

IBM® Analytical Decision Management
Version 17 Release 0

*Performance and Troubleshooting
Guide*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 25.

Product Information

This edition applies to version 17, release 0, modification 0 of IBM Analytical Decision Management and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2010, 2015.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v	Chapter 4. Performance best practices	13
Chapter 1. Basic topics	1	Number of source records	13
Techniques for troubleshooting problems	1	Number of fields in the project data model.	14
Searching knowledge bases	2	Uploading data	14
Getting fixes	3	Dimensions (hierarchies) and dimension members	14
Contacting IBM Support	4	Optimizing stream generation	15
Exchanging information with IBM	4	SQL pushback	15
Sending information to IBM Support	4	SQL pushback and model derived attributes	15
Receiving information from IBM Support.	5	Static versus dynamic allocation	18
Subscribing to Support updates	5	Referencing models.	19
Chapter 2. General troubleshooting	7	Referencing external rules	20
Troubleshooting checklist for IBM Analytical Decision		CPLEX optimization	20
Management	7	Problem size	21
Collecting data for IBM Analytical Decision		Business problem modeling in the OPL	22
Management	7	OPL model complexity	22
Chapter 3. IBM Analytical Decision		Data.	22
Management troubleshooting	9	Algorithm	23
Out of memory	9	Resources (memory and CPU) available	23
Correcting incomplete Gains chart output	9	Database access when using the Scoring Service	23
Managing versions of streams	9	Notices	25
Verifying IBM SPSS Modeler Server connectivity	10	Trademarks	26
Segment rules	10	Index	29
CPLEX models	10		

Preface

About IBM Business Analytics

IBM Business Analytics software delivers complete, consistent, and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of business intelligence, predictive analytics, financial performance and strategy management, and analytic applications provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices, and professional services, organizations of every size can drive the highest productivity, confidently automate decisions, and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government, and academic customers worldwide rely on IBM® SPSS® technology as a competitive advantage in attracting, retaining, and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises - able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative, visit <http://www.ibm.com/spss>.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. web site at <http://www.ibm.com/support>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Chapter 1. Basic topics

Techniques for troubleshooting problems

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the IBM technical-support representative know where to start to find the cause of the problem. This step includes asking yourself basic questions:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, which can then lead you to a problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is: What is the problem? This question might seem straightforward; however, you can break it down into several more-focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to consider when you are investigating problems.

The following questions help you to focus on where the problem occurs to isolate the problem layer:

- Is the problem specific to one platform or operating system, or is it common across multiple platforms or operating systems?
- Is the current environment and configuration supported?

If one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily develop a timeline by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log.

To develop a detailed timeline of events, answer these questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can give you a frame of reference in which to investigate the problem.

Under which conditions does the problem occur?

Knowing which systems and applications are running at the time a problem occurs is an important part of troubleshooting. These questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

Can the problem be reproduced?

From a troubleshooting standpoint, the ideal problem is one that can be reproduced. Typically, when a problem can be reproduced you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve. However, problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur. If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

- Can the problem be re-created on a test system?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, or a particular application?

Searching knowledge bases

You can often find solutions to problems by searching IBM knowledge bases. You can optimize your results by using available resources, support tools, and search methods.

About this task

You can find useful information by searching the information center, but sometimes you need to look beyond the information center to answer your questions or resolve problems.

Procedure

To search knowledge bases for information that you need, use one or more of the following approaches:

- Find the content you need by using the IBM Support Portal. Add **SPSS Decision Management** to your list of products, along with any other IBM products you have.

The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution. Familiarize yourself with the IBM Support Portal by viewing the demo videos about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.

- Search for content about IBM Analytical Decision Management by using one of the following additional technical resources:
 - IBM Analytical Decision Management technotes and APARs (problem reports)
 - IBM Analytical Decision Management Support web site
 - IBM Redbooks®
- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com® page.
- Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com.

Tip: Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.

Getting fixes

A product fix might be available to resolve your problem.

Procedure

To find and install fixes:

1. Determine which fix you need via IBM Fix Central. Note that at the time of writing this guide, no fixes were yet available for IBM Analytical Decision Management.
 - a. Check the recommended fixes for IBM Analytical Decision Management to confirm that your software is at the latest maintenance level.
 - b. Determine whether IBM has already published an individual fix to resolve your problem. Check the list of problems that have been fixed in the IBM Analytical Decision Management readme documentation that is available for each listed fix pack and refresh pack.
2. Download the fix.
 - a. Open the download document and follow the link in the Download Package section.
 - b. When downloading the file, ensure that the name of the maintenance file is not changed. Changes might be intentional changes or they might be inadvertent changes that are caused by certain web browsers or download utilities.
3. Apply the fix.
 - a. Follow the instructions in the Installation Instructions section of the download document.
 - b. Subscribe to receive weekly e-mail notifications about fixes and other IBM Support information.

Contacting IBM Support

IBM Support provides assistance with product defects, answering FAQs, and performing rediscovery.

Before you begin

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company must have an active IBM software subscription and support contract, and you must be authorized to submit problems to IBM. For information about the types of support available, see the Support portfolio topic in the *Software Support Handbook*.

Procedure

Complete the following steps to contact IBM Support with a problem:

1. Define the problem, gather background information, and determine the severity of the problem. For help, see the Getting IBM® support topic in the *Software Support Handbook*.
2. Gather diagnostic information. See the topic “Collecting data for IBM Analytical Decision Management” on page 7 for more information.
3. Submit your problem to IBM Support in one of the following ways:
 - Online through the IBM Support Portal: You can open, update, and view all your Service Requests from the Service Request portlet on the Service Request page.
 - By phone: For the phone number to call in your region, see the Directory of worldwide contacts web page.

Results

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support web site, so that other users who experience the same problem can benefit from the same resolution.

Be prepared to work with the IBM technical-support representative by using IBM Assist On-Site, which is a remote-assistance plug-in that you can download to your computer. The IBM technical-support representative can use IBM Assist On-Site to view your desktop and share control of your mouse and keyboard. This tool can shorten the time it takes to identify the problem, collect the necessary data, and solve the problem. For more information, see IBM Assist On-Site.

Exchanging information with IBM

To diagnose or identify a problem, you might need to provide IBM Support with data and information from your system. In other cases, IBM Support might provide you with tools or utilities to use for problem determination.

Sending information to IBM Support

To reduce the time it takes to resolve your problem, you can send trace and diagnostic information to IBM Support.

Procedure

To submit diagnostic information to IBM Support:

1. Open a problem management record (PMR). You can use the Service Request tool.

2. Collect the diagnostic data you need. Diagnostic data helps reduce the time it takes to resolve your PMR.
3. Compress the files by using the ZIP or TAR format.
4. Transfer the files to IBM. You can use one of the following methods:
 - The Service Request tool
 - Standard data upload methods: FTP, HTTP
 - Secure data upload methods: FTPS, SFTP, HTTPS
 - E-mail

All of these data exchange methods are explained on the IBM Support web site.

