOfflineBinaryLoader	ServiceFactory
RankPred	RWDOfflineMode	<ServiceHandler>
ReaderSettings	RWDrillAndBinaryDataService	ServiceImpl
MBase::Allocator< _Ty >::rebind< _UT >	RWDrillService	SessionService
MBase::Allocator< void >::rebind< _UT >	RWGetInboxMsgs	SessionStateWrapper
<ReconcileListener>	RWGridDrillTaskRequest	SetAndPathName
ReconcileService	RWLinkDrillAndBinaryDataService	SettingController
<ReconcileServiceHandler>	RWManipulationAndBinaryDataService	SettingControllerFactory
RectangleDrawer	RWManipulationService	SettingsFieldsFactory
RectangleFieldModel	RWRemoveInboxMsgs	SettingsTextField
RectangleViewer	RWSaveToInbox	ShortcutNode
MBase::ReferenceCounted	RWTaskRequest	SimpleFeedbackView
RefreshService	SampleActionSheet	SimpleFormatter
RefreshSubscriptionTaskRequest	SATypeHolder	SimpleRWDMemoryMeasurer
RegisterDeviceTokenService	SaveSubscriptionService	SimpleXMLBuilder
RegularDrillService	SaveSubscriptionTaskRequest	SimpleXMLNode
RemoveDeviceTokenService	SAXStringHandler	SimpleXMLParser
ReportCacheMap	SAXSupport	
ReportDataRow	MBase::ScopeGuardImpl0< Function >	
ReportDataCell	MBase::ScopeGuardImpl1< Function,	

SimpleXMLWriter	tagDISPPARAMS	TimeSeriesWidgetViewer
SlaveChartTask	tagELEMDESC	<TimeSpanHandleDelegate>
SlaveChartView	tagEXCEPINFO	TimeSpanHandleView
SliceInfo	tagFUNCDESC	TitleBar
SliderPopoverBox	tagIDLDESC	TitleLabel
SliderPopText	tagPARAMDESC	<TitleLabelDelegate>
SlidingControl	tagPARAMDESCEX	TOCTableCell
<SlidingControlDelegate>	tagDEC::tagParts1	TOCViewController
MSynch::SmartArrayPtr< T >	tagDEC::tagParts2	ToggleValue
SmartAutoreleasePool	tagPOINT	TooltipContext
MSynch::SmartBase< T, DeleteOperator >	tagRECT	<TooltipDelegate>
MSynch::CriticalSection::SmartLock	tagSAFEARRAY	TooltipView
MSynch::SmartPtr< T, DeleteOperator >	tagSAFEARRAYBOUND	TooltipViewBullet
MBase::SmartPtr< ReferenceCountedT >	tagSTATSTG	TooltipViewer
SortGridService	tagTLIBATTR	TransactionCommander
SortGridTaskRequest	tagTYPEATTR	TransactionDelegate
SparklineProps	tagTYPEDESC	TransactionHelper
SpecialElement	tagVARDESC	TransactionParameters
Stack	tagVARIANT	TransactionRecord
StateTracker	Task	TransactionSubmitService
StatusCell	TaskGroup	<Tree>
<StatusChangeListener>	TaskInfo	<TreeNode>
StatusController	TaskRequest	TriangleView
StatusDetailsController	TaskRequest2	TrimTipView
StatusManager	TaskState	TypeFormatter
MBase::StringConversionException	TemplateMetricsPlaceHolder	UIDevice
StringPreference	TemplateWithProperties	UIImage(Scale)
StringUtils	TestCrosstabData	UINavigationBar
MBase::StrongArrayPtr< T >	TestCrosstabDataWithMultipleDataSets	UIView
MBase::StrongBase< T, DeleteOperator >	TestLoginTaskRequest	UniformChartData
MBase::StrongBufferPtr< T >	TestTabularData	UnitAndExtra
MBase::StrongPtr< T, DeleteOperator >	TestTabularDataWithMultipleDataSets	UnsubscribeService
StructDSSObjectReference	TestURLHelper	UnsubscribeTaskRequest
StructDSSUserAuditResult	TestVisPList	UnsupportedExpressionNode
MFormat::SubFormat	TextFieldAlertView	UnsupportedPrompt
SubscribeService	TextFieldModel	URLHelper
SubscribeServiceResult	TextItemRenderer	URLLaunchHelper
SubscribeTaskRequest	TEXTMETRIC	URLParser
SubsectionViewer	TextPosition	URWSTimeSeriesHandler
SubtotalElement	TextViewer	URWSTimeSeriesHandlerBl
SubtotalInfo	TextViewerContainer	UserRuntimeDataCache
SubviewLayoutInfo	ThresholdBand	UserRuntimeService
SummaryCell	ThresholdHelper	UserSelectionView
SummaryController	ThresholdImageDict	Utility
SwitchContentView	ThresholdSliderData	DSSCellHandle::value
SwitchScrollView	ThresholdWrapper	ValueNode
SyncScrollView	TimeInterval	VGAMirror
SystemScreenController	TimeLabel	ViewerDataModel
TabularData	TimeNode	VisPList
<TabularDataProtocol>	<TimeSelectorDelegate>	WaitView
tagBINDPTR	TimeSelectorLabel	WebRequest
tagCONNECTDATA	TimeSeriesPropertiesStore	<WebServerAuthModeCallback>
tagCY	TimeSeriesToolTipView	WebServerAuthModeController
tagDEC	<TimeSeriesWidgetDelegate>	WebServerConfiguration

WebServerCredentials	com.microstrategy.web.vf.viewer.properties	ErrorLogEntry
WebServerDefaultCredsController	com.microstrategy.web.vf.viewer.renderer	Expression
WebServerInfo		ExpressionConstructionHelper
WebServerInfoController		ExpressionFilterObject
WebServerList	MicroStrategy Visualization Framework API Class List	ExpressionOperator
WebServerTypeController	AIRSettings	ExpressionSliceCommandImpl
<WebViewControllerDelegate>	ApplicationSettings	ExpressionSliceTarget
WeekButtonBarDividersView	BaseChartTraversalCallback	FilteredUnitMetricImpl
WidgetColors	BaseCrosstabTraversalCallback	FilteredUnitSliceTarget
<WidgetGridSwitchProtocol>	BaseModel	FilterExpression
<WidgetItemRenderer>	BasePropertiesViewer	FilterSpecification
WidgetUtils	BaseTabularListTraversalCallback	FolderBrowseTask
WidgetViewer	BaseTabularTraversalCallback	IAggregationSpecification
WindowEventInterceptor	CalculatedMetric	IAttribute
XMLAPIService	CalculatedMetrics	IAttributeElement
XMLAPITaskRequest	CategoryInfo	IAttributeForm
XMLAPITaskRequest2	CommandFactory	IAttributeHeader
XMLBuilder	CommandManager	IAxis
XTabConsolidationElement	Comparable	IBaseVisualizer
XTabCustomGroupElement	Condition	ICalculatedMetric
ZoomedGridCellView	ConditionalRenderer	ICalculatedMetricMXMLObject
ZoomedInChartInfoWrapper	ConditionalRendererObject	ICalculatedMetricSpecification
ZoomedInViewController	ConnectionInfo	ICategoryAxisInfo
ZoomedXAxisView	Controller	IChartData
ZOrderHelper	ControllerEvent	IChartDataVisitor
	ControllerProgressBar	IChartItemSelector
MicroStrategy Visualization Framework API Packages	CrosstabViewer	IChartTraversalCallback
com.microstrategy.web.vf.controller	DAG	IConsolidation
com.microstrategy.web.vf.controller.render	DataSource	IConsolidationElement
com.microstrategy.web.vf.controller.tasks	DataSourceList	IConsolidationHeader
com.microstrategy.web.vf.model	DebugViewer	IControllerFormat
com.microstrategy.web.vf.model.cm	DefaultChartTraversalCallback	IControllerListener
com.microstrategy.web.vf.model.cr	DefaultCrosstabTraversalCallback	ICrosstabCell
com.microstrategy.web.vf.model.commands	DefaultRowDisplayString	ICrosstabData
com.microstrategy.web.vf.model.data	DefaultTabularTraversalCallback	ICrosstabDataVisitor
com.microstrategy.web.vf.model.expression	DiagnosticChartViewer	ICrosstabTraversalCallback
com.microstrategy.web.vf.model.filter	DiagnosticCrosstabViewer	ICustomGroup
com.microstrategy.web.vf.model.linkdrill	DiagnosticTabularViewer	ICustomGroupElement
com.microstrategy.web.vf.model.modifiers	DisplayObjectEvent	ICustomGroupHeader
com.microstrategy.web.vf.model.sort	DocumentResultsTask	ICustomGroupItem
com.microstrategy.web.vf.model.subtotal	EnumAxisName	ICustomGroupItemElement
com.microstrategy.web.vf.model.tags	EnumFilterMergingLogic	ICustomGroupItemHeader
com.microstrategy.web.vf.model.template	EnumHeaderType	IDisplayInfo
com.microstrategy.web.vf.model.traversal	EnumLogEntryTypes	IDisplayObject
com.microstrategy.web.vf.spark	EnumObjectTypes	IElement
com.microstrategy.web.vf.utils	EnumSliderSelectorTypes	IElementHeader
com.microstrategy.web.vf.viewer	EnumSubtotalDisplayPositions	IExpression
com.microstrategy.web.vf.viewer.internalClass	EnumSubtotalLevels	IExpressionFilterObject
	EnumSubtotalTypes	IExpressionNode
	EnumTemplateUnitType	IFilteredUnit
	EnumVFCCommandTypes	IFilteredUnitElement
	ErrorEvent	IFilteredUnitMetric
	ErrorLog	IFilteredUnitMetrics
		IFilteredUnits

IFilterExpression	MicroBarChart	MstrPropertiesHelper
IFilterExpressionNode	MicroBubbleChart	MstrRectangle
IFilterSpecification	MicroBulletGraph	MstrShape
IHeader	MicroColumnChart	MstrTextInput
IHeaderValue	Model	MstrTriangle
IMesh	ModelCommand	MstrViewerControlDelegate
IMetric	ModelEvent	MstrVSlider
IMetricFormat	ModelUtils	MstrWedge
IMetricFormats	MstrAirController	MstrWindowedApplication
IMetricHeader	MstrApplication	MstrWindowedApplication
IMetricValue	MstrApplication	NamespaceHelper
IModel	MstrApplicationViewer	PriorityQueue
IModelSpecification	MstrAreaChart	ReportLink
IMstrApplication	MstrArealItemRenderer	ReportLinks
IMstrDashboardWidget	MstrBarChart	ReportResultsTask
IMstrViewerControl	MstrBarItemRenderer	SelectionExpression
IReportData	MstrBoxItemRenderer	SelectionHelper
IReportDataFormat	MstrBubbleChart	SelectorMap
IReportRawData	MstrBubbleItemRenderer	SeriesInfo
IReportRawDataRow	MstrBubbleMarker	SliceTarget
IReportRawDataRows	MstrButton	SparkLineChart
IRowDisplayString	MstrButton	Subtotal
IRowsManipulator	MstrCandlestickChart	SubtotalDefinition
ISelectionData	MstrCircle	SubtotalLevelByPosition
ISort	MstrColumnChart	TabularDataHelper
ISortDefinition	MstrColumnItemRenderer	TextTraversalCallback
ISubtotal	MstrComboBox	Vertex
ISubtotalDefinition	MstrConditionalItemRenderer	VIterator
ISubtotalElement	MstrCrosstabGrid	VRenderer
ISubtotalLevel	MstrDataGrid	VRendererTask
ISubtotalMetricValue	MstrDataGridItemRenderer	ViewerEvent
ITabularData	MstrDataPointItemRenderer	VisFrameworkHelper
ITabularDataHeader	MstrDownloadProgressBar	XMLHandle
ITabularDataVisitor	MstrErrorDisplay	
ITabularListData	MstrHLOCChart	MicroStrategy Widget API Packages
ITabularListDataVisitor	MstrHSlider	com.microstrategy.flex.controller
ITabularTraversalCallback	MstrImage	com.microstrategy.flex.controller.
ITemplateAttribute	MstrImageMarker	commands
ITemplateConsolidation	MstrLabel	com.microstrategy.flex.controller.
ITemplateCustomGroup	MstrLabel	commands.updatemgr
ITemplateDefn	MstrLineChart	com.microstrategy.flex.controller.events
ITemplateFormat	MstrLineItemRenderer	com.microstrategy.flex.controller.render
ITemplateMetric	MstrList	com.microstrategy.flex.controller.
ITemplateMetricFormat	MstrList	resources
ITemplateMetrics	MstrMap	com.microstrategy.flex.controller.tasks
ITemplateSpecification	MstrMarkerMap	com.microstrategy.flex.model.common
ITemplateUnit	MstrMetricFormatter	com.microstrategy.flex.model.constants
ITemplateUnitFormat	MstrOptionGroup	com.microstrategy.flex.model.filter
ITraversal	MstrOptionGroup	com.microstrategy.flex.model.format
IValue	MstrPieChart	com.microstrategy.flex.model.
IVFController	MstrPieItemRenderer	manipulation.command
LogicOperator	MstrPlotChart	com.microstrategy.flex.model.rwd
MetricsPosition	MstrPolygon	com.microstrategy.flex.model.rwditerator
MicroAreaChart	MstrProgressBar	com.microstrategy.flex.model.styles
		com.microstrategy.flex.model.template

