|  |
| --- |
| How to get the maximum return on your forum question?  Hints & tips to use the forum efficiently  Published: August 2016  Author: Peter Geelen  Version: 2.0    Abstract  Whether you’re a first time user or a long-time expert, when you post a question to a forum, you expect an answer that is quick, to-the-point, and preferably customized to your needs.  But in lots of cases it actually takes a while (meaning additional questions and answers back and forth) before the community actually can get to the root cause of the issue.  I would like to provide you with practical hints and tips, and background information of them, to get the maximum result from your postings on the forum, with the minimum of effort.  If you ask the right question, the right way, you’ll get your answer much faster. |

Table of Contents

[Introduction 3](#_Toc457809221)

[Gathering the right data 3](#_Toc457809222)

[Important data components 3](#_Toc457809223)

[Basic ILM/MIIS server configuration 3](#_Toc457809224)

[SQL Server related configuration data 5](#_Toc457809225)

[Management agent related configuration 5](#_Toc457809226)

[Error codes and error conditions 5](#_Toc457809227)

[Troubleshooting rules extensions 6](#_Toc457809228)

[Information on configuration circumstances 11](#_Toc457809229)

[Optional data parameters 12](#_Toc457809230)

[System information you should not show at all 12](#_Toc457809231)

[Architecture and design questions 12](#_Toc457809232)

[Summary 13](#_Toc457809233)

[Check list 14](#_Toc457809234)

[Acronyms & abbreviations 15](#_Toc457809235)

# Introduction

What do you expect if you post a question on a forum?

A quick answer, solving the issue you’re struggling with, in a timely fashion?

However, the reality looks often different:

* Most of questions posted on the forum require some knowledge of your environment. To understand your question, people need background information.
* In many cases the initial question does not provide sufficient data to start helping.

What's the most efficient way to allow the community to help you?

The key to the quick solution is how you present the question.

In this article, I will give you suggestions and recommendations that will help you to get the most out of this forum.

# Gathering the right data

As a first step, you should gather all data that is related to your question. What the right data is depends on the type of answer you expect.

A general, more high-level question in conjunction with the architecture and the design requires a different approach than a practical question to troubleshoot an issue.

When troubleshooting your setup:

* Take time to **gather** **detailed data** before you post
* Provide **concise, basic information** about your environment
* Describe **error conditions in detail**

Even for more general architectural questions, take a moment and ask yourself:

* What is the situation you are in now?
* Where did you start from?
* What are you **trying to accomplish**?
* Do you have **examples** to help outline what you are aiming at?

# Important data components

In most cases, there are some core components you should address in your initial question. These components include:

* Basic FIM/MIM server configuration
* SQL server configuration
* Management agent information
* Error codes and error conditions

## Basic ILM/MIIS server configuration

The following list highlights important data points regarding your ILM/MIIS server configuration:

* **FIM/MIM** **build** version
* **SQL Server build** version
* SQL Server location (**local or remote**)
* **License type**: MSDN, volume, evaluation
* **Server type:** development, test, production

|  |
| --- |
| **Note** |
| You should include information about these components in all posts that are related to troubleshooting. |

### Retrieving the FIM/MIM build version

The **build version** of the platform is really essential, as specific builds solve specific known issues.

(Or cause specific issues for the pessimists among us.)

Just an illustration: the Microsoft Help and Support site is one of the sources that provides hotfixes and services packs. If you [check for the FIM/MIM hotfixes and service packs](http://support.microsoft.com/search/default.aspx?mode=r&query=identity+integration&spid=787&catalog=LCID%3D1033&1033comm=1&res=20), you’ll immediately notice the build numbers. Most of these articles also provide detailed information which issues are fixed.

The version number is pretty important and also easy to retrieve.

To get the version number, perform the following steps:

* Open the FIM/MIM **Identity Manager**
* To open the **About Identity Manager** dialog, click **About** in the **Help** menu.

The following screenshots show examples for these steps.

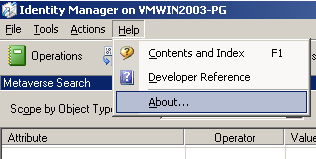


Fig. 1 Menu Help > About

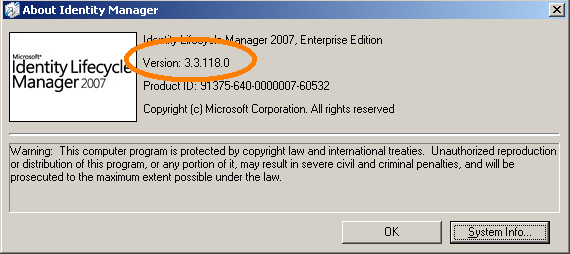


Fig. 2 ILM build number lookup

Carefully check the **license type** which you used to install your FIM/MIM with. It represents an important data point in the context of upgrading your system, as certain upgrade paths are not supported.

The **kind of server** (development, test or production) you’re working with influences the approach to troubleshooting. For example, on a development server, it usually doesn’t hurt to take drastic measures, like completely deleting data, cleaning connector spaces or cleaning the metaverse and start all over again.

However, in production, deleting a connector space with 100K or more objects shouldn’t happen without planning.

If your audience is aware of your server type, you will get better suited troubleshooting suggestions.

## SQL Server related configuration data

As the backend for the managed data, SQL server plays an essential role in your deployment.  
This is why you should include some information about your SQL server configuration into your post.

When you review some of the posts on this forum, you will find many questions that are related to the performance of an environment.   
For example, using FIM/MIM with a remote SQL server has a significant impact on the overall performance of a deployed solution.

Another reason to include information about your SQL setup is network and server security.