Receiving information from IBM Support

Occasionally an IBM technical-support representative will ask you to download diagnostic tools or other files. You can use FTP to download these files.

Before you begin

Ensure that your IBM technical-support representative provided you with the preferred server to use for downloading the files and the exact directory and file names to access.

Procedure

To download files from IBM Support:

1. Use FTP to connect to the site that your IBM technical-support representative provided and log in as **anonymous**. Use your e-mail address as the password.
2. Change to the appropriate directory:
 - a. Change to the /fromibm directory.
`cd fromibm`
 - b. Change to the directory that your IBM technical-support representative provided.
`cd nameofdirectory`
3. Enable binary mode for your session.
`binary`
4. Use the get command to download the file your IBM technical-support representative specified.
`get filename.extension`
5. End your FTP session.
`quit`

Subscribing to Support updates

To stay informed of important information about the IBM products you use, you can subscribe to updates.

About this task

By subscribing to receive updates, you can receive important technical information and updates for specific IBM Support tools and resources. You can subscribe to updates by using one of two approaches:

RSS feeds and social media subscriptions

The following RSS feeds and social media subscriptions are available for IBM Analytical Decision Management:

- IBM Analytical Decision Management Knowledge Center, which has an RSS feed you can use to be informed of updates to the information.

For general information about RSS, including steps for getting started and a list of RSS-enabled IBM web pages, visit the www.ibm.com/software/support/rss/ site.

My Notifications

With My Notifications, you can subscribe to Support updates for any IBM product. (My Notifications replaces My Support, which is a similar tool you might have used in the past.) With My Notifications, you can specify that you want to receive daily or weekly e-mail announcements. You can specify what type of information you want to receive (such as publications, hints and tips, product flashes (also known as alerts), downloads, and drivers). My Notifications enables you to customize and categorize the products about which you want to be informed and the delivery methods that best suit your needs.

Procedure

To subscribe to Support updates:

1. Subscribe to the IBM Analytical Decision Management Knowledge Center RSS feed, which enables you to stay informed of regular updates to the information. To set up your RSS feed, go to the information center RSS feed setup page.
2. Subscribe to the IBM Analytical Decision Management RSS feed, which enables you to stay up-to-date with the latest IBM Software Support content for the product. To subscribe, go to the IBM Analytical Decision Management RSS feeds page.
3. To subscribe to My Notifications, go to the IBM Support Portal and click **Manage all my subscriptions** in the Notifications portlet.
4. Sign in using your IBM ID and password, and click **Submit**.
5. Identify what and how you want to receive updates.
 - a. Click the **Subscribe** tab.
 - b. Click **Information Management** or another software brand or type of hardware.
 - c. Click **SPSS Decision Management** and select the names of any additional products you use. Click **Continue**.
 - d. Select your preferences for how to receive updates, whether by e-mail, online in a designated folder, or as an RSS or Atom feed.
 - e. Select the types of documentation updates you want to receive, for example, new information about product downloads and discussion group comments.
 - f. Click **Submit**.

Results

Until you modify your RSS feeds and My Notifications preferences, you receive notifications of updates that you have requested. You can modify your preferences when needed (for example, if you stop using one product and begin using another product).

Related information

[IBM Software Support RSS feeds](#)

[Subscribe to My Notifications support content updates](#)

[My notifications for IBM technical support sign-in page](#)

[My notifications for IBM technical support overview tour](#)

Chapter 2. General troubleshooting

Troubleshooting checklist for IBM Analytical Decision Management

Are the required prerequisites installed and running?

IBM Analytical Decision Management requires the following prerequisite software. When you encounter problems, verify the following prerequisites are installed properly. See the installation documentation for more detailed system requirements information.

- IBM SPSS Collaboration and Deployment Services Server. See the IBM SPSS Collaboration and Deployment Services documentation for installation instructions and troubleshooting information.
- IBM SPSS Modeler Server. You can install on a different machine or the same machine as IBM SPSS Collaboration and Deployment Services, however IBM SPSS Modeler Server must be running in the same time zone as the application server used by IBM SPSS Collaboration and Deployment Services. See the IBM SPSS Modeler documentation for installation instructions.
- IBM SPSS Modeler Adapter. This is provided with IBM SPSS Modeler, and must be installed on the same machine where IBM SPSS Collaboration and Deployment Services is installed. See the IBM SPSS Modeler documentation for installation instructions.

The IBM SPSS Collaboration and Deployment Services Server and the IBM SPSS Modeler Server must both be running before launching IBM Analytical Decision Management. Wait several minutes after starting them before launching the product.

Have you set up the required server definition and credentials for IBM SPSS Modeler Server?

Before using IBM Analytical Decision Management, you must use IBM SPSS Collaboration and Deployment Services Deployment Manager to set up a server definition and credentials for your IBM SPSS Modeler Server. Otherwise users will be unable to log in to IBM Analytical Decision Management. See the IBM Analytical Decision Management installation documentation for instructions.

Are you using a custom URL?

The default IBM Analytical Decision Management URL is still the same (*http://hostname:port/DM*). But starting with version 6 of IBM SPSS Collaboration and Deployment Services, the base URL for launching IBM SPSS Collaboration and Deployment Services browser-based applications and IBM Analytical Decision Management is now customizable.

If you have problems accessing the IBM Analytical Decision Management URL or any IBM SPSS Collaboration and Deployment Services URLs, an administrator may have customized them. Contact your administrator. More information is available in the *Application context roots* chapter of the *IBM SPSS Collaboration and Deployment Services Repository Installation and Configuration Guide*.

Collecting data for IBM Analytical Decision Management

When troubleshooting IBM Analytical Decision Management problems, and before contacting IBM Support, the following data should be collected.

- Take note of the following information. Much of this information can be collected using the browser-based IBM SPSS Collaboration and Deployment Services Deployment Manager (*http://hostname:port/security/login*). After logging on, click **About** and then click **Show Detail**. Versions of many components will be listed. You can also use the links at the bottom of the screen to download version details, system details, and logs.

- Operating system
- IBM Analytical Decision Management version
- IBM SPSS Collaboration and Deployment Services version
- IBM SPSS Modeler Server version
- Application server and version
- Database and version
- Take note of which IBM Analytical Decision Management application was being used when the problem occurred, along with the specific data source used.
- If the IBM Analytical Decision Management installation fails, see the log files in the IBM SPSS Collaboration and Deployment Services \Server\log directory (for example, *C:\Program Files\IBM\SPSS\Deployment\7.0\Server\log*).

Chapter 3. IBM Analytical Decision Management troubleshooting

The topics in this section provide troubleshooting information related to IBM Analytical Decision Management. For troubleshooting information related to IBM SPSS Collaboration and Deployment Services, see the *IBM SPSS Collaboration and Deployment Services Troubleshooting Guide*.