com.microstrategy.flex.model.xtab	com.microstrategy.flex.viewer.	AppearanceVisible
com.microstrategy.flex.utils	slideOutTrayClasses	ApplyToGraphThreshold
com.microstrategy.flex.viewer	com.microstrategy.flex.viewer.	AreaSeriesEvent
com.microstrategy.flex.viewer.	thermometerViewerClasses	Attribute
accordionClasses	com.microstrategy.flex.viewer.	AttributeElement
com.microstrategy.flex.viewer.	TimeSeriesSliderClasses	AttributeForm
chartViewerClasses	com.microstrategy.flex.viewer.	AttributeFormSorter
com.microstrategy.flex.viewer.	TimeSeriesSliderClasses.charts	AttributeUnit
colorSliderClasses	com.microstrategy.flex.viewer.	AutoResizelnDesign
com.microstrategy.flex.viewer.constants	TimeSeriesSliderClasses.charts.events	AutoSubmit
com.microstrategy.flex.viewer.	com.microstrategy.flex.viewer.utils	AvailableViewModes
cylinderViewerClasses	com.microstrategy.flex.viewer.	BackgroundAlpha
com.microstrategy.flex.viewer.	widgetsCommonClasses	BackgroundColor
dashboardWidgetsCommonClasses	com.microstrategy.flex.viewer.	BackgroundStyle
com.microstrategy.flex.viewer.	widgetsCommonClasses.events	BackgroundStyles
dropDownClasses	com.microstrategy.flex.viewer.	BandColorSlider
com.microstrategy.flex.viewer.effects	widgetsCommonClasses.labels	BandColorSliderBand
com.microstrategy.flex.viewer.effects	com.microstrategy.flex.viewer.	BandColorSliderBandHandler
effectClasses	widgetsCommonClasses.ticks	BandColorSliderDirectionSymbol
com.microstrategy.flex.viewer.effects	com.microstrategy.flex.viewer.	BandColorSliderEvent
threeDEngine	widgetsCommonClasses.ticks.circular	BandDirection
com.microstrategy.flex.viewer.	com.microstrategy.flex.viewer.	BandingEnabled
frequencyVisualizationChartClasses	widgetsCommonClasses.ticks.linear	BandingGroup1
com.microstrategy.flex.viewer.	com.microstrategy.flex.viewer.	BandingGroup2
graphViewerClasses.graphModel	widgetsCommonClasses.utils	BandingOptions
com.microstrategy.flex.viewer.	com.microstrategy.flex.viewer.	BandingRowsColumns
graphViewerClasses.graphModel.axes	widgetsCommonClasses.utils.Lasso	BandingStyleLayer
com.microstrategy.flex.viewer.		BaseAppController
graphViewerClasses.graphModel.constants		BaseAppControllerEvent
com.microstrategy.flex.viewer.		BaseCollection
graphViewerClasses.graphModel.events		BaseCommand
com.microstrategy.flex.viewer.		BaseDataLoader
graphViewerClasses.graphModel.modelBuilders		BaseDocumentPropertyValue
com.microstrategy.flex.viewer.		BaseElement
graphViewerClasses.graphModel.series		BaseEntityToolTip
com.microstrategy.flex.viewer.		BaseError
graphViewerClasses.graphModel.structure.buildingBlocks		BaseExtendedPropertyValue
com.microstrategy.flex.viewer.		BaseModel
graphViewerClasses.graphModel.structure.fills		BasePropertyValue
com.microstrategy.flex.viewer.		BaseRepeatedTask
graphViewerClasses.graphModel.structure.uiElements		BaseStyle
com.microstrategy.flex.viewer.		BaseTask
graphViewerClasses.scaleBreak		BaseToolTipMenu
com.microstrategy.flex.viewer.		BevelAreaRenderer
heatMapViewClasses		BigDecimal
com.microstrategy.flex.viewer.interactiveStackedGraphClasses		BinaryBlockType
com.microstrategy.flex.viewer.linkDrill		BinaryDataReader
com.microstrategy.flex.viewer.model		BinaryDataType
com.microstrategy.flex.viewer.properties		BinaryDefnBlockType
com.microstrategy.flex.viewer.resizeHandleClasses		BinaryDefnReader
com.microstrategy.flex.viewer.selectorViewerClasses		BlackAndWhite
		Border3DStyle
		Border3DWeight
		BorderColorBottom

MicroStrategy Widget API Class List

AccordionContainer	AccordionExpandCollapseEffect	AccordionEditor	AccordionEntry	AccordionEntryHeader	AccordionEvent	AccordionExpandCollapseEffectInstance	AccordionHandle	AccordionTitleBar	AdaptiveURLoaderTask	AddButtonBarPadding	AddFieldEvent	AddNodeEvent	AlignmentTextDirection	AlignmentVertical	AlignmentVerticalValues	AORunDocument	AORunReport	AORunRWDocument	AORWBeanManipulation	AOSetCurrentGroupByElement	AppControllerFactory	AppearanceHideDuplicates	AppearanceName	AppearanceSpecialEffect	AppearanceTooltip	AppearanceTooltipText	AppearanceVisible	ApplyToGraphThreshold	AreaSeriesEvent	Attribute	AttributeElement	AttributeForm	AttributeFormSorter	AttributeUnit	AutoResizelnDesign	AutoSubmit	AvailableViewModes	BackgroundAlpha	BackgroundColor	BackgroundStyle	BackgroundStyles	BandColorSlider	BandColorSliderBand	BandColorSliderBandHandler	BandColorSliderDirectionSymbol	BandColorSliderEvent	BandDirection	BandingEnabled	BandingGroup1	BandingGroup2	BandingOptions	BandingRowsColumns	BandingStyleLayer	BaseAppController	BaseAppControllerEvent	BaseCollection	BaseCommand	BaseDataLoader	BaseDocumentPropertyValue	BaseElement	BaseEntityToolTip	BaseError	BaseExtendedPropertyValue	BaseModel	BasePropertyValue	BaseRepeatedTask	BaseStyle	BaseTask	BaseToolTipMenu	BevelAreaRenderer	BigDecimal	BinaryBlockType	BinaryDataReader	BinaryDataType	BinaryDefnBlockType	BinaryDefnReader	BlackAndWhite	Border3DStyle	Border3DWeight	BorderColorBottom
--------------------	-------------------------------	-----------------	----------------	----------------------	----------------	---------------------------------------	-----------------	-------------------	----------------------	---------------------	---------------	--------------	------------------------	-------------------	-------------------------	---------------	-------------	-----------------	----------------------	----------------------------	----------------------	--------------------------	----------------	-------------------------	-------------------	-----------------------	-------------------	-----------------------	-----------------	-----------	------------------	---------------	---------------------	---------------	--------------------	------------	--------------------	-----------------	-----------------	-----------------	------------------	-----------------	---------------------	----------------------------	--------------------------------	----------------------	---------------	----------------	---------------	---------------	----------------	--------------------	-------------------	-------------------	------------------------	----------------	-------------	----------------	---------------------------	-------------	-------------------	-----------	---------------------------	-----------	-------------------	------------------	-----------	----------	-----------------	-------------------	------------	-----------------	------------------	----------------	---------------------	------------------	---------------	---------------	----------------	-------------------

BorderColorLeft	CDSSFilterElementProxy	CDSSTemplateConsolidation
BorderColorRight	CDSSFormat	CDSSTemplateCustomGroup
BorderColorTop	CDSSFormatProperty	CDSSTemplateDimension
BorderStyleBottom	CDSSFormatPropertySet	CDSSTemplateMetric
BorderStyleLeft	CDSSGovernor	CDSSTemplateMetrics
BorderStyleRight	CDSSGraphCell	CDSSTemplateUnit
BorderStyles	CDSSGraphHeaderCell	CDSSTemplateUnitAndForm
BorderStyleTop	CDSSGraphInfo	CDSSTemplateUnitObject
BottomLeftCornerRadius	CDSSGraphIterator	CDSSThreshold
BottomRightCornerRadius	CDSSGridCell	CDSSThresholds
CalendarColorPicker	CDSSGridIterator	CDSSUnitObject
CanDropLabelsCategory	CDSSGroupByCell	CDSSUnitObjects
CanDropLabelsLinear	CDSSGroupByElement	CElementID
CanGrow	CDSSGroupByIterator	CellDataStyleLayer
CanGrowHorizontal	CDSSGroupByUnit	CellFmtData
CanShrink	CDSSGroupByUnits	CGroupByDataModel
CanShrinkHorizontal	CDSSJointConcreteElementProxy	CheckBoxDataGridItemRenderer
CanStagger	CDSSMetric	CheckBoxValidator
CAtributeElementID	CDSSMetricElementProxy	CircularTickProps
CDaDataModel	CDSSMetrics	CircularTicks
CDesiredElement	CDSSMSTRCurrencyFormatter	Color
CDSSAggMetric	CDSSMSTRNumberFormatter	ColorGradient
CDSSAllElementProxy	CDSSNode	ColorGroup1
CDSSAttribute	CDSSNodeElementProxy	ColorGroup2
CDSSAttributeElementProxy	CDSSNullElementProxy	ColorManip
CDSSAttributeForm	CDSSNumberFormatter	ColorSlider
CDSSAttributeForms	CDSSObjectInfo	ColorSliderConvertHelper
CDSSAttributeFormWrapper	CDSSPaddingCellLeft	ColorSliderDataTip
CDSSAxis	CDSSPaddingCellUpper	ColorSliderLabel
CDSSBandElementProxy	CDSSPrivileges	ColorSliderTrackSkin
CDSSBaseAttributeForm	CDSSProperty	ColumnScenario
CDSSBaseAttributeForms	CDSSPropertySet	ColumnScenarios
CDSSBaseElementProxy	CDSSReportDefinition	ComboBoxEx
CDSSBaseElementsProxy	CDSSReportInstance	CommandEvent
CDSSConcreteElementProxy	CDSSRWControl	Consolidation
CDSSConsolidation	CDSSRWController	ConsolidationElement
CDSSConsolidationElement	CDSSRWControlNode	ConsolidationSorter
CDSSConsolidationElementProxy	CDSSRWControls	ConsolidationStyleLayer
CDSSConsolidationElements	CDSSRWField	ConsolidationUnit
CDSSControlCell	CDSSRWFieldGroupNode	Constants
CDSSCustomGroup	CDSSRWFields	ControlAndElements
CDSSCustomGroupAndDepth	CDSSRWIterator	ControlElement
CDSSCustomNumberFormatter	CDSSRWNode	ControllerEvent
CDSSCustomSubFormat	CDSSRWSectionIterator	ControlNodeDataLoader
CDSSCustomSubFormats	CDSSRWSectionNode	ControlNodeModel
CDSSDataElementsIterator	CDSSRWTemplateIterator	ControlPropertiesChangedEvent
CDSSDimension	CDSSRWTemplateNode	ControlStyle
CDSSDocumentDefinition	CDSSSecondClassObject	Coordinates
CDSSDocumentInstance	CDSSSort	CreateResourceBundleTask
CDSSDocumentSubtotal	CDSSSorts	Cube
CDSSDocumentSubtotalRefs	CDSSSubexpressionElementProxy	CubeRotate
CDSSDocumentSubtotals	CDSSSubtotalElementProxy	CubeRotateInstance
CDSSFieldGroupCell	CDSSTemplate	CurrencyPosition
CDSSFieldGroupIterator	CDSSTemplateAttribute	CurrencySymbol