A configuration with a remote SQL server requires a slightly different approach, in comparison with SQL installed on the FIM/MIM server.

Typical SQL server parameters are:

* Remote connections
* SQL server build
* SQL server type
* SQL security permissions

When you post a question, at a minimum, provide these SQL server parameters:

* What is the **SQL Server build** version?
* How is SQL installed: **Locally** on the FIM/MIM server or **remotely**?
* **License type**: MSDN, volume, evaluation, production,…

## Management agent related configuration

Management agents are used to exchange data between FIM/MIM and connected data sources. The type of a management agent used in your environment has a big impact on the response you get.

When posting a question, you should address the following questions about your management agents:

* What are the **data sources** FIM/MIM is connected to?
* **What MAs** have you configured?
* What 3rd party MAs, XMAs have you installed?
* What is the **object count** in the MAs?
* What is the object count in the MV?

Information about your management agent configuration helps to get a global picture of the setup.

Again, the management agent configuration influences the kind of approach to take for a specific configuration.

If you get an error when configuring a MA, verify that the version of your data source is supported.

Check the FIM/MIM FAQ for more information, first.

The **MV and CS object count** is a practical indicator for the system load.

The ratio of the CS objects and the MV objects provides an overview of the join and projection rules.

## Error codes and error conditions

Error codes play an essential role in a troubleshooting scenario. However, in many posts I have seen, people tend to provide incomplete information about the errors that are documented by a system.

In conjunction with error codes, you should follow the following guidelines:

* Always include the **exact error messages** you get from the system.
* Drill down to the **detailed content** of the messages.
* Check the **event viewer** for related messages.
* Review logs created by extensions (if applicable).
* Provide log information details (verbose logging).

This is the easy part. Just make sure you catch enough details:

* Copy and paste the error messages in your post.
* Error codes, specific messages allow you to get results quickly.

|  |
| --- |
| **Note** |
| Additionally, this is exact the information you need to provide to Microsoft Product Support Services, in case you need official support. |

## Troubleshooting rules extensions

When troubleshooting your MA and MV extensions, make sure you’re compiling them in debug mode.

Make sure you enable the advanced build configurations in Visual Studio.

To enable the advanced build configuration in Visual Studio

* Start Visual Studio
* Open the **Options** dialog, by clicking **Tools** from the **Options** menu.
* In the tree view, click **General** in the **Projects and Solutions** node.
* Select ‘**Show advanced build configuration’**.

The following screenshot shows an example for this:

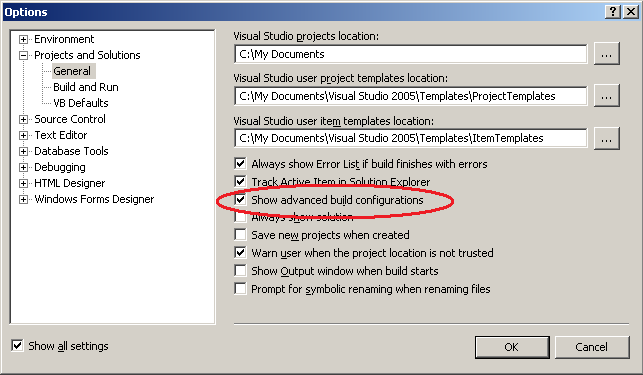


Fig. 3 Menu Tools > Options

When you enable the advanced build configurations, you have some interesting tools available, for example:

* Full debug information provides detailed error information when errors occur at run-time
* Breakpoints

I’ll discuss these options in the next section.

Full debug information provides detailed error information when errors occur at run-time.

To see full debug configuration information, you also need to compile the solution in debug mode.

To review the debug information:

* In Visual studio
  + Set the extension compilation to debug mode
  + Set the debug information generation to full
  + Save and recompile the solution
* In FIM/MIM
  + Click the extension errors in the management run profile details or run history.

### Compiling your VS extension in debug mode

To compile your solution in debug mode, open your extension in VS.NET, and check the solution properties. (Right click the solution in the VS Solution Explorer > select Properties).

Check the ‘Compile’ tab, option ‘Configuration’, as shown by the screenshot below.

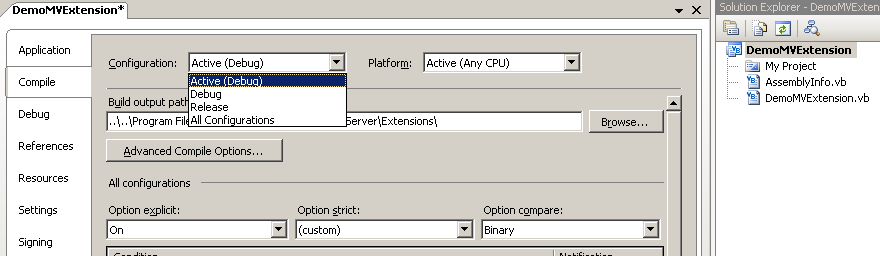


Fig. 4 Debug or Release mode

Debug mode and Release mode are different configurations for building your solution.

The **Release mode** optimizes the binary it produces (but optimizations can greatly complicate debugging), and generates no additional data to aid debugging.

As the name implies, you generally use the **Debug mode** for debugging your project, and the Release mode for the final build for FIM/MIM.

The debug mode disables optimizations and generates extra debug data.

When you run a program in Debug mode, Visual Studio enables you to use breakpoints to pause the program to examine the state of variables and objects.

### Enable generation of full debug information.

It’s not enough to enable the debug mode, you also need to enable the generation of full debug information. In that way, the error messages contain more details. And the more details you get, the better you can troubleshoot the extension.

In the screenshots below, I show you how to do so.

In the solution properties screen, select the ‘Advanced Compile options…’ button