Out of memory

If you encounter out of memory issues, you may need to increase your application server memory setting. For example, if the default WebSphere Application Server memory setting of 1024MB is not large enough, increasing it to 2048MB or larger may solve your out of memory issues.

On WebSphere Application Server:

Open the WebSphere Application Server administrative console and change the **Maximum heap size** setting to a larger value such as **2048**.

OR

Open the WebSphere *server.xml* file (for example, *C:\profiles\AppSrv01\config\cells\Config9601VM0Node01Cell\nodes\Config9601VM0Node01\servers\server1\server.xml*) and change the `maximumHeapSize` setting to a larger value such as 2048.

On JBoss Application Server:

Open the `$JBOSS_EAP_HOME/bin/standalone.conf.bat` file and change the memory setting:

```
rem # JVM memory allocation pool parameters - modify as appropriate.
set "JAVA_OPTS=-Xms1200M -Xmx1300M -XX:MaxPermSize=256M"
```

On Weblogic Application Server:

Open the `$domain_home/bin/setCDSEnv.sh` file and change the memory setting:

```
USER_MEM_ARGS="$USER_MEM_ARGS -Xms256m -Xmx1536m
-XX:MaxPermSize=512m"
```

Correcting incomplete Gains chart output

If you are using the JRockit JVM, you may see incomplete *Gains* chart output. When you run a job containing an IBM SPSS Modeler scenario file, with an Evaluation type of *Gains*, the resulting *Gains* chart may be incomplete.

JRockit produces wrong numerical results for expressions including NaN. To work around these issues, specify `-XstrictFP` as a parameter to the JVM or switch to a different JVM.

Managing versions of streams

IBM Analytical Decision Management projects are stored as IBM SPSS Modeler streams in the repository. Each time a stream is saved in the repository, a new version is created. Streams with a large number of versions may perform slower than streams with a smaller number of versions. We recommend managing the number of versions that exist in the repository by not keeping versions that are no longer needed. If

you have an extremely large number of unneeded versions in your repository and are having trouble cleaning them up, you can contact your IBM SPSS Support representative for a repository utility that can help automate the clean-up work.

Before cleaning up old versions, we highly recommend backing up your repository as a precaution in case any needed files are accidentally deleted.

Verifying IBM SPSS Modeler Server connectivity

When IBM Analytical Decision Management cannot connect to the IBM SPSS Modeler Server, a vague error message is displayed when users try to log on. If you encounter this problem, do the following:

- Verify the server definition and credentials definition for the IBM SPSS Modeler Server are correct in IBM SPSS Collaboration and Deployment Services Deployment Manager client, and browser-based IBM SPSS Collaboration and Deployment Services Deployment Manager. See the *IBM Analytical Decision Management Installation Guide* for more information.
- Verify the IBM SPSS Modeler Server is set up correctly. See the following Technote for details.
<http://www-01.ibm.com/support/docview.wss?uid=swg21488191>

Segment rules

When creating segment rules in IBM Analytical Decision Management, note that a rule such as Age BETWEEN 50 and 20 will not return results. It is the same as the expression $x \leq \text{value}$ and $\text{value} < y$, which can never be true. This is consistent with SQL Server behavior. Age BETWEEN 20 and 50 is the correct format.

CPLEX models

Advanced users can tune the CPLEX parameters for IBM Analytical Decision Management to attain better performance on large data sets. We also recommend minimizing the number of possible allocations presented to CPLEX (the data volume). In other words, you should reduce the number of combinations (for example, the number of alternative offers allocated to each customer).

Administrators can use the **Advanced Options** button on the Optimize tab in IBM Analytical Decision Management to export CPLEX *.lp* or *.mps* files for troubleshooting purposes or to control certain CPLEX parameters.

Following is a list of CPLEX parameters that can be used for the CPLEX node in IBM SPSS Modeler client. These are the valid custom names that are recognized by the CPLEX integration with IBM Analytical Decision Management.

Important: Refer to the IBM ILOG CPLEX Optimization Studio documentation for details about these settings. Note that they may have an adverse affect on IBM Analytical Decision Management's performance.

Table 1. CPLEX parameters

Boolean Settings	Floating Point Settings	Integer Settings	String Settings
CloneLog	EpMrk	AdvInd	WorkDir
PreInd	EpOpt	RootAlg	FileEncoding
MIPOrdInd	EpPer	NodeAlg	APIEncoding
MPSLongNum	EpRHS	MIPEmphasis	IntSolFilePrefix
LBHeur	NetEpOpt	AggInd	
PerInd	NetEpRHS	ClockType	

Table 1. CPLEX parameters (continued)

Boolean Settings	Floating Point Settings	Integer Settings	String Settings
PreLinear	TiLim	CraInd	
DataCheck	DetTiLim	DepInd	
QPmakePSDInd	TuningTiLim	PreDual	
MemoryEmphasis	BtTol	PrePass	
NumericalEmphasis	CutLo	RelaxPreInd	
	CutUp	RepeatPresolve	
	EpGap	Symmetry	
	EpInt	DPriInd	
	EpAGap	PriceLim	
	EpRelax	SimDisplay	
	ObjDif	NetFind	
	ObjLLim	PerLim	
	ObjULim	PPriInd	
	PolishTime	ReInv	
	PolishAfterEpAGap	ScaInd	
	PolishAfterEpGap	Threads	
	PolishAfterTime	AuxRootThreads	
	ProbeTime	ParallelMode	
	RelObjDif	SingLim	
	CutsFactor	Reduce	
	TreLim	ColReadLim	
	SolnPoolGap	RowReadLim	
	SolnPoolAGap	WriteLevel	
	WorkMem	SiftDisplay	
	BarEpComp	SiftAlg	
	BarQCPEpComp	BrDir	
	BarGrowth	Cliques	
	BarObjRng	CoeRedInd	
	EpLin	Covers	
		MIPDisplay	
		NodeFileInd	
		NodeSel	
		VarSel	
		BndStrenInd	
		FPHeur	
		MIPOrdType	
		FlowCovers	
		ImplBd	
		Probe	
		GUBCovers	

Table 1. CPLEX parameters (continued)

Boolean Settings	Floating Point Settings	Integer Settings	String Settings
		StrongCandLim	
		FracCand	
		FracCuts	
		PreslvNd	
		FlowPaths	
		MIRCuts	
		DisjCuts	
		ZeroHalfCuts	
		MCFCuts	
		AggCutLim	
		EachCutLim	
		DiveType	
		MIPSearch	
		MIQCPStrat	
		SolnPoolCapacity	
		SolnPoolReplace	
		SolnPoolIntensity	
		PopulateLim	
		BarAlg	
		BarColNz	
		BarDisplay	
		BarOrder	
		BarCrossAlg	
		BarStartAlg	
		NetPPriInd	
		NetDisplay	
		ConflictDisplay	
		FeasOptMode	
		TuningMeasure	
		TuningRepeat	
		TuningDisplay	
		MIPKappaStats	
		SolutionTarget	

Chapter 4. Performance best practices

The topics in this section provide performance best practices for IBM Analytical Decision Management. Various practices can impact performance. This section describes implementation approaches that can help you balance performance with your needs. Note that some of the information and recommendations may not apply to your specific environment or usage of the product.