CustomGroup	EnumApplicationType	EnumDSSSubtotalsPosition
CustomGroupElement	EnumBinaryContextDetailFlags	EnumDSSSubTypes
CustomGroupSorter	EnumBinaryContextFlags	EnumDSTemplateSubtotalType
CustomGroupStyleLayer	EnumControllerCommandTypes	EnumDSTemplateUnitType
CustomGroupUnit	EnumDescriptorNames	EnumDSThresholdScope
CustomToolTipManager	EnumDSSAccessRightFlags	EnumDSSUserFilter
CylinderProps	EnumDSSAutoText	EnumDSSViewMedia
CylinderViewer	EnumDSSAxesBitMap	EnumDSSXMLDocResultFlags
DashboardWidget	EnumDSSAxisName	EnumDSSXMLRWSelectorUnsetStatus
DataCell	EnumDSSBaseFormType	EnumDSSXMLSymbol
DataGridMenuCategories	EnumDSSCellInfoFlags	EnumFlashVars
DataModelError	EnumDSSCurrentElementStatus	EnumFormatPadPosition
DataPointRenderer	EnumDSSCustomGroupDisplay	EnumGraphAxisBooleanProperty
DataProviderEnabledDataPointRenderer	EnumDSSCustomGroupSortType	EnumGraphAxisDoubleProperty
DatasetResolvedMacros	EnumDSSDataType	EnumGraphAxisLabelLocation
DataTypeConverter	EnumDSSDisplayMode	EnumGraphAxisLongProperty
DecimalPlaces	EnumDSSDocumentType	EnumGraphAxisName
DecoratedDispatcher	EnumDSSDrillImportance	EnumGraphAxisScaleOptions
DefaultFormatPropertyDefinition	EnumDSSDrillType	EnumGraphBevelType
DefaultGridStyle	EnumDSSElementType	EnumGraphBooleanProperty
DefaultsVersion	EnumDSSExpressionType	EnumGraphBoxPlotTailShape
DefaultsVersions	EnumDSSFieldGroupDataLevel	EnumGraphBoxStyle
DefaultViewMode	EnumDSSFieldGroupJoin	EnumGraphColorMode
DefnModelError	EnumDSSFormType	EnumGraphDataLineType
DeleteEvent	EnumDSSFunction	EnumGraphDataTextPosition
DerivedElementSorter	EnumDSSGraphHeaderProperties	EnumGraphDataTextPositionX
DerivedElementStyleLayer	EnumDSSGridCellDisplayStyle	EnumGraphDataTextShowType
DimensionHandle	EnumDSSGridCellProperty	EnumGraphDefDataTextPosition
DocBinaryResultsTask	EnumDSSGridCellType	EnumGraphDefFontAlignment
DocLoaderTask	EnumDSSGroupProperty	EnumGraphDefGradientType
DocumentIteratorEvent	EnumDSSMetricFunction	EnumGraphDefLegendPosition
DraggableEvent	EnumDSSMetricType	EnumGraphDefMarkerLayout
DraggableTextField	EnumDSSNodeType	EnumGraphDefMarkerShape
DrillPath	EnumDSSObjectType	EnumGraphDepthMode
DropShadowDepth	EnumDSSOrderParentFirst	EnumGraphDoubleProperty
DropShadowEffect	EnumDSSPrivilegeType	EnumGraphFillEffect
DropZoneEvent	EnumDSSReportFilter	EnumGraphFontAlignment
DropZones	EnumDSSReportType	EnumGraphFontBoxType
DummyUnit	EnumDSSRWControlFormatType	EnumGraphFontRotation
DynamicResourceBundle	EnumDSSRWControlType	EnumGraphFontStyle
EditableViewMode	EnumDSSRWFieldType	EnumGraphGaugeBorderStyle
ElementAndNode	EnumDSSRWIteratorType	EnumGraphGaugeNeedleStyle
ElementsAndNode	EnumDSSRWLinkType	EnumGraphGaugeStyle
EmbedFonts	EnumDSSRWNodFormatType	EnumGraphGradientType
EnableGridPadding	EnumDSSRWNodType	EnumGraphGridLineType
EnableHorizontalSwipeToChange	EnumDSSRWOptions	EnumGraphHiloMinorTypes
EnhancedEntityToolTip	EnumDSSRWPageByStyle	EnumGraphLegendPosition
EnhancedTooltip	EnumDSSRWRetrieveFieldsOption	EnumGraphLineStyle
Entity	EnumDSSRWSectionType	EnumGraphLongProperty
EntityContainer	EnumDSSRWTemplateIteratorType	EnumGraphMajorType
EntityToolTip	EnumDSSRWTemplateOrigin	EnumGraphMarkerLayout
EnumAccordionConstants	EnumDSSRWTreeType	EnumGraphMarkerShape
EnumAEUnsetType	EnumDSS>ShowAxisMinMaxLabels	EnumGraphMetricValueStyle
EnumApplicationSubtype	EnumDSSSortType	EnumGraphNFCurrencyPosition

EnumGraphNFNegativeNumber	EnumMFGraphLegendPosition	FontWeightValues
EnumGraphNumberFormatType	EnumMFGraphLineStyle	ForceNewPage
EnumGraphObjectID	EnumMFGraphMajorType	FormatPropertyDefinition
EnumGraphPictureFlipType	EnumMFGraphMarkerLayout	FormatPropertySetMap
EnumGraphPieMinorTypes	EnumMFGraphMarkerShape	FrequencyVisualization
EnumGraphProperties	EnumMFGraphObjectID	FrequencyVisualizationAreaSeries
EnumGraphRegressionType	EnumMFGraphPieMinorTypes	GenericWebServerTask
EnumGraphRiserEmphasisType	EnumMFGraphPieSubtype	GlassDataGrid
EnumGraphSeriesBooleanProperty	EnumMFGraphRiserEmphasisType	GlassDataGridHeaderSeparator
EnumGraphSeriesLongProperty	EnumMFGraphSeriesTypes	Global
EnumGraphShowAxis	EnumMFGraphShowAxis	GlobalModel
EnumGraphStringProperty	EnumMFGraphSizingPreference	GradientAngle
EnumGraphTitleName	EnumMFGraphTickMarkStyle	GradientBar
EnumGraphType	EnumMFGraphTitleName	GradientColor
EnumHorizontalAreaMinorTypes	EnumMFGraphTrendlineTypes	GradientColorSlider
EnumHorizontalBarMinorTypes	EnumMFGraphVerticalAreaMinorTypes	GradientColorSliderEvent
EnumHorizontalLineMinorTypes	EnumMFGraphVerticalAreaSubtype	GradientColorSliderThumb
EnumInteractiveStackedGraph	EnumMFGraphVerticalBarMinorTypes	GradientColorSliderTrackSkin
EnumLassoConstants	EnumMFGraphVerticalBarSubtype	GradientXOffset
EnumLinkAnswerMode	EnumMFGraphVerticalLineMinorTypes	GradientYOffset
EnumLinkDisplayStyle	EnumMFGraphVerticalLineSubtype	GraphAxes
EnumLinkTargetSubtype	EnumOperatorType	GraphAxis
EnumLinkTargetType	EnumPropertyHelperTranslate	GraphFill
EnumLinkType	EnumResultKeys	GraphFillBevel
EnumManipulationCommandTypes	EnumVerticalAreaMinorTypes	GraphFillGradient
EnumMFGraphAttributeID	EnumVerticalBarMinorTypes	GraphFont
EnumMFGraphAxisName	EnumVerticalLineMinorTypes	GraphFormat
EnumMFGraphAxisScaleOptions	ExecuteLinkDrillCommand	GraphLegend
EnumMFGraphBevelType	ExpandToFit	GraphLine
EnumMFGraphBorderStyles	ExpNode	GraphModelBuilderFactory
EnumMFGraphBoxType	ExpNodeConstant	GraphModelError
EnumMFGraphColorMode	ExpNodeElementsObject	GraphNumberFormat
EnumMFGraphCombinationMinorTypes	ExpNodeFormShortcut	GraphObject
EnumMFGraphCombinationTypes	ExpNodeOperator	GraphObjectConstructionHelper
EnumMFGraphDataLineType	ExpNodeShortcut	GraphObjectConstructionHelperBinary
EnumMFGraphDataTextAlignment	Expression	GraphQualityDPI
EnumMFGraphDataTextPosition	ExtraInfoLoaderTask	GraphSeries
EnumMFGraphDataTextPositionX	ExtraWidth	GraphSeriesCollection
EnumMFGraphDataTextShowType	Field	GraphTitle
EnumMFGraphFillEffect	FieldGroupDataLoader	GraphTitles
EnumMFGraphFontAlignment	FieldGroupModel	GridAreaPercent
EnumMFGraphFontBoxType	FieldGroupView	GridPosition
EnumMFGraphFontRotation	FieldPropertiesChangedEvent	GridPositions
EnumMFGraphFontSize	FieldSet	GridStyleMap
EnumMFGraphGradientType	FieldSetBorder	GridTextAlign
EnumMFGraphGridLineType	FillStyle	GridWidth
EnumMFGraphHiloMinorTypes	Filter	GroupbyDataLoader
EnumMFGraphHiloSubtype	FontFamily	GroupbyElement
EnumMFGraphHorizontalAreaMinorTypes	FontScript	GroupbyHeader
EnumMFGraphHorizontalAreaSubtype	FontSize	GroupbyModel
EnumMFGraphHorizontalBarMinorTypes	FontStrikeout	GroupbyPath
EnumMFGraphHorizontalBarSubtype	FontStyle	GroupByUnitAndNode
EnumMFGraphHorizontalLineMinorTypes	FontStyleValues	Hash
EnumMFGraphHorizontalLineSubtype	FontWeight	HasRoundedCorners

HeaderWidth	IDSSElements	IGraphLine
HeatMapLegend	IDSSFieldGroupIterator	IGraphModelBuilder
HeatMapMenuController	IDSSFormat	IGraphObject
HeatMapMenuDropDown	IDSSGraphCell	IGraphSeries
HeatMapTaskPane	IDSSGraphIterator	IGraphSeriesCollection
HeatMapView	IDSSGridIterator	IGraphTitle
Height	IDSSGroupByIterator	IGraphTitles
HeightMode	IDSSGroupByUnit	IHeatMapContextOptions
HeightModes	IDSSGroupByUnits	IndexFilteredArrayConsumer
Helper	IDSSMetric	ItemRendererEx
HiddenLinearAxis	IDSSMetrics	ImageComboBoxEx
HiddenTickGraph	IDSSObjectInfo	ImageListItemRenderer
HideIfEmpty	IDSSProperty	IManipulationAddAttributeFormToTemplateCommand
HInsideColor	IDSSPropertySet	IManipulationAddAttributeToTemplateCommand
HInsideStyle	IDSSReportDefinition	IManipulationAddMetricToTemplateCommand
HTMLText	IDSSReportInstance	IManipulationAddTemplateCommand
HtmlToolTip	IDSSRWControl	IManipulationAddUnitToTemplateCommand
HTMLType	IDSSRWControlIterator	IManipulationCommand
HTMLTypes	IDSSRWControlNode	IManipulationDrill
HTTPServiceTask	IDSSRWControls	IManipulationMoveMetricCommand
Hyperlink	IDSSRWField	IManipulationPivotAllCommand
HyperlinkHelper	IDSSRWFieldGroupNode	IManipulationPivotUnitCommand
Hyperlinks	IDSSRWFields	IManipulationRemoveUnitCommand
IApController	IDSSRWIterator	IManipulationSetGroupByElementCommand
IAttributeElementID	IDSSRWNode	IManipulationSetLayoutCommand
IBaseCollection	IDSSRWSectionIterator	IManipulationSetSelectionFromControlCommand
ICellFmtData	IDSSRWSectionNode	IManipulationSetSelectionFromTemplateCommand
IChartElementEx	IDSSRWTemplaterator	IManipulationSetTemplateDisplayModeCommand
IChartElementExHoriz	IDSSRWTemplatenode	IManipulationSetTemplateWindowStateCommand
IChartElementExSeriesByColumn	IDSSSort	IManipulationSortTemplateCommand
ICommand	IDSSSorts	IncludeBookmarks
ICommandDrillWithXTabSelectionElements	IDSSTemplate	Indented
ICommandFilterXTabSelection	IDSSTemplateAttribute	IndexFilteredArrayCollection
ICommandReplaceWithXTabSelectionData	IDSSTemplateConsolidation	InteractiveStackedGraph
IDAndAlias	IDSSTemplateCustomGroup	IPathTransformer
IDataModel	IDSSTemplateDimension	IRecordableCommand
IDataRendererEx	IDSSTemplateMetric	IRepeatedTask
IdentityPathTransformer	IDSSTemplateMetrics	IResizable
IDockable	IDSSTemplateSubtotal	IResizeHandle
IDocker	IDSSTemplateSubtotals	IResourceBundle2
IDSSAggMetric	IDSSTemplateUnit	IsChildrenHorizontal
IDSSAttribute	IDSSThreshold	IsDataHorizontal
IDSSAttributeForm	IDSSThresholds	ISeriesEx
IDSSAttributeForms	IElementID	ISeriesExProperties
IDSSAxis	IEntity	ISetRWDPropertiesCommand
IDSSBaseAttributeForm	IExecuteLinkDrillCommand	ISetTemplatePropertyCommand
IDSSConsolidation	IFieldGroupView	IsHyperlink
IDSSConsolidationElement	IGraphAxes	ITask
IDSSConsolidationElements	IGraphAxis	ITaskProgressBar
IDSSControlCell	IGraphFill	ITemplateStyleLayer
IDSCustomGroup	IGraphFillBevel	IteratorError
IDSSDocumentDefinition	IGraphFillGradient	IThresholdDataCell
IDSSDocumentInstance	IGraphFont	IUMActionObject
IDSElement	IGraphLegend	IUPersistable