Number of source records

The number of rows of data in your data source can affect IBM Analytical Decision Management performance. In particular, the number of data rows can affect how long it takes to run the following operations:

- Data Scan
- Scenario
- Score

These operations typically have a fixed overhead for initiating the operation and for gathering the results. For non-trivial sized data sources, the actual operations scale linearly with the amount of data processed. All things being equal, an operation that processes twice as many records generally takes twice as long.

The Data tab in an IBM Analytical Decision Management project requires users to select a single source as the *project data model*. The project data model defines the fields that the rest of the project uses. IBM Analytical Decision Management also allows users to create additional sources that are compatible with the project data model source. By creating additional compatible sources with differing numbers of records, the user can choose an appropriately sized source to reduce execution times.

For example, suppose that the following sources are defined:

- A **main source** that contains 1,000,000 records and is set as the project data model.
- A **quick scenario source** that contains 50,000 records and is compatible with the project data model.
- A **medium scenario source** that contains 20,000,000 records and is compatible with the project data model.
- An **all records source** that contains 100,000,000 records and is compatible with the project data model.

Initially, the user performs a data scan using the **main source**. In this case, the records of the main source are sufficient to define all of the fields along with their categorical values.

The user goes to the Global Selections tab and starts creating rules. The user is experimenting with several rules and wants to try them out to see their rough effect. The user chooses the **quick scenario source** to return quick, preliminary results.

Eventually, the user is satisfied with their preliminary efforts and chooses to run a scenario using the larger data sources (**medium scenario source** or **all records source**). Although these take proportionately longer to run, the user is willing to wait for the more comprehensive result.

Data source size and the effects on Test

The Test operation is designed to operate on few records. Often, just the first 10 records are read from the data source and the size of the data source does not matter.

However, if multiple tables are used (for example, Customers and Contacts), then the records from these sources must be joined and be part of the Test run. Because the join can result in all records from the

sources being processed, performance can be greatly improved by using sources with a small number of records.

Data source size and projects that use Optimization

Some IBM Analytical Decision Management applications, such as IBM Analytical Decision Management for Campaign Optimization and IBM Analytical Decision Management for Demand Optimization, use CPLEX based optimization. These applications optimize over all source records as a whole, as opposed to simply one record at a time. As a result, the execution times for operations that run the CPLEX based Optimize step do not scale linearly with the amount of data processed. An increase in the number of records typically results in execution times that are proportionately much longer.

Operations that do not run the Optimize step, such as a data scan, continue to scale linearly.

See “CPLEX optimization” on page 20.

Number of fields in the project data model

Much like how the number data rows can affect IBM Analytical Decision Management performance, the number of fields also has an impact. For non-trivial sized sources, the actual operations scale roughly linearly with the amount of data processed. After ignoring any initialization overhead, an operation that processes twice as many fields will generally take twice as long.

As a best practice, use data sources that only contain the fields you actually need.

Uploading data

The Data tab in IBM Analytical Decision Management allows users to upload data files. This feature is designed for relatively small files. Avoid using this feature for files larger than 750 megabytes, as transfers will take a long time. For large files, use an alternative upload method such as FTP to transfer files directly to the IBM SPSS Modeler Server file system.

Dimensions (hierarchies) and dimension members

In general, adding additional dimensions and dimension members will increase stream complexity and worsen performance.

The overhead of adding a dimension member is dependent on several factors:

- **Child vs parent dimension.** Adding a dimension member to the parent dimension (for example, adding a new campaign) typically involves more overhead than adding another child dimension member (for example, adding another offer). Adding a parent member has a compounding impact due to its allocation rule, which in turn will allocate additional child members.
- **Selection rule.** A member containing a selection rule will require additional processing as compared to a member that does not have a selection rule. Members on the parent dimension (for example, campaign) have a default selection rule. However, results will vary based on the number of records actually filtered out by the rule. A selection rule which filters out most records will have a positive impact to performance because it will result in fewer allocations (to optimize).
- **Variables.** Member-specific variables will require additional processing for each variable. Consider placing prioritization, optimization, and/or return variables at the largest scope possible. For example, consider a project that has 2 campaigns and each campaign will allocate 10 offers. If the variable (probability to respond) is the same for all offers allocated by *campaign1*, then place it on *campaign1* instead of on each of the 10 offers. In contrast, consider a project that has 10 campaigns and 2 *shared* offers. If the variable (probability to respond) is the same for all campaigns, it is better to place it at the offer level (on the 2 offers instead of on the 10 campaigns). Variables typically have less impact than adding dimensions or members.

- **Number of rules.** The number of rules and rule segments (within a selection or allocation rule) can also impact performance if the number of rules becomes large.
- **Model and external rule references.** Referencing models or external rules (rulesets in IBM Operational Decision Management) can also impact performance.
See “Referencing models” on page 19 and “Referencing external rules” on page 20.

Optimizing stream generation

Under certain conditions, IBM Analytical Decision Management can generate a more optimal stream that avoids unnecessary branches and runs more efficiently.

A more efficient stream is generated when all of the dimension members on the leaf level dimension (such as Offers) satisfy all of the following conditions:

- Selection rules are not used on the leaf level dimension (Offer, for example)
- Variables (prioritize/optimize variables, for example) are not used at the leaf level
- Return variables are not used at the leaf level dimension
- The leaf level is not configured as the priorityDimension
- For applications that use an allocation rule, the parent dimension (Campaign, for example) uses allocation rules with the **First valid** option selected (as opposed to **All valid**). Further, the allocation rule must generate a single outcome (for example, **Allocate using rules** or **Allocate randomly** as opposed to **Multiple Allocation**).
- For applications that use an aggregation rule, the parent dimension (Claim Area, for example) uses an aggregation rule where the remainder option is set to **Apply when no rules hit** (as opposed to **Always apply**).

A project that is constructed in this way can be thought of as an "allocation rule only" project and can often be suitable where maximum execution speed is essential. An example is if the project is deployed with the IBM SPSS Analytics Toolkit for InfoSphere Streams.

SQL pushback

SQL Pushback is a technique IBM SPSS Modeler Server uses to improve performance for database sources. Database joins and other computations are automatically embedded into the generated SQL statements. Processing is pushed into the database where it can be handled more efficiently. It can also minimize the amount of data that IBM SPSS Modeler Server must read into memory, which again improves performance.

The use of SQL pushback can result in 10 times better performance (or more).