IViewController	MergeSorter	MFScatterGraph
IViewer	Metric	MFScatterSeries
IViewerContainer	MetricDefaultSorter	MFSeriesCollection
IViewerMap	MetricSorter	MFSolidFill
IWidgetMenuController	MetricUnit	MFText
IXTabHeader	MetricUnitSorter	MFTextureFill
IXTabHeaderElement	MFAbstractBorder	MFTitle
IXTabView	MFAbstractFill	MFToolTip
JSHelper	MFAbstractGridLine	MFTrendline
KeepGroupTogether	MFAbstractSeries	MFVerticalAreaGraph
KeepSectionTogether	MFAreaGraph	MFVerticalAreaSeries
LabelEvent	MFAreaSeries	MFVerticalBarGraph
LabelGridItemRenderer	MFAxis	MFVerticalBarSeries
LabelProps	MFAxisCollection	MFVerticalLineGraph
Lasso	MFAxisLabel	MFVerticalLineSeries
LassoEvent	MFBackgroundFormat	MinMaxNumberValidator
LassoResizeHandle	MFBarGraph	ModelEvent
LassoUtil	MFBarSeries	ModelStyleDefinition
LayoutAnimator	MFBevel	MoveAndResizeEntity
LayoutTarget	MFBubbleGraph	MstrChartLabel
Left	MFBubbleSeries	MstrColorPicker
LegendMarker	MFCategoryAxis	MstrComboBoxItemRenderer
LinearTickProps	MFCustomLine	MstrDataTip
LinearTicks	MFDATAlabel	MstrLinearAxis
LineColor	MFDepth	MstrLinkButton
LineColor2	MFErrorBar	MstrMultilineRadioButton
LineOrientation	MFFont	MstrNumberValidator
LineSpacing	MFGradientFill	MstrPreloaderProgressBar
LineStyle	MFGraph	MstrSliderDataTip
LineWeight	MFGridLines	MstrTooltipBorderSkin
Link	MFHLoGraph	MstrToolTipMenu
LinkAnswer	MFHLoSeries	MstrUITextField
LinkExecuter	MFHorizontalAreaGraph	MultiHash
Links	MFHorizontalAreaSeries	MultiHashNode
LinksXMLConstants	MFHorizontalBarGraph	MultiSliderAlignment
LoadFontSWFsTask	MFHorizontalBarSeries	Name
LockAspectRatio	MFHorizontalLineGraph	NegativeNumbers
LockColumnHeaders	MFHorizontalLineSeries	NestedLabel
LockRowHeaders	MFLLabelFormat	NestedLabelBracket
LoginTask	MFLegend	NewRowOrColumn
LogoutTask	MFLine	NoCommandViewController
LongNames	MFLineGraph	NodeAndControls
MarginBottom	MFLineSeries	NodePropertiesChangedEvent
MarginLeft	MFMajorGridLine	NoMetricsStyle
MarginRight	MFMarker	NumberCategory
MarginTop	MFMinorGridLine	NumberFormat
MaskedAreaRenderer	MFNumericAxis	Object2XML
Matrix3D	MFPatternFill	ObjectInfoData
MaxHeight	MFPictureFill	ObjectMap
MaxPieLabelRadius	MFPieGraph	OldHeight
MaxWidth	MFPieSeries	OldWidth
Mercury	MFPieSeriesCollection	OpenInNewWindow
MergeCells	MFRiser	OverflowMode
MergeColumnCells	MFScaleBreak	PaddingBottom

PaddingLeft	SelectorInclude	TextResizeInstance
PaddingRight	SelectorItemSpacing	ThermometerProps
PaddingTop	SelectorItemWidthMode	ThermometerViewer
PageBreakBetweenGroups	SelectorMetricConditionType	ThousandsSeparator
PageHeight	SelectorMouseOverAction	ThresholdDataCell
PageOrientation	SelectorMultiselect	ThresholdQuickSymbols
PageScaling	SelectorOrientation	ThresholdStyleLayer
PageWidth	SelectorSelection	TimerUtils
PanelStyle	SelectorStyle	TimeSeriesSlider
PaperType	SelectorTitleHeight	TimeSeriesSliderGraphStyleHelper
ParentCheckAxisRenderer	ShowBookmarks	TitleBarDisplay
PatternColor	ShowColHeaders	TitleBarDisplayModes
PatternStyle	ShowControlTitleBar	TitleOverlapViewMode
PictureAlignment	ShowFloatingToolbars	TitleStyle
PictureMode	ShowGaugeLabel	ToolTipHelper
Prefetch	ShowInHTMLViewMode	TooltipText
PrintFooterOn	ShowRowHeaders	Top
PrintHeaderOn	ShowTitleBar	TopLeftCornerRadius
Projection3D	ShowWidgetTitleBar	TopRightCornerRadius
PropertiesHelper	SinglePointReadyLineSeries	TransitionDuration
Quaternion	SingleSliderAlignment	TransitionID
RecorderViewer	Slice	TransitionIDs
RecorderWindow	SlideOutTray	TransitionUse
RecordInfo	SlideOutTrayEvent	Triangle
RemoveFieldEvent	SliderFitToContents	TypeConverter
RemoveNodeEvent	SortConverter	UMActionObject
RepeatHorizontally	SortDefinition	UMPersistable
RepeatSection	Sorter	Unit
ResizeHandle	SortInfo	UnitConditionEvent
ResourceLog	StackedAreaRenderer	UpdateManager
ResourceLogEntry	StateForcingAreaChart	URL
ResourceLogViewer	StyleName	URLLoaderTask
RestartPageNumbering	Subtotal	UseAsInformationWindow
RotationConstants	SuppressLastLevel	UseGaugeDefaultSeriesColors
RowHeight	SWFLoaderTask	UseMasterPageHeaderFooter
RowScenario	TaskEvent	UsePageWidthAsLayoutWidth
RowScenarios	TaskList	Vector2D
ScaleBreakStyle	TaskProgressBar	Vector3D
ScaleBreakType	TemplateBanding	ViewMap
ScalePageHeaderFooter	TemplateShowColHeaders	ViewModelInteractivityLevel
ScalePagesTall	TemplateShowRowHeaders	ViewModelInteractivityLevels
ScalePagesWide	TemplateStyle	ViewPortAreaChart
ScalePercentage	TemplateStyleLayer	ViewPortAreaChartEvent
SecondaryDataProviders	TemplateStyleMap	ViewTitle
SectionCaption	TextAlign	VInsideColor
SectionExpanded	TextAlignValues	VInsideStyle
SectionHorizontalOverflow	TextDecoration	WaitingEvent
SectionModel	TextDecorationValues	WebServerTask
SegmentAreaRenderer	TextFieldGridItemRenderer	Widget
SelectEvent	TextMixer	WidgetAboutWindow
SelectionColor	TextMixerEvent	WidgetClassName
SelectionEvent	TextOverflow	WidgetDragEvent
SelectionFillStyle	TextResize	WidgetDragUtil
SelectionFillStyles	TextResizeEvent	WidgetEnable

WidgetGraphStyleHelper	ConfigOptions.CSVAutoFormatNumber	ConfigOptions.ExcelFlashSizeLimitMinimum
WidgetGridStyleHelper	ConfigOptions.CSVAutoFormatBorder	ConfigOptions.
WidgetIlsApp	ConfigOptions.CSVAutoFormatFont	ExcelFlashSizeLimitMaximum
WidgetLegend	ConfigOptions.CSVAutoFormatPatterns	ConfigOptions.ExcelFormatLocale
WidgetLegendItem	ConfigOptions.CSVAutoFormatAlignment	ConfigOptions.ExcelSheetNaming
WidgetLegendUtil	ConfigOptions.CSVAutoFormatWidthHeight	ConfigOptions.ExplicitLocale
WidgetProperties	ConfigOptions.CSVFormatOnRefresh	ConfigOptions.ExplicitExcelFormatLocale
WidgetPropertiesEditor	ConfigOptions.CSVFlatten	ConfigOptions.FormatResults
WidgetPropertiesEditorEvent	ConfigOptions.CSVRemoveExtraColumn	ConfigOptions.FormatOnRefresh
WidgetStringUtils	ConfigOptions.EnableLogging	ConfigOptions.FlashControlPath
WidgetStyleHelper	ConfigOptions.EventLateBinding	ConfigOptions.FlashImagePath
WidgetSWFPath	ConfigOptions.ExcelFetchInitialRowSize	ConfigOptions.FlashDataPath
WidgetTitlesUtil	ConfigOptions.ExcelFetchInitialRowSizeMinimum	ConfigOptions.SaveFlashData
WidgetVersion	ConfigOptions.ExcelFetchInitialRowSizeMaximum	ConfigOptions.FlashTrustFlashPath
Width	ConfigOptions.ExcelChartFixedSizeUnit	ConfigOptions.FlashTrustImagePath
WidthMode	ConfigOptions.ExcelChartHeight	ConfigOptions.FlashTrustDataPath
WidthModes	ConfigOptions.ExcelChartHeightRatio	ConfigOptions.FullWebServicesErrors
WindowState	ConfigOptions.ExcelChartLockAspectRatio	ConfigOptions.GetQueryString
WindowStates	ConfigOptions.ExcelChartRefreshUsingPersistedSize	ConfigOptions.GridFormatNumber
WordWrap	ConfigOptions.ExcelChartScaleOption	ConfigOptions.GridFormatBorder
XMLConstants	ConfigOptions.ExcelChartWidth	ConfigOptions.GridFormatFont
XmlName	ConfigOptions.ExcelChartWidthRatio	ConfigOptions.GridFormatPatterns
XTabDataLoader	ConfigOptions.ExcelPlacementFirstResult	ConfigOptions.GridFormatAlignment
XTabHeader	ConfigOptions.ExcelPlacementSubsequentResults	ConfigOptions.GridLayoutDefinition
XTabHeaderElement	ConfigOptions.ExcelPlacementPageByResult	ConfigOptions.GridOnly_ShowDataOnly
XTabHeaderItem	ConfigOptions.ExcelPlacementVerticalLayout	ConfigOptions.HistoryListUnreadOnly
XTabHeaderLayout	ConfigOptions.ExcelPlacementVerticalLayoutMaximum	ConfigOptions.IncrementalFetchMode
XTabIndex	ConfigOptions.ExcelPlacementVerticalLayoutMinimum	ConfigOptions.
XTabIndexPool	ConfigOptions.ExcelPlacementSubsequentBelowRows	IncrementalFetchRefreshMode
XTabModel	ConfigOptions.ExcelPlacementSubsequent-	ConfigOptions.KeepPlaceHolder
XTabMultiView	BelowRowsMinimum	ConfigOptions.ListRequiredPromptsFirst
XTabSelection	ConfigOptions.	ConfigOptions.LoadInExcel
XTabSelectionData	ExcelPlacementSubsequentBelowRowsMaximum	ConfigOptions.LoadInPowerPoint
XTabSelectionElements	ConfigOptions.ExcelPlacementSubsequent-	ConfigOptions.LoadInWord
XTabSelectionUnit	RightColumns	ConfigOptions.Locale
XTabView	ConfigOptions.	ConfigOptions.LockPersistence
XTabViewHeader	ExcelPlacementSubsequentRightColumnsMinimum	ConfigOptions.LogFileDirectory
ZIndex	ConfigOptions.	ConfigOptions.LogToScreen
ZoomEven	ExcelPlacementSubsequentRightColumnsMaximum	ConfigOptions.
	ConfigOptions.ExcelResultRowLimit	MaximumConcurrentReports
	ConfigOptions.ExcelResultRowLimitMinimum	ConfigOptions.
	ConfigOptions.ExcelResultRowLimitMaximum	MaximumConcurrentReportsMinimum
	ConfigOptions.ExcelResultColumnLimit	ConfigOptions.
	ConfigOptions.ExcelResultColumnLimitMinimum	MaximumConcurrentReportsMaximum
	ConfigOptions.ExcelResultColumnLimitMaximum	ConfigOptions.MessageDisplayNoData
	ConfigOptions.ExcelResultPageLimit	ConfigOptions.
	ConfigOptions.ExcelResultPageLimitMinimum	MessagePersistPartialReport
	ConfigOptions.ExcelResultPageLimitMaximum	ConfigOptions.
	ConfigOptions.ExcelFetchCellSize	MessageDisplayPageOverLimit
	ConfigOptions.ExcelFetchCellSizeMinimum	ConfigOptions.MessageDisplayResult
	ConfigOptions.ExcelFetchCellSizeMaximum	OverLimit
	ConfigOptions.ExcelFlashDisplay	ConfigOptions.
	ConfigOptions.FlashOtherFilesToRetrieve	MessageRecreateMissingWorksheet
	ConfigOptions.ExcelFlashHeight	ConfigOptions.MessageCancelAllPrompt
	ConfigOptions.ExcelFlashWidth	ConfigOptions.MessageContinueDelete
		ConfigOptions.