In IBM Analytical Decision Management generated streams, SQL pushback is often best leveraged when using Analytic Data View sources.

In general, SQL pushback does not work through model references. Model references prevent SQL pushback from working on all downstream nodes. Provided there are no model references preventing it, SQL pushback can be used on Global Selections rules and on the selection rules on the dimension members of the first dimension (Claim Area or Campaign, for example).

SQL pushback and model derived attributes

On the **Derived** sub tab of the **Data** tab in IBM Analytical Decision Management, you can create additional attributes using a model to aggregate data from a secondary data source. By using ODBC sources (either directly or via the Analytic Data View), the join can be pushed back into the database. The resulting performance improvement might be dramatic. In some cases, it can be by a factor of 10 or more (SQL pushback taking one tenth of the time or less as compared to the non-pushback flat file).

However, the join will only push back for the first secondary table. The model reference prevents the join for subsequent model references and subsequent secondary tables from pushing back.

The Analytic Data View source node must be used to achieve SQL pushback with multiple model references and/or multiple secondary tables.

SQL pushback using more than one secondary table

To use SQL pushback with more than one secondary table, the Analytic Data View must be used. Instead of using a model to derive fields from a secondary table, embed the join for the secondary table along with the derivation logic into the Analytic Data View.

For example, suppose you have two database tables: Customer and ContactHistory. To use a model to derive fields from a secondary table in IBM Analytical Decision Management, you need to define ContactHistory as a secondary table:



Figure 1. Secondary table

On the **Data** tab, go to the **Derive** tab and use a model to derive NumContacts.

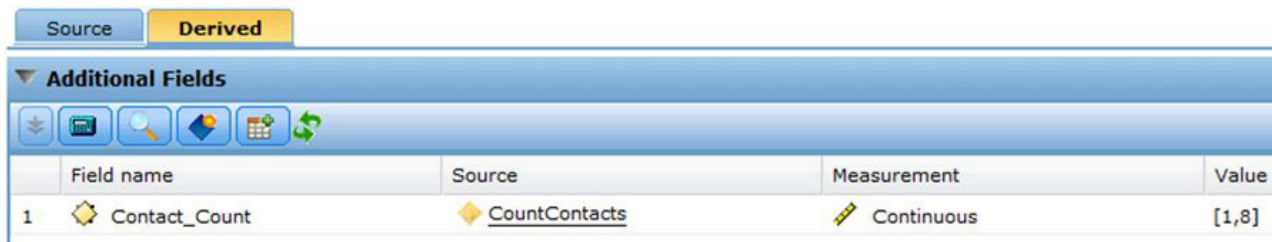


Figure 2. Additional fields

The referenced *CountContacts* model might look like this:



Figure 3. Example CountContacts model

This process can be repeated by adding additional secondary tables and referencing additional models on the **Derive** tab. For example, you may have a ProductsPurchased table and you might have a stream that aggregates (counts) the number products previously purchased by each customer.

In contrast, to accomplish the same thing using an Analytic Data View, create a stream that defines the Data View Table. It will do the join and the aggregation(s). In a simple case with a single secondary table, it might look something like this:

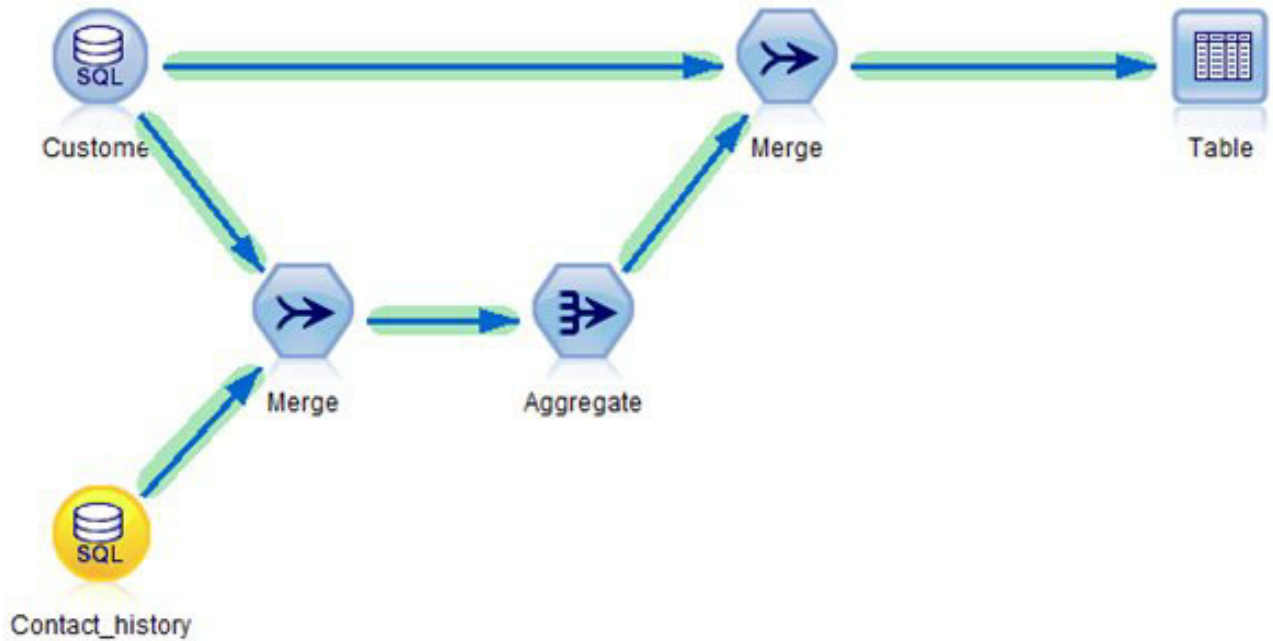


Figure 4. Example using an Analytic Data View

If derived attributes are needed from multiple secondary tables, the stream that is used to define the Analytic Data View might look like this:

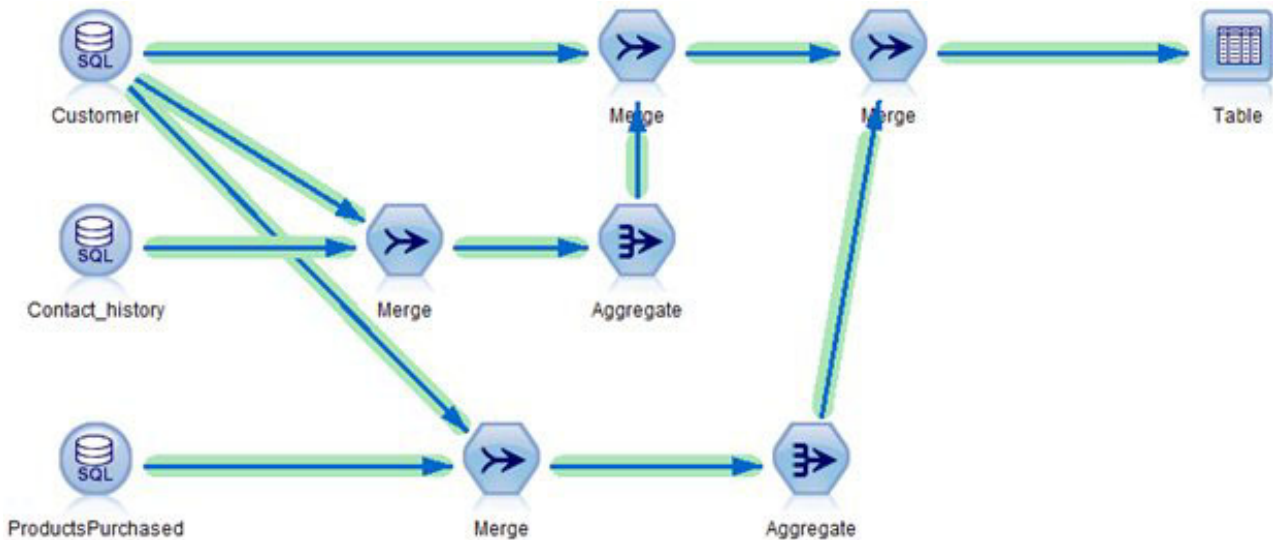


Figure 5. Example using multiple secondary tables

Defining the Analytic Data View in Deployment Manager client looks something like this:

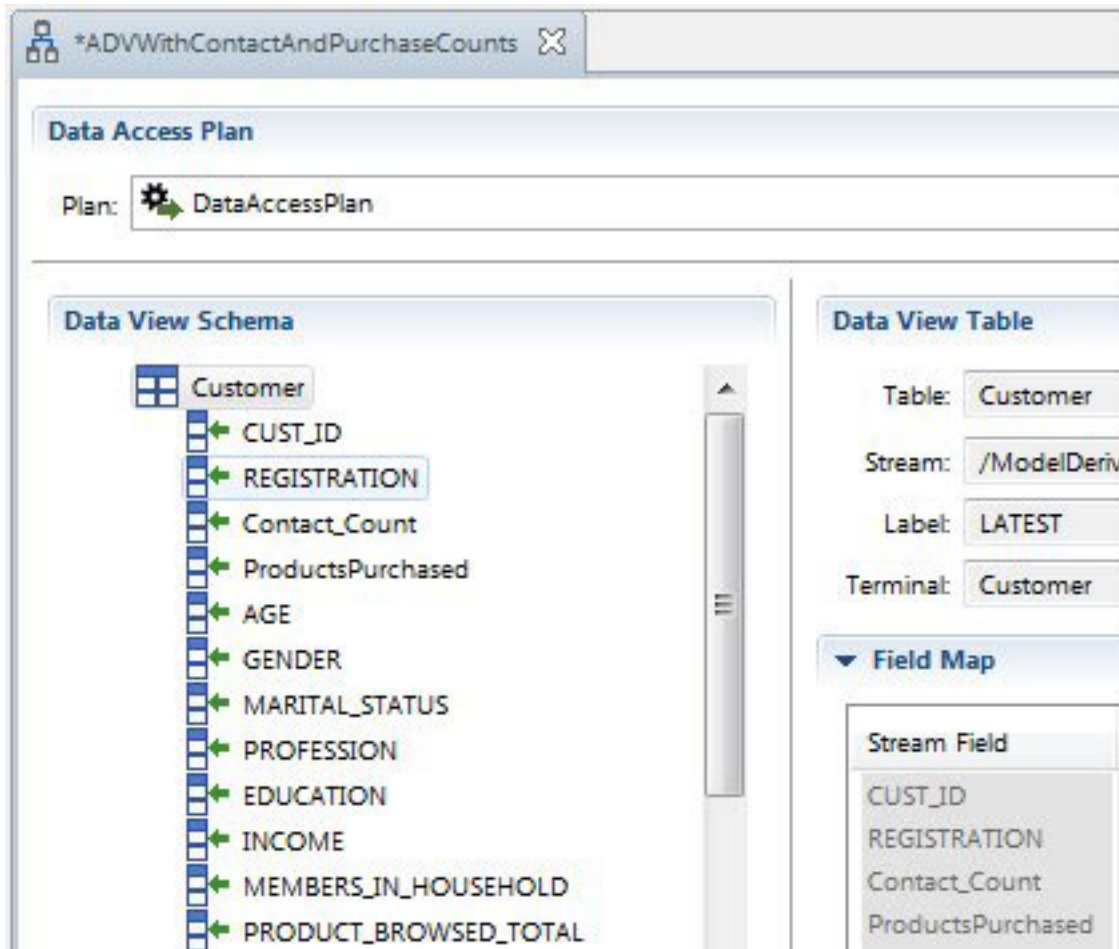


Figure 6. Deployment Manager client

Note that the Analytic Data View schema contains the fields `Contact_Count`, `ProductsPurchased`, and the fields from the `Customer` table. Since these fields are in the Analytic Data View, they will be present in the IBM Analytical Decision Management project data model, and secondary tables will not be defined in the IBM Analytical Decision Management user interface. Again, this approach can use SQL pushback and can be 10 times faster (or more) than alternatives that cannot be pushed back to the database.

Using derived expressions with the Analytic Data View editor is another alternative approach that also makes use of SQL pushback when database sources are used and the source stream types a field as the "record ID." See the IBM SPSS Collaboration and Deployment Services Analytic Data View documentation for more information.

Deriving multiple attributes from the same secondary table

If you choose to derive attributes by using models on the **Derive** sub tab of the **Data** tab, it is generally more efficient to combine the derivation of attributes from a given secondary table into a single referenced model. For example, assume `ContactCount` and `MostRecentContactDate` are both calculated using the `Contact_History` table. It is more efficient to combine the logic into a single referenced stream that produces both outputs (`ContactCount` and `MostRecentContactDate`) as opposed to referencing two separate streams on the **Derive** tab.

Static versus dynamic allocation

IBM Analytical Decision Management supports two types of allocation methods on the Define tab:

- **Static allocation.** All dimension members (myOffer1 and myOffer2, for example) are specified in the dimension tree on the Define tab and become part of the project stream.
- **Dynamic allocation.** Some dimension members are read in from a data source.

Performance varies between these two approaches. In general, static allocation performs better because the member names are local to the project and processing can avoid the extra I/O overhead of reading the member names from an external source.

Static allocation

Since all dimension members (myOffer1 and myOffer2, for example) are part of the project stream, there is no additional I/O overhead in joining and/or reading secondary files. However, the stream typically requires a branch per dimension member. These additional branches can have a negative impact on performance.

Dynamic allocation

For dynamic allocation, allocated members must be read in from a source file. Typically, the members are in a secondary source file. One of two approaches can be used:

- Secondary table is linked to the primary table
- Secondary table is not linked

If the secondary table is linked to the primary table, then a join must be performed. In the current stream construction, this join is not eligible for SQL pushback and the data must be read into IBM SPSS Modeler Server. Performance varies based on the size of the primary and secondary tables.

If the secondary table is not linked to the primary table, then all records are read from the secondary table, resulting in an allocated member for each record in the table. In this case, the total number of records in the secondary table is expected to be relatively small. Even though the join is avoided, static allocation is still faster. However, dynamic allocation offers better scalability as members need not be manually typed into the IBM Analytical Decision Management user interface.

Referencing models

In contrast to using a model to derive a field from a secondary table, IBM Analytical Decision Management also allows models to be used in various conditions such as:

- Referencing a model from a rule
- Referencing a model from a (prioritization/optimization) variable
- Referencing a model from a return field value
- Referencing a model from the data tab to define a new attribute from the primary project data model source (not the secondary source)

Under these conditions, a model is invoked and operates on single row of data. The model is not aggregating multiple rows of data. Since the model operates on a single row of data, calling it requires minimum data overhead. This is in contrast to using a model on the **Derive** sub tab of the **Data** tab to aggregate data from a secondary table where data must first be joined.

Before scoring any rows of data, the model must be prepared for execution. This process involves reading the model from the repository and building the necessary data structures in memory. This overhead occurs regardless of the number of rows of data scored. This can affect the run times for **Scenario**, **Test**, and batch scoring (**Score Now**) operations when using small data sets. It can also affect the time that is required to create a scoring configuration in Deployment Manager client.

The per row execution cost of referencing a model depends on the complexity of the model. For example, calling a simple model that performs a trivial calculation has minimal performance overhead. Calling a complex model that contains many nodes and many branches has a more significant negative impact on performance.

For models deployed through the Scoring Service:

- If the model is referenced directly and is not referenced by a rule, then for a given score request (a given entity such as *customer*), a referenced model is only scored once and only if needed to determine the score. Its result is cached and reused if it is needed again during the same score request (for example, the same model referenced by two different prioritization variables on the **Prioritize** tab).

The model does not run if the result is not needed. For example, suppose that a model is used to derive a field on the **Data** tab and that field is then referenced by a rule in the **Define** tab. If a score request results in the entity being excluded on the **Global Selections** tab, then processing stops (at global selections) and the model does not run.

For scoring on the IBM SPSS Modeler Server (**Scenario** and **Score Now**):

- At run time, while scoring a given entity (input record), results are not cached. If multiple references are made to the same model (on two different variables, for example), then the model runs multiple times - once for each reference. The impact is largely determined by how fast the model runs (simple model versus a large text mining model, for example).
- If the same model is referenced multiple times, you can improve performance by referencing the model on the **Data** tab (**Derive** subtab) to derive a new field. Then, reference the derived field throughout your project. By doing so, the model is always called exactly once. Note that the model reference prevents downstream SQL pushback (it prevents SQL pushback on the global selections).
- If a model is seldom used (for example, it is referenced on a variable for a member such as *offer* that is almost never allocated), then it might be better to place the reference on the member (*offer*) variable instead of on the **Data** tab. If the member is not allocated, then the model does not run.

Referencing external rules

Referencing external rules (such as IBM Operational Decision Management rule sets) has the same implications and behaves the same as referencing models. See “Referencing models” on page 19.

Additionally, external rules can operate in two execution modes:

- **Web service.** In this mode, a remote web service call is performed each time the rule set is called. This approach is often easier to configure. However, it is the slower performing of the two options. Performance depends, in part, on network performance and the performance of the web service.
- **Local (in process).** In this mode, rules run in-process. Data is passed in memory and network overhead is eliminated. This approach is the faster of the two.

IBM Operational Decision Management rules that run locally can result in overall project run time being about 20% faster compared to projects that use IBM Operational Decision Management rules run by a web service. Results vary depending on many factors such as:

- The amount of rule set processing relative to the overall processing of the project
- Network overhead when using the web service mode

CPLEX optimization

IBM Analytical Decision Management applications can be configured to include the Optimization tab and use complex mathematical (CPLEX) based optimization. An example of such an application is IBM Analytical Decision Management for Campaign Optimization.

Scoring in a CPLEX based application occurs in two phases:

1. Phase 1 processes the rules and models. This is the same as any other project that does not use CPLEX.
2. Phase 2 invokes CPLEX to solve the optimization. This phase is unique to IBM Analytical Decision Management projects that use CPLEX.

Projects that use CPLEX based optimization generally do not scale linearly with the number of records scored. All things being equal, increasing the number of records by 10X often results in score times that are significantly larger than 10X (the score times will likely rise exponentially). This occurs because the CPLEX solving phase does not scale linearly. *In fact, the solving time is not predictable. It is possible that seemingly larger problems solve faster than smaller problems.* This occurs because optimization determines the best combination across all entities taken as a whole, as opposed to determining the best result for each entity taken individually. The algorithms that are used do not scale predictably as the problem size increases.

Performance and resource requirements (memory) are largely indeterminate. In general, more combinations and higher complexity require more resources and more time. The actual data values impact the speed of the optimization. Performance does not scale linearly and is not predictable. Some problems cannot be solved.

The performance of CPLEX based optimization is influenced by many factors such as:

- Problem size
- Business problem modeling in the Optimization Programming Language (OPL)
- The complexity of the OPL model
- The data (values) being optimized
- Resources (memory and CPU) available

Problem size

Problem size refers to the number of cases that are input to the optimization process. In IBM Analytical Decision Management, problem size is the number of cases (records) that are passed from the Define step to the Optimization step. In CPLEX terms, problem size is the number of input tuples. Increasing the number of cases can often increase the problem size and result in longer run times.

The number of cases that are input to the optimization phase can be calculated by:

$$\text{Number of entity records} * \text{average number of allocations per entity}$$

To illustrate this, consider an IBM Analytical Decision Management for Campaign Optimization project that has an input file of 2 million entities (customers). Suppose that the Global Selections exclude half of them, leaving 1 million entities. On the Define tab, suppose that on average, each customer is eligible (based on Campaign selection rules) for 10 different campaigns. Further, assume the allocation rule for each campaign is set to *stop on first hit* – resulting in one offer per campaign. The channel dimension is used, but there is only one channel defined. This results in:

$$\text{million customers} * 10 \text{ campaigns} * 1 \text{ offer} * 1 \text{ channel} = 10,000,000 \text{ cases}$$

To see the impact of channels, suppose that a customer can be allocated to one of eight channels:

$$1 \text{ million customers} * 10 \text{ campaigns} * 1 \text{ offer} * 8 \text{ channels} = 80,000,000 \text{ cases}$$

In addition to the added channels, the Campaign allocation rule is not set to stop on first hit, but to execute all. On average, for each campaign, 15 offers are allocated to each customer.

$$1 \text{ million customers} * 10 \text{ campaigns} * 15 \text{ offers} * 8 \text{ channels} = 1,200,000,000 \text{ cases}$$

The task of the optimizer is to determine the optimal combination of the 1,200,000,000 tuples given the limits that are imposed by the constraints.