MicroStrategy Office API

Interfaces

ConfigOptions.AllowBlankPassword	ConfigOptions.ExcelResultColumnLimitMinimum	ConfigOptions.
ConfigOptions.ApplyColumnWidths	ConfigOptions.ExcelResultColumnLimitMaximum	MaximumConcurrentReportsMaximum
ConfigOptions.ApplyDateTimeFormatting	ConfigOptions.ExcelResultPageLimit	ConfigOptions.MessageDisplayNoData
ConfigOptions.ApplyRowHeights	ConfigOptions.ExcelResultPageLimitMinimum	ConfigOptions.
ConfigOptions.AutoFitColumnsGrid	ConfigOptions.ExcelResultPageLimitMaximum	MessagePersistPartialReport
ConfigOptions.AutoFitRowsGrid	ConfigOptions.ExcelFetchCellSize	ConfigOptions.
ConfigOptions.AutoFilters	ConfigOptions.ExcelFetchCellSizeMinimum	MessageDisplayPageOverLimit
ConfigOptions.AutoGroup	ConfigOptions.ExcelFetchCellSizeMaximum	ConfigOptions.MessageDisplayResult
ConfigOptions.AutoHistoryList	ConfigOptions.ExcelFlashDisplay	OverLimit
ConfigOptions.AutoNTAuthentication	ConfigOptions.FlashOtherFilesToRetrieve	ConfigOptions.
ConfigOptions.ApplyBorder	ConfigOptions.ExcelFlashHeight	MessageRecreateMissingWorksheet
ConfigOptions.CSVAutoFormat	ConfigOptions.ExcelFlashWidth	ConfigOptions.MessageCancelAllPrompt
		ConfigOptions.MessageContinueDelete
		ConfigOptions.

MessageWarningMissingFlashFiles	ConfigOptions.PowerPointChartNotExceedSlide	ConfigOptions.
ConfigOptions.	ConfigOptions.	PowerPointPositionRefreshUsingPersistedLocation
MessageCloseDocumentForPackageUtility	PowerPointChartNotExceedSlidePercent	ConfigOptions.PowerPointPositionCenterOnSlide
ConfigOptions.MessageConfirmDeletion	ConfigOptions.	ConfigOptions.PresentationType
ConfigOptions.OfficeChartAutoScale	PowerPointChartRefreshUsingPersistedSize	ConfigOptions.ProductName
ConfigOptions.	ConfigOptions.PowerPointNonChartHeight	ConfigOptions.PromptElementFetchSize
OfficeChartAlwaysUseDefaultType	ConfigOptions.PowerPointNonChartWidth	ConfigOptions.PromptElementFetchSizeMinimum
ConfigOptions.OfficeChartCreate2D	ConfigOptions.PowerPointNonChartFixedSizeUnit	ConfigOptions.PromptElementFetchSizeMaximum
ConfigOptions.OfficeChartFootnoteInTitle	ConfigOptions.PowerPointNonChartScaleOption	ConfigOptions.PromptRenameMessageAllowed
ConfigOptions.OfficeChartImagesType	ConfigOptions.PowerPointNonChartWidthRatio	ConfigOptions.PromptReuseAnswers
ConfigOptions.OfficeChartImages	ConfigOptions.PowerPointNonChartHeightRatio	ConfigOptions.PromptShareAnswers
ConfigOptions.	ConfigOptions.	ConfigOptions.PromptUseDefault
OfficeChartShowDataTable	PowerPointNonChartLockAspectRatio	ConfigOptions.ProxyAddress
ConfigOptions.	ConfigOptions.PowerPointNonChartNotExceedSlide	ConfigOptions.ProxyBypassLocal
OfficeChartShowLegendKey	ConfigOptions.	ConfigOptions.ProxyPort
ConfigOptions.OfficeChartWidth	PowerPointNonChartNotExceedSlidePercent	ConfigOptions.ProxyUseDefault
ConfigOptions.	ConfigOptions.	ConfigOptions.ProxyUseProxy
OfficeChartWidthMinimum	PowerPointNonChartRefreshUsingPersistedSize	ConfigOptions.QuietMode
ConfigOptions.	ConfigOptions.PowerPointPlacementResult	ConfigOptions.ReconnectSession
OfficeChartWidthMaximum	ConfigOptions.PowerPointMultiplePlacementResult	ConfigOptions.RefreshOnOpen
ConfigOptions.OfficeChartHeight	ConfigOptions.PowerPointResultRowLimit	ConfigOptions.RefreshOnOpenPrompt
ConfigOptions.	ConfigOptions.PowerPointResultRowLimitMinimum	ConfigOptions.ReportDetailsDisplay
OfficeChartHeightMinimum	ConfigOptions.	ConfigOptions.ReportDetailsComments
ConfigOptions.	PowerPointResultRowLimitMaximum	ConfigOptions.RSUseOOXML
OfficeChartHeightMaximum	ConfigOptions.PowerPointResultColumnLimit	ConfigOptions.SaveReportXML
ConfigOptions.	ConfigOptions.	ConfigOptions.SaveReportXMLPath
OfficeChartRefreshFormatting	PowerPointResultColumnLimitMinimum	ConfigOptions.ScaleRatioLimitMinimum
ConfigOptions.OfficeCharType	ConfigOptions.	ConfigOptions.ScaleRatioLimitMaximum
ConfigOptions.PageHeaderDisplay	PowerPointResultColumnLimitMaximum	ConfigOptions.ScaleInchLimitMinimum
ConfigOptions.	ConfigOptions.PowerPointResultPageLimit	ConfigOptions.ScaleInchLimitMaximum
PageHeaderCompactDisplay	ConfigOptions.	ConfigOptions.ScaleCentimeterLimitMinimum
ConfigOptions.PageSelection	PowerPointResultPageLimitMinimum	ConfigOptions.ScaleCentimeterLimitMaximum
ConfigOptions.PageRefresh	ConfigOptions.	ConfigOptions.ScalePointLimitMinimum
ConfigOptions.RSLayoutSelection	PowerPointResultPageLimitMaximum	ConfigOptions.ScalePointLimitMaximum
ConfigOptions.RSLayoutRefresh	ConfigOptions.PowerPointDisplayRowLimit	ConfigOptions.ScalePixelLimitMinimum
ConfigOptions.RSGrouping	ConfigOptions.	ConfigOptions.ScalePixelLimitMaximum
ConfigOptions.RSChartDisplay	PowerPointDisplayRowLimitMinimum	ConfigOptions.ScaleCellLimitMinimum
ConfigOptions.RSRefreshGroupBy	ConfigOptions.	ConfigOptions.ScaleCellLimitMaximum
ConfigOptions.PersistLayout	PowerPointDisplayRowLimitMaximum	ConfigOptions.SearchObjectsLimit
ConfigOptions.PersistLoginID	ConfigOptions.PowerPointDisplayColumnLimit	ConfigOptions.SearchObjectsLimitMinimum
ConfigOptions.PersistSmartTagLoginID	ConfigOptions.	ConfigOptions.SearchObjectsLimitMaximum
ConfigOptions.PivotTableAutoFormat	PowerPointDisplayColumnLimitMinimum	ConfigOptions.SearchObjectsFetchSize
ConfigOptions.PivotTableLayout	ConfigOptions.	ConfigOptions.SearchObjectsFetchSizeMinimum
ConfigOptions.	PowerPointDisplayColumnLimitMaximum	ConfigOptions.SearchObjectsFetchSizeMaximum
PivotTableShowItemWithNoData	ConfigOptions.PowerPointFlashHeight	ConfigOptions.ServerTimeZone
ConfigOptions.PowerPointChartHeight	ConfigOptions.PowerPointFlashWidth	ConfigOptions.ShowAttributeFormHeaders
ConfigOptions.PowerPointChartWidth	ConfigOptions.PowerPointFlashSizeLimitMinimum	ConfigOptions.ShowHeaderDescriptions
ConfigOptions.	ConfigOptions.PowerPointFlashSizeLimitMaximum	ConfigOptions.ShowLoadedProjectsBeforeLogin
PowerPointChartFixedSizeUnit	ConfigOptions.PowerPointGridPasteType	ConfigOptions.StyleSharingMode
ConfigOptions.	ConfigOptions.PowerPointPositionHorizontalFrom	ConfigOptions.StyleSharingGroupName
PowerPointChartScaleOption	ConfigOptions.PowerPointPositionHorizontalOffset	ConfigOptions.TrimWhitespace
ConfigOptions.	ConfigOptions.	ConfigOptions.UseProjectNameToConnect
PowerPointChartWidthRatio	PowerPointPositionHorizontalOffsetSizeUnit	ConfigOptions.
ConfigOptions.	ConfigOptions.PowerPointPositionVerticalFrom	UseProjectSourceNameToConnect
PowerPointChartHeightRatio	ConfigOptions.PowerPointPositionVerticalOffset	
ConfigOptions.	ConfigOptions.	
PowerPointChartLockAspectRatio	PowerPointPositionVerticalOffsetSizeUnit	

ConfigOptions.UseStyles	ConfigOptions.WordChartScaleOption	DisplaySettings.ReportDetailsOn
ConfigOptions.WebOutlineStyle	ConfigOptions.WordChartWidthRatio	DisplaySettings.TrimWhitespace
ConfigOptions.WebPromptStyle	ConfigOptions.WordChartHeightRatio	Document.Access
ConfigOptions.	ConfigOptions.	Document.CreationTime
WebServicesExecuteTimeout	WordChartLockAspectRatio	Document.Description
ConfigOptions.	ConfigOptions.WordChartNotExceedPage	Document.
WebServicesExecuteTimeoutMinimum	ConfigOptions.	ExecuteAndDisplayDefaultInExcel
ConfigOptions.	WordChartNotExceedPagePercent	Document.
WebServicesExecuteTimeoutMaximum	ConfigOptions.	ExecuteAndDisplayDefaultInPowerPoint
ConfigOptions.	WordChartRefreshUsingPersistedSize	Document.
WebServicesBrowseTimeout	ConfigOptions.WordNonChartHeight	ExecuteAndDisplayDefaultInWord
ConfigOptions.	ConfigOptions.WordNonChartWidth	Document.ExecuteDocument
WebServicesBrowseTimeoutMinimum	ConfigOptions.	Document.FolderID
ConfigOptions.	WordNonChartFixedSizeUnit	Document.Initialize
WebServicesBrowseTimeoutMaximum	ConfigOptions.	Document.isHidden
ConfigOptions.	WordNonChartScaleOption	Document.ModificationTime
WebServicesResultPollingTimer	ConfigOptions.WordNonChartWidthRatio	Document.ObjectID
ConfigOptions.	ConfigOptions.	Document.ObjectName
WebServicesResultPollingTimerMinimum	WordNonChartHeightRatio	Document.Owner
ConfigOptions.	ConfigOptions.	Document.Path
WebServicesResultPollingTimerMaximum	WordNonChartLockAspectRatio	Document.SubType
ConfigOptions.WebServicesURL	ConfigOptions.	Document.Type
ConfigOptions.WebURL	WordNonChartNotExceedPage	DocumentInstance.answerPrompt
ConfigOptions.WordGridPasteType	ConfigOptions.	DocumentInstance.GetDocumentResults
ConfigOptions.WordResultRowLimit	WordNonChartNotExceedPagePercent	DocumentInstance.GetPrompts
ConfigOptions.	ConfigOptions.	Document.MessageID
WordResultRowLimitMinimum	WordNonChartRefreshUsingPersistedSize	Documents.Add
ConfigOptions.	ConfigOptions.LoadFromFile	Documents.Count
WordResultRowLimitMaximum	ConfigOptions.SaveToFile	Documents.ExecuteAndDisplayObjects
ConfigOptions.WordResultColumnLimit	DisplaySettings.ApplyBorder	Documents.Item
ConfigOptions.	DisplaySettings.ApplyColumnWidths	Documents.Remove
WordResultColumnLimitMinimum	DisplaySettings.ApplyDateFormatting	EnumApplyColumnWidths
ConfigOptions.	DisplaySettings.ApplyMSTRChartFormats	EnumApplyRowHeights
WordResultColumnLimitMaximum	DisplaySettings.ApplyMSTRFormats	EnumChartAutoScale
ConfigOptions.WordResultPageLimit	DisplaySettings.AutoFilter	EnumChartImageType
ConfigOptions.	DisplaySettings.AutoGroup	EnumCustomGridDisplayElement
WordResultPageLimitMinimum	DisplaySettings.ChartAsImage	EnumDataFetch
ConfigOptions.	DisplaySettings.ChatDisplayFootnoteInTitle	EnumDetailsItem
WordResultPageLimitMaximum	DisplaySettings.ChartImageType	EnumDialogQuestionAnswers
ConfigOptions.WordDisplayRow	DisplaySettings.ChartShowDataTable	EnumDimensions
ConfigOptions.WordDisplayRowMinimum	DisplaySettings.ChartShowLegendKey	EnumDisplayStatus
ConfigOptions.	DisplaySettings.FlattenQuickGrid	EnumDynamicDateType
WordDisplayRowMaximum	DisplaySettings.GridAutoFitColumns	EnumExcelSheetNaming
ConfigOptions.WordDisplayColumn	DisplaySettings.GridAutoFitRows	EnumExportingPageOption
ConfigOptions.	DisplaySettings.GridAutoFormat	EnumExpressionType
WordDisplayColumnMinimum	DisplaySettings.GridOutlineWebstyle	EnumFindProjectStatus
ConfigOptions.	DisplaySettings.	EnumFlashDisplay
WordDisplayColumnMaximum	GridShowHeaderDescriptions	EnumFlashRetrieval
ConfigOptions.WordFlashHeight	DisplaySettings.LockPersistence	EnumGridAutoFormat
ConfigOptions.WordFlashWidth	DisplaySettings.OutlineMode	EnumGridFormatComponents
ConfigOptions.	DisplaySettings.PageHeadersCompactOn	EnumGridReportHeaderOptions
WordFlashSizeLimitMinimum	DisplaySettings.PageHeadersOn	EnumMOIAggregationType
ConfigOptions.	DisplaySettings.PageSelection	EnumMOIDimlyUnitType
WordFlashSizeLimitMaximum	DisplaySettings.presentAs	EnumMOIDisplayType
ConfigOptions.WordChartHeight	DisplaySettings.ReportDetailsInComments	EnumMOIErrorStatus
ConfigOptions.WordChartWidth		
ConfigOptions.WordChartFixedSizeUnit		

EnumMOIExecutionType	EnumProcessStatus	HistoryItem.GetHistoryResults_2
EnumMOIFilteringType	EnumPromptAnswerReuse	HistoryItem.HistoryItemID
EnumMOIFunctionType	EnumReferencePosition	HistoryItem.HistoryItemName
EnumMOIOperatorType	EnumResultPlacement	HistoryItem.HistoryItemSubType
EnumMOIPromptOutputLevel	EnumRSChartDisplay	HistoryItem.HistoryItemType
EnumMOIPromptType	EnumRSGroupingOptions	HistoryItem.IsCustomFlashEnabled
EnumMWSAccessRights	EnumRSLayoutOptions	HistoryItem.IsFlashEnabled
EnumMWSAttachmentType	EnumRSLayoutRefreshOptions	HistoryItem.IsVisualizationEnabled
EnumMWSAttributeFormCategories	EnumRSRefreshGroupBy	HistoryItem.LastUpdateTime
EnumMWSAttributeFormsSetting	EnumRWViewModes	HistoryItem.Message
EnumMWSAttributeLockType	EnumScaleDisplayOption	HistoryItem.MessageID
EnumMWSAuthMode	EnumScaleFixedSizeUnit	HistoryItem.MessageStatus
EnumMWSBrowseStatus	EnumShareStyles	HistoryItem.Owner
EnumMWSConnectProperties	EnumSmartTagType	HistoryItem.StartTime
EnumMWSConnectProperty	EnumTitleOptions	HistoryItem.VisualizationList
EnumMWSDataType	EnumXmlDataType	HistoryItem.VisualizationViewMode
EnumMWSElementRestriction	EnumXmlObjectSubtype	HistoryItems.Count
EnumMWSElementSourceFlags	EnumXmlObjectType	HistoryItems.Item
EnumMWSExecutionFlags	Folder.Access	IMOISavedPromptAnswer.AnswerXML
EnumMWSExecutionObjType	Folder.Contents	IMOISavedPromptAnswer.Delete
EnumMWSExportSetting	Folder.CreationTime	IMOISavedPromptAnswer.GUID
EnumMWSFunctionalityProperty	Folder.Description	IMOISavedPromptAnswer.Name
EnumMWSGraphType	Folder.FolderID	IMOISavedPromptAnswer.Rename
EnumMWSHistoryListAction	Folder.FolderName	IMOISavedPromptAnswer.Retrieve
EnumMWSMessageStatus	Folder.GetContents	IMOISavedPromptAnswer.Reuse
EnumMWSRequestStatus	Folder.Initialize	IMOISavedPromptAnswer.Save
EnumMWSInfoProperty	Folder.Initialize_2	IMOISavedPromptAnswer.
EnumMWSObjectType	Folder.isHidden	SaveOnExecution
EnumMWSObjSubType	Folder.ModificationTime	MOIOOffice.Connect
EnumMWSPProjectFolder	Folder.ObjectID	MOIOOffice.DeleteDisplayedElement
EnumMWSPProjectSessionProperty	Folder.ObjectName	MOIOOffice.DisplayReportXMLFileInExcel
EnumMWSPProjectSetting	Folder.Owner	MOIOOffice.
EnumMWSPProjectWellKnownObj	Folder.Path	DisplayReportXMLFileInPowerPoint
EnumMWSProperties	Folder.SubType	MOIOOffice.DisplayReportXMLFileInWord
EnumMWSResultFlags	Folder.Type	MOIOOffice.DisplayReportXMLInExcel
EnumMWSRSExecutionFlags	Folders.Count	MOIOOffice.
EnumMWSRSExportSetting	Folders.Item	DisplayReportXMLInPowerPoint
EnumMWSRSResultFlags	HistoryItem.CancelHistoryItem	MOIOOffice.DisplayReportXMLInWord
EnumMWSRSSimpleResultFlag	HistoryItem.DefaultViewMode	MOIOOffice.FindReportAtCell
EnumMWSRunStatusType	HistoryItem.DefaultVisualization	MOIOOffice.FindReportAtCurrentLocation
EnumMWSSearchFlags	HistoryItem.Description	MOIOOffice.GetConfigOptions
EnumMWSSearchRestriction	HistoryItem.	MOIOOffice.GetErrorMessage
EnumMWSSyncCommunicationSetting	ExecuteAndDisplayDefaultInExcel	MOIOOffice.GetPersistedReportingElements
EnumMWSUserPrivilege	HistoryItem.	MOIOOffice.GetPersistedReportLocations
EnumMWSViewMedia	ExecuteAndDisplayDefaultInPowerPoint	MOIOOffice.GetProjectSources
EnumMWSVisualizationScope	HistoryItem.	MOIOOffice.GetPromptAnswers
EnumNodeType	ExecuteAndDisplayDefaultInWord	MOIOOffice.Initialize
EnumOfficeChartType	HistoryItem.ExecuteAndDisplayInExcel	MOIOOffice.InitializeDocument
EnumOverlapResolution	HistoryItem.	MOIOOfficeOfflineInitialize
EnumPageBy	ExecuteAndDisplayInPowerPoint	MOIOOffice.ReExecuteAllReports
EnumPasteType	HistoryItem.ExecuteAndDisplayInWord	MOIOOffice.RefreshAllReports
EnumPivotTableAutoFormat	HistoryItem.ExecuteReport	MOIOOffice.RefreshDisplayedElement
EnumPivotTableLayout	HistoryItem.FinishTime	MOIOOffice.RePromptAllReports
EnumPresentAs	HistoryItem.Folder	MOIOOffice.Uninitialize
	HistoryItem.GetHistoryResults	ObjectInfo.Access

ObjectInfo.CreationTime	Project.GetHistoryList_2	PromptAnswerGroups.Remove
ObjectInfo.Description	Project.MyReportsFolderID	PromptAnswers.Add
ObjectInfo.	ProjectMyReportsFolderPath	PromptAnswers.Clear
ExecuteAndDisplayDefaultInExcel	Project.ProjectID	PromptAnswers.Count
ObjectInfo.	Project.projectName	PromptAnswers.
ExecuteAndDisplayDefaultInPowerPoint	Project.ReportObjectTemplatesFolderID	GenerateAnsswerXMLString
ObjectInfo.	Project.ReportObjectTemplatesFolderPath	PromptAnswers.Item
ExecuteAndDisplayDefaultInWord	Project.RootFolderID	PromptAnswers.Remove
ObjectInfo.ExecuteAndDisplayInExcel	Project.SearchObjects	PromptAnswerSet.Answers
ObjectInfo.	Project.SharedReportsFolderID	PromptAnswerSet.Operator
ExecuteAndDisplayInPowerPoint	Project.SharedReportsFolderPath	PromptAnswerSets.Add
ObjectInfo.ExecuteAndDisplayInWord	Projects.Count	PromptAnswerSets.Clear
ObjectInfo.ExecuteDocument	Projects.Item	PromptAnswerSets.Count
ObjectInfo.ExecuteReport	ProjectSource.AuthenticationMode	PromptAnswerSets.Insert
ObjectInfo.ExecuteRSDocument	ProjectSource.DisconnectProjects	PromptAnswerSets.Item
ObjectInfo.FolderID	ProjectSource.GetProjects	PromptAnswerSets.Remove
ObjectInfo.isHidden	ProjectSource.GetProjectsWithAuthMode	PromptAnswerSets.Remove_2
ObjectInfo.ModificationTime	ProjectSource.ProjectSourceName	PromptAttrExpression.
ObjectInfo.ObjectID	ProjectSource.serverName	GetAttributeAnswer
ObjectInfo.ObjectName	ProjectSources.Count	PromptAttrExpression.GetAttributeForms
ObjectInfo.Owner	ProjectSources.Item	PromptAttrExpression.
ObjectInfo.Path	Prompt.GetAttrExpressionPrompt	GetAvailableAttributes
ObjectInfo.SubType	Prompt.GetDateTimePrompt	PromptAttrExpression.SetAttributeAnswer
ObjectInfo.Type	Prompt.GetHierarchyPrompt	PromptAttributeAnswer.AnswerValue1
ObjectInfos.Add	Prompt.GetLevelPrompt	PromptAttributeAnswer.AnswerValue2
ObjectInfos.Count	Prompt.GetMetricExpressionPrompt	PromptAttributeAnswer.GetAttribute
ObjectInfos.ExecuteAndDisplayObjects	Prompt.GetObjectPrompt	PromptAttributeAnswer.GetQualifier
ObjectInfos.Item	Prompt.GetPromptElements	PromptAttributeAnswer.
ObjectInfos.Remove	Prompt.GetSAPExpressionPrompt	GetSelectedElements
ObjectInstance.Equals	Prompt.GetSimplePrompt	PromptAttributeAnswer.Operator
ObjectInstance.GetHashCode	Prompt.HasDefaultAnswer	PromptAttributeAnswer.SetAttribute
ObjectInstance.GetType	Prompt.Maximum	PromptAttributeAnswer.SetQualifier
ObjectInstance.ToString	Prompt.Meaning	PromptAttributeAnswer.
ObjectSubTypes.Add	Prompt.Minimum	SetSelectedElements
ObjectSubTypes.Count	Prompt.Name	PromptAttributeAnswers.Add
ObjectSubTypes.Item	Prompt.PromptID	PromptAttributeAnswers.Count
ObjectSubTypes.Remove	Prompt.PromptIndex	PromptAttributeAnswers.Item
ObjectTypes.Add	Prompt.PromptKey	PromptAttributeAnswers.Remove
ObjectTypes.Count	Prompt.PromptType	PromptAttributeInfo.AttributeUID
ObjectTypes.Item	Prompt.Required	PromptAttributeInfo.DisplayName
ObjectTypes.Remove	Prompt.SaveAnswersRestriction	PromptAttributeInfo.
Project.AnswerRestriction	Prompt.SelectPromptAnswer	GetAvailableQualifiers
Project.Connect	Prompt.Title	PromptAttributeInfo.LockLimit
Project.Disconnect	PromptAnswer.PromptDisplayAnswer	PromptAttributeInfo.LockReason
Project.	PromptAnswer.PromptID	PromptAttributeInfo.LockType
DocumentObjectTemplatesFolderID	PromptAnswer.PromptName	PromptAttributeInfo.Type
Project.	PromptAnswer.PromptName	PromptAttributeInfos.Count
DocumentObjectTemplatesFolderPath	PromptAnswer.PromptXML	PromptAttributeInfos.Item
Project.GetAllReports	PromptAnswerGroup.AnswerSets	PromptBaseAnswer.AnswerValue1
Project.GetAttributeElements	PromptAnswerGroup.Operator	PromptBaseAnswer.AnswerValue2
Project.GetFolderContents	PromptAnswerGroups.Add	PromptBaseAnswer.Operator
Project.GetFolderContents_2	PromptAnswerGroups.Clear	PromptBaseAnswers.Add
Project.GetFolderContents_3	PromptAnswerGroups.Count	PromptBaseAnswers.Clear
Project.GetFolderContents_4	PromptAnswerGroups.Item	PromptBaseAnswers.Contains
Project.GetHistoryList		PromptBaseAnswers.Count