Business problem modeling in the OPL

The way the business problem is modeled in the Optimization Programming Language (OPL) model can have a large impact on performance. When creating custom applications, the application designer must take care when writing the OPL so that the models can be solved efficiently. Different approaches in the OPL model can result in different performance.

OPL model complexity

The complexity of the Optimization Programming Language (OPL) model influences performance. Following are some factors that influence model complexity:

- **The Objective function.** A more complex objective function increases the time that it takes to produce the optimal solution. A linear objective function is less complex than a quadratic objective function. The objective function influences the algorithms that CPLEX can choose from. Quadratic objectives must be solved using the barrier algorithm, while linear objectives allow CPLEX to use more algorithms. In IBM Analytical Decision Management, the objective function is determined by the application you are using (for example, IBM Analytical Decision Management for Campaign Optimization uses Mixed Integer Programming).
- **Number and placement of constraints.** Increasing the number of constraints increases complexities. Note that a single constraint gesture in IBM Analytical Decision Management might be reflected in multiple constraints to CPLEX.
- **Modeling of the constraints.** Again, the way a business problem is modeled in the OPL can impact performance. This is true in the modeling of constraints. When creating custom applications, the application designer must take care when modeling business constraints in the OPL.

Global and entity constraints

For example, a constraint in IBM Analytical Decision Management of `Max_Offers =1` placed on the entity might be looked upon as a single constraint to the IBM Analytical Decision Management user. However, during the optimization process, this really means that each of the entities has a `Max_Offers` constraint of 1. In effect, in the previous example, this translates to 1,000,000 constraints in CPLEX because it must be stated on each entity.

Contrast this with a `Total Budget` constraint, which is really a single global constraint as seen by CPLEX optimization. `Total Budget` results in a single constraint in CPLEX. The budget constraint sums up all the "places" where the budget can be spent, and can be expressed by a single sum.

Dimension member constraints

An example of a constraint on a dimension member is `Offers Available`. If there are 100 different offers, then CPLEX sees it as 100 separate constraints. CPLEX considers a single shared offer (an offer that is shared across multiple campaigns) as one unique offer with one constraint. 10 campaigns each allocating a single shared offer results in one `Offers Available` constraint, while 10 campaigns that allocate a single distinct offer per campaign results in 10 `Offers Available` constraints (one per unique offer).

Data

The actual data values can impact optimization speed.

Data values impact how the processing progresses and how long it takes. This is known as *data variability*. Two problems of the same size can differ considerably regarding the amount of time (memory and CPU) needed to solve the problem. This is due to differences in the model structure and/or the variables on which decisions must be made by the algorithm (a solution represents all the decisions on all the variables of the model).

Depending on the data, some problems might not be solvable. For example, consider a problem with 10000 entities, each entity must receive an offer, each offer costs more than \$1, and the budget is \$9000.

Algorithm

Based on the objective function and other inputs, CPLEX chooses an algorithm to solve the problem. Different algorithms have differing capabilities and perform differently.

Applications that require Mixed Integer Programming (MIP) can take a long time to solve. The reason is that no efficient (by the measure of polynomial metric) algorithm currently exists for Mixed Integer Programming. This is in contrast to Linear Programming, where efficient algorithms do exist.

Resources (memory and CPU) available

The resources available (memory and CPU) influence the time it takes to solve a CPLEX problem. Results vary. Memory is taken to hold the initial setup of the model in terms of the "matrix." Then, memory is taken to hold the search tree. Once the RAM is exhausted but the solve is not finished, it's possible to migrate the search tree to the hard disk (a CPLEX setting exists for whether this is done or not). Since the access time for disk is much longer than for RAM, the search significantly slows down.

CPLEX uses multiple threads whenever possible. By default, the number of threads it creates is determined by the number of CPUs on the system. Running on a multi-processor system can reduce the time that it takes to produce the optimized solution. The threads process the branches in the search tree as parallel as possible. By default, the parallel CPLEX algorithm is used.

Database access when using the Scoring Service

This section includes considerations for deploying IBM Analytical Decision Management projects to the IBM SPSS Collaboration and Deployment Services Scoring Service for real time scoring.

Unlike batch scoring, where all records of a database table are scored, real time scoring operates on one record (the record being scored) at a time. When the Scoring Service receives a scoring request, the caller can pass the input data on the call or pass a key field so that the Scoring Service can fetch the data from a database table as defined by a Real Time Data Access Plan of an Analytic Data View.

In either case, SQL pushback does not apply as it would when running a batch stream on the IBM SPSS Modeler Server. Instead, the data that is retrieved from a database is controlled by the filters that are specified in the Real Time Data Access Plan. The Scoring Service uses these filters to generate appropriate WHERE clauses when fetching the data for a score request. Typically, these filters result in few records being fetched.

For example, consider the previous example where the project has Customer and ContactHistory as input tables. The Real Time Data Access Plan can be configured to filter by CUST_ID. The Scoring Service generates the appropriate WHERE clause to fetch just the records for this customer from each table. For the Customer table, there is only one record – the one representing this customer. For the ContactHistory table, zero or more records are fetched – one record for each time the customer was contacted. All contact records for this customer are then passed to the IBM SPSS Modeler stream (referenced from either IBM Analytical Decision Management or from within the Analytic Data View).

Avoid situations where the number of filtered records is large (for example, where the customer is typically contacted thousands of times).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.

Index

F

fixes

getting 3

I

IBM Support

contacting 4

sending and receiving information 4

subscribing to updates 5

troubleshooting (*continued*)

identifying problems 1

known issues 9

performance best practices 13

searching for troubleshooting
solutions 2

searching knowledge bases 2

subscribing to Support updates 5

systematic problem solving 1

P

performance best practices 13

allocation methods 18

CPLEX optimization 20

algorithm 23

business problem modeling in the

OPL 22

data 22

dimension member constraints 22

global and entity constraints 22

memory and CPU 23

OPL model complexity 22

problem size 21

data upload 14

database access with Scoring

Service 23

dimension members 14

dimensions (hierarchies) 14

dynamic allocation 18

number of fields in the project data

model 14

number of source records 13

optimized stream generation 15

referencing external rules 20

referencing models 19

SQL pushback 15

deriving multiple attributes from

the same secondary table 18

SQL pushback and model derived

attributes 15

SQL pushback using more than one

secondary table 16

static allocation 18

static vs dynamic allocation 18

problem determination

exchanging information with IBM

Support 4

T

troubleshooting

checklist 7

collecting data 7

contacting IBM Support 4

exchanging information with IBM

Support 4

getting fixes 3