PromptBaseAnswers.Insert	PromptHierarchyAnswer.AnswerOperator	PromptObjectInfos.Count
PromptBaseAnswers.Item	PromptHierarchyAnswer.	PromptObjectInfos.Item
PromptBaseAnswers.Remove	GetAttributeAnswer	PromptObjectInfos.Remove
PromptBaseAnswers.Remove_2	PromptHierarchyAnswer.	PromptQualifierInfo.AttributeFormName
PromptDateTime.AdditionalDays	GetMetricAnswer	PromptQualifierInfo.DataType
PromptDateTime.AdditionalHours	PromptHierarchyAnswer.	PromptQualifierInfo.DisplayName
PromptDateTime.AdditionalMinutes	SetAttributeAnswer	PromptQualifierInfo.Initialize
PromptDateTime.AdditionalMonths	PromtpHierarchyAnswer.	PromptQualifierInfo.QualifierID
PromptDateTime.AdditionalSeconds	GetMetricAnswer	PromptQualifierInfo.QualifierUID
PromptDateTime.AdditionalWeeks	PromptHierarchyAnswers.Add	PromptQualifierInfos.Count
PromptDateTime.AdditionalYears	PromptHierarchyAnswers.Clear	PromptQualifierInfos.Item
PromptDateTime.DayOfWeek	PromptHierarchyAnswers.Count	Prompts.Count
PromptDateTime.DynamicDateType	PromptHierarchyAnswers.Insert	Prompts.Item
PromptDateTime.IsDynamicHour	PromptHierarchyAnswers.Item	PromptSAPExpression.AllowedOperator
PromptDateTime.IsDynamicMinute	PromptHierarchyAnswers.Remove	PromptSAPExpression.
PromptDateTime.IsHourPM	PromptLevel.GetAvailableLevelObjects	GetAvailableAttributes
PromptDateTime.IsStaticDate	PromptLevel.GetLevelAnswer	PromptSAPExpression.
PromptDateTime.IsStaticTime	PromptLevel.SetLevelAnswer	GetExcludingAttributeAnswers
PromptDateTime.MonthOfYear	PromptLevelInfo.AggregationType	PromptSAPExpression.
PromptDateTime.ShowTime	PromptLevelInfo.FilteringType	GetIncludingAttributeAnswers
PromptDateTime.SpecificDayOfMonth	PromptLevelInfo.GetPromptObject	PromptSAPExpression.
PromptDateTime.StaticDate	PromptLevelInfo.SetPromptObject	SetExcludingAttributeAnswers
PromptDateTime.StaticHour	PromptLevelInfos.Add	PromptSAPExpression.
PromptDateTime.StaticMinute	PromptLevelInfos.Count	SetIncludingAttributeAnswers
PromptDateTime.WeekNumber	PromptLevelInfos.Item	PromptSimple.Answer
PromptElementInfo.Abbreviation	PromptLevelInfos.Remove	Report.Answer
PromptElementInfo.Description	PromptMetricAnswer.AnswerValue1	Report.CreationTime
PromptElementInfo.DisplayName	PromptMetricAnswer.AnswerValue2	Report.DefaultVisualization
PromptElementInfo.ElementID	PromptMetricAnswer.attributes	Report.Description
PromptElementInfo.ElementNameCount	PromptMetricAnswer.Function	Report.ExecuteAndDisplayDefaultInExcel
PromptElementInfo.GetAttribute	PromptMetricAnswer.Metric	Report.
PromptElementInfo.GetElementNames	PromptMetricAnswer.Operator	ExecuteAndDisplayDefaultInPowerPoint
PromptElementInfo.ObjectUID	PromptMetricAnswer.OutputLevel	Report.ExecuteAndDisplayDefaultInWord
PromptElementInfo.State	PromptMetricExpression.	Report.ExecuteAndDisplayInExcel
PromptElementInfo.SubType	GetAvailableMetrics	Report.ExecuteAndDisplayInPowerPoint
PromptElementInfo.Type	PromptMetricExpression.	Report.ExecuteAndDisplayInWord
PromptElementInfos.Add	GetMetricAnswer	Report.ExecuteReport
PromptElementInfos.Count	PromptMetricExpression.SetMetricAnswer	Report.FolderID
PromptElementInfos.Item	PromptObject.GetAnswerObjects	Report.Initialize
PromptElementInfos.Remove	PromptObject.GetAttribute	Report.IsCustomFlashEnabled
PromptHierarchy.AnswerGroup	PromptObject.GetAttributeElements	Report.ModificationTime
PromptHierarchy.DrillElementsToAttribute	PromptObject.GetAvailableObjects	Report.ObjectID
PromptHierarchy.DrillElementToAttribute	PromptObject.IsObjectPrompt	Report.ObjectName
PromptHierarchy.GetAllFolderContents	PromptObject.IsPredefinedList	Report.Owner
PromptHierarchy.GetAttributeChildren	PromptObject.Restrictions	Report.Path
PromptHierarchy.GetAttributeElements	PromptObject.SetAnswer	Report.SubType
PromptHierarchy.GetAttributeParents	PromptObjectInfo.Abbreviation	Report.Type
PromptHierarchy.GetAvailableAttributes	PromptObjectInfo.Description	Report.VisualizationList
PromptHierarchy.GetAvailableHierarchies	PromptObjectInfo.DisplayName	Report.VisualizationViewMode
PromptHierarchy.GetFolderContents	PromptObjectInfo.ObjectID	ReportingElement.answerPrompt
PromptHierarchy.GetFolderContents_2	PromptObjectInfo.ObjectType	ReportingElement.Cancel
PromptHierarchy.GetHierarchyAnswers	PromptObjectInfo.ObjectUID	ReportingElement.Delete
PromptHierarchy.GetHierarchyAttributes	PromptObjectInfo.State	ReportingElement.DisplayDefaultInExcel
PromptHierarchy.SetAnswer	PromptObjectInfo.Subtype	ReportingElement.GetDisplaySettings
	PromptObjectInfos.Add	ReportingElement.

GetExcelReportProperties	ReportLocation.GetPDFResults	ReportProperties.ChartShowLegendKey
ReportingElement.GetPDFResults	ReportLocation.	ReportProperties.Columns
ReportingElement.	GetPowerPointReportProperties	ReportProperties.CSVAutoFormat
GetPowerPointReportProperties	ReportLocation.GetPrompts	ReportProperties.CSVFlatten
ReportingElement.	ReportLocation.GetReportProperties	ReportProperties.DisplayColumns
GetPromptAnswerXMLString	ReportLocation.GetReportResults	ReportProperties.DisplayRows
ReportingElement.GetPrompts	ReportLocation.GetWordReportProperties	ReportProperties.FlashAssetMapFileName
ReportingElement.GetReportLocation	ReportLocation.MessageID	ReportProperties.FlashControlPath
ReportingElement.GetReportProperties	ReportLocation.Name	ReportProperties.FlashDataFileName
ReportingElement.GetReportResults	ReportLocation.ObjectID	ReportProperties.FlashDataPath
ReportingElement.GetRSLocation	ReportLocation.PDFFileName	ReportProperties.FlashDefinitionFileName
ReportingElement.	ReportLocation.PersistReport	ReportProperties.FlashDefinitionPath
GetWordReportProperties	ReportLocation.QueryString	ReportProperties.
ReportingElement.MessageID	ReportLocation.RefreshDisplay	FlashLocalizationFileName
ReportingElement.Name	ReportLocation.RefreshReport	ReportProperties.FlashSWFFileName
ReportingElement.ObjectID	ReportLocation.RefreshReport_2	ReportProperties.FlashVars
ReportingElement.PageCount	ReportLocation.	ReportProperties.
ReportingElement.PDFFileName	RefreshReportDisplayPrompts	GridReportHeaderOption
ReportingElement.Persist	ReportLocation.ReportLocationID	ReportProperties.IsFlashModeAllowed
ReportingElement.QueryString	ReportLocation.ReportCSV	ReportProperties.LastUpdated
ReportingElement.RefreshDisplay	ReportLocation.ResultHTML	ReportProperties.LockPersistence
ReportingElement.RefreshReport	ReportLocation.ResultXml	ReportProperties.loginID
ReportingElement.	ReportLocation.	ReportProperties.OutlineMode
RefreshReportDisplayPrompts	RetrieveVisualizationResults	ReportProperties.
ReportingElement.	ReportLocation.SaveExcelReportProperties	PageHeadersCompactOn
SaveExcelReportProperties	ReportLocation.	ReportProperties.PageHeadersOn
ReportingElement.	SavePowerPointReportProperties	ReportProperties.PageName
SavePowerPointReportProperties	ReportLocation.SaveReportProperties	ReportProperties.PageSelection
ReportingElement.SaveReportProperties	ReportLocation.	ReportProperties.presentAs
ReportingElement.	SaveWordReportProperties	ReportProperties.Project
SaveWordReportProperties	ReportLocation.TotalColumns	ReportProperties.ProjectSource
ReportingElements.Add	ReportLocation.TotalRows	ReportProperties.Rank
ReportingElements.Count	ReportLocations.Add	ReportProperties.ReportDisplayStatus
ReportingElements.InList	ReportLocations.Count	ReportProperties.reportID
ReportingElements.Item	ReportLocations.Item	ReportProperties.reportName
ReportingElements.Item_2	ReportLocations.Remove	ReportProperties.ReportPath
ReportingElements.Remove	ReportProperties.ApplyBorder	ReportProperties.Rows
ReportLocation.answerPrompt	ReportProperties.ApplyColumnWidths	ReportProperties.
ReportLocation.Cancel	ReportProperties.	RSDocumentChartDisplay
ReportLocation.CancelReport	ApplyDateTimeFormatting	ReportProperties.
ReportLocation.Delete	ReportProperties.	RSDocumentGroupByRefreshSelection
ReportLocation.DeleteReport	ApplyMSTRChartFormats	ReportProperties.
ReportLocation.DisplayDefaultInExcel	ReportProperties.ApplyOutlineWebstyle	RSDocumentLayoutRefreshSelection
ReportLocation.	ReportProperties.ApplyRowHeights	ReportProperties.selectedVisualization
DisplayDefaultInPowerPoint	ReportProperties.authMode	ReportProperties.Server
ReportLocation.DisplayDefaultInWord	ReportProperties.AutoFilter	ReportProperties.ShowAttributeForms
ReportLocation.DisplayReportInExcel	ReportProperties.AutoFitColumns	ReportProperties.ShowDataOnly
ReportLocation.	ReportProperties.AutoFitRows	ReportProperties.ShowHeaderDescriptions
DisplayReportInPowerPoint	ReportProperties.AutoGroup	ReportProperties.StyleSharingMode
ReportLocation.DisplayReportInWord	ReportProperties.BlockDisplayStatus	ReportProperties.StyleSharingName
ReportLocation.DisplayReportXMLInExcel	ReportProperties.BlockID	ReportProperties.TrimWhitespace
ReportLocation.	ReportProperties.ChartAsImage	Reports.Add
DisplayReportXMLInPowerPoint	ReportProperties.ChartCreate2D	Reports.Count
ReportLocation.DisplayReportXMLInWord	ReportProperties.ChartImageType	Reports.ExecuteAndDisplayObjects
ReportLocation.GetDisplaySettings	ReportProperties.ChartShowDataTable	Reports.Item
ReportLocation.GetExcelReportProperties		

Reports.Remove	RSLocation.GetPDFResults	WordReportProperties.
Restriction.Flag	RSLocation.	ApplyOutlineWebstyle
Restriction.Value	GetPowerPointReportProperties	WordReportProperties.ApplyRowHeights
Restrictions.Add	RSLocation.GetPromptAnswerXMLString	WordReportProperties.authMode
Restrictions.Count	RSLocation.GetPrompts	WordReportProperties.AutoFilter
Restrictions.Item	RSLocation.GetReportProperties	WordReportProperties.AutoFitColumns
RSDocument.Access	RSLocation.GetReportResults	WordReportProperties.AutoFitRows
RSDocument.CreationTime	RSLocation.GetWordReportProperties	WordReportProperties.AutoGroup
RSDocument.DefaultViewMode	RSLocation.MessageID	WordReportProperties.BlockDisplayStatus
RSDocument.DefaultVisualization	RSLocation.Name	WordReportProperties.BlockID
RSDocument.Description	RSLocation.ObjectID	WordReportProperties.ChartAsImage
RSDocument.	RSLocation.PDFFileName	WordReportProperties.ChartCreate2D
ExecuteAndDisplayDefaultInExcel	RSLocation.PersistRSDocument	WordReportProperties.ChartImageType
RSDocument.	RSLocation.QueryString	WordReportProperties.
ExecuteAndDisplayDefaultInPowerPoint	RSLocation.RefreshDisplay	ChartShowDataTable
RSDocument.	RSLocation.RefreshReportDisplayPrompts	WordReportProperties.
ExecuteAndDisplayDefaultInWord	RSLocation.RefreshRSDocument	ChartShowLegendKey
RSDocument.	RSLocation.RefreshRSDocument_2	WordReportProperties.Columns
ExecuteAndDisplayDocument	RSLocation.ResultMHT	WordReportProperties.CSVAutoFormat
RSDocument.ExecuteRSDocument	RSLocation.RetrieveVisualizationResults	WordReportProperties.CSVFlatten
RSDocument.FolderID	RSLocation.RSLocationID	WordReportProperties.DisplayColumns
RSDocument.Initialize	RSLocation.SaveExcelReportProperties	WordReportProperties.DisplayRows
RSDocuments.Count	RSLocation.	WordReportProperties.
RSDocument.IsCustomFlashEnabled	SavePowerPointReportProperties	FlashAssetMapFileName
RSDocument.ModificationTime	RSLocation.SaveReportProperties	WordReportProperties.FlashControl Path
RSDocument.ObjectID	RSLocation.SaveWordReportProperties	WordReportProperties.FlashDataFileName
RSDocument.ObjectName	SavedPromptAnswers.Add	WordReportProperties.Flash DataPath
RSDocument.Path	SavedPromptAnswers.Count	WordReportProperties.
RSDocument.SubType	SavedPromptAnswers.Item	FlashDefinitionFileName
RSDocument.Type	SavedPromptAnswers.Remove	WordReportProperties.FlashImagePath
RSDocument.VisualizationList	Shortcut.DefaultViewMode	WordReportProperties.
RSDocument.VisualizationViewMode	Shortcut.DefaultVisualization	FlashLocalizationFileName
RSDocuments.Add	Shortcut.IsCustomFlashEnabled	WordReportProperties.FlashSWFFileName
RSDocuments.Count	Shortcut.IsFlashEnabled	WordReportProperties.
RSDocuments.ExecuteAndDisplayObjects	Shortcut.IsVisualizationEnabled	IsFlashModeAllowed
RSDocuments.Item	Shortcut.Project	WordReportProperties.LastUpdated
RSDocuments.Remove	Shortcut.ProjectID	WordReportProperties.LockPersistence
RSLocation.answerPrompt	Shortcut.ShortcutName	WordReportProperties.loginID
RSLocation.Cancel	Shortcut.TargetID	WordReportProperties.OutlineMode
RSLocation.CancelRSDocument	Shortcut.TargetName	WordReportProperties.
RSLocation.Delete	Shortcut.TargetSubType	PageHeadersCompactOn
RSLocation.DeleteRSDocument	Shortcut.TargetType	WordReportProperties.PageHeadersOn
RSLocation.DisplayDefaultInExcel	Shortcut.VisualizationList	WordReportProperties.PageName
RSLocation.DisplayDefaultInPowerPoint	Shortcut.VisualizationViewMode	WordReportProperties.PasteType
RSLocation.DisplayDefaultInWord	Shortcuts.Count	WordReportProperties.presentAs
RSLocation.DisplayRSDocumentInExcel	Shortcuts.Item	WordReportProperties.Project
RSLocation.DisplayRSDocumentInExcel_2	VisualizationProperties.DisplayName	WordReportProperties.ProjectSource
RSLocation.	VisualizationProperties.StyleName	WordReportProperties.Rank
DisplayRSDocumentInPowerPoint	WordReportProperties.ApplyBorder	WordReportProperties.
RSLocation.	WordReportProperties.	ReportDisplayStatus
DisplayRSDocumentInPowerPoint_2	ApplyColumnWidths	WordReportProperties.reportID
RSLocation.DisplayRSDocumentInWord	WordReportProperties.	WordReportProperties.reportName
RSLocation.DisplayRSDocumentInWord_2	ApplyDateTimeFormatting	WordReportProperties.ReportPath
RSLocation.GetDisplaySettings	WordReportProperties.	WordReportProperties.Rows
RSLocation.GetExcelReportProperties	ApplyMSTRChartFormats	WordReportProperties.Server
		WordReportProperties.

ShowAttributeForms	IDSSCatalog	IDSSContentServer
WordReportProperties.	IDSSCatalog2	IDSSContentSource
ShowHeaderDescriptions	IDSSCatalog3	IDSSContextCollection
WordReportProperties.StyleSharingMode	IDSSCatalogDefn	IDSSContractInfo
WordReportProperties.StyleSharingName	IDSSCatalogDefn2	IDSSContractInfoCollection
WordReportProperties.TrimWhitespace	IDSSCatalogDefn3	IDSSContractManager
	IDSSCatalogDefn4	IDSSContractManager2
	IDSSCellData	IDSSCreate
	IDSSChangeJournalEntry	IDSSCreate2
	IDSSChangeJournalLinkItemEntry	IDSSCreate3
	IDSSChangeJournalLinkItemList	IDSCubeAdmin
	IDSSChangeJournalList	IDSCubeInfo
	IDSSChangeJournalObjectEntry	IDSCubeInfo2
	IDSSChangeJournalObjectList	IDSCubeInfo3
	IDSSChangeJournalSearch	IDSCubeInfos
	IDSSChangeTypes	IDSCubeSource
	IDSSChannel	IDSCubeSourceEvents
	IDSSChannels	IDSCustomGroup
	IDSSClientServices	IDSCustomGroup2
	IDSSClientServices2	IDSCustomGroup3
	IDSSClientServices3	IDSCustomSQL
	IDSSClientServices4	IDSCustomSQLColumn
	IDSSClientServices5	IDSCustomSQLColumn2
	IDSSClientServices6	IDSCustomSQLQueryBuilder
	IDSSClientServices7	IDSCustomSubtotal
	IDSSClusterManager	IDSCustomSubtotals
	IDSSClusterManager2	IDSDashboardTemplate
	IDSSClusterManager3	IDSDashboardTemplateAndThumbnail
	IDSSClusterManager4	IDSDashboardTemplateAndThumbnails
	IDSSClusterMember	IDSDatabaseAdministrator
	IDSSClusterMembers	IDSDatabaseMonitor
	IDSSCollection	IDSDatabaseMonitors
	IDSSCollection2	IDSDataElement
	IDSSColumn	IDSDataElement2
	IDSSColumn2	IDSDataElements
	IDSSColumns	IDSDatamapColumn
	IDSSColumns2	IDSDatamart
	IDSSColumns3	IDSDatamartColumn
	IDSSColumns4	IDSDatamartReport
	IDSSCommand	IDSDatamartUnit
	IDSSCommands	IDSDataSource
	IDSSComments2	IDSDatasource2
	IDSSComponent	IDSDatasource3
	IDSSConditionality	IDSDatasource4
	IDSSConfiguration	IDSDatasource5
	IDSSConfiguration2	IDSDatasource6
	IDSSConsolidation	IDSDatasourceEnumerator
	IDSSConsolidation2	IDSDatatype
	IDSSConsolidationElement	IDSSDBConnection
	IDSSConsolidationElement2	IDSSDBConnectionMap
	IDSSConsolidationElement3	IDSSDBConnectionMaps
	IDSSConstant	IDSSDBConnections
	IDSSConstant2	IDSSDBElementServer
		IDSSDBLogin

IDSSDBMonitor	IDSSDimensionInfo4	IDSSDocumentManipulator9
IDSSDBMonitors	IDSSDimensionInfos	IDSSDocumentNetClient
IDSSDBMonitors2	IDSSDimensionInfos2	IDSSDocumentServices
IDSSDBMS	IDSSDimensionMap	IDSSDocumentSource
IDSSDBRole	IDSSDimensions	IDSSDocumentSource2
IDSSDBRole2	IDSSDimensions2	IDSSDocumentSource3
IDSSDBRole3	IDSSDimensionWrapper	IDSSDocumentSource4
IDSSDBRoleMonitor	IDSSDimty	IDSSDocumentSourceEvents
IDSSDBRoleMonitors	IDSSDimtyUnit	IDSSDocumentSourceServer
IDSSDBRoles	IDSSDirectedAttribute	IDSSDrillAction
IDSSDBTable	IDSSDirectedAttributes	IDSSDrillAction2
IDSSDBTable2	IDSSDocumentDefinition	IDSSDrillAction3
IDSSDBTable3	IDSSDocumentDefinition2	IDSSDrillAction4
IDSSDBTable4	IDSSDocumentDefinition3	IDSSDrillAction5
IDSSDBTableKey	IDSSDocumentDefinition4	IDSSDrillAction6
IDSSDBTableKey2	IDSSDocumentDefinition5	IDSSDrillAction7
IDSSDBTableKeys	IDSSDocumentInstance	IDSSDrillMap
IDSSDBTables	IDSSDocumentInstance2	IDSSDrillPath
IDSSDBTables2	IDSSDocumentInstance3	IDSSDrillPath2
IDSSDelete	IDSSDocumentInstance4	IDSSDrillPath3
IDSSDeliverySection	IDSSDocumentInstance5	IDSSElement
IDSSDeliverySections	IDSSDocumentInstance6	IDSSElement2
IDSSDependents	IDSSDocumentInstance7	IDSSElement3
IDSSDerivedElements	IDSSDocumentInstance8	IDSSElementFilters
IDSSDevice	IDSSDocumentManipulator	IDSSElementNetClient
IDSSDevices	IDSSDocumentManipulator10	IDSSElementNetServer
IDSSDimension	IDSSDocumentManipulator11	IDSSElementReport
IDSSDimension2	IDSSDocumentManipulator2	IDSSElements
IDSSDimension3	IDSSDocumentManipulator3	IDSSElements2
IDSSDimension4	IDSSDocumentManipulator4	IDSSElementsCollection
IDSSDimension5	IDSSDocumentManipulator5	IDSSElementServer
IDSSDimensionInfo	IDSSDocumentManipulator6	IDSSElementsObject
IDSSDimensionInfo2	IDSSDocumentManipulator7	IDSSElementSource
IDSSDimensionInfo3	IDSSDocumentManipulator8	

COLL-1066 0512



MicroStrategy Incorporated | 1850 Towers Crescent Plaza, Tysons Corner, VA 22182 | 703-848-8600 | www.microstrategy.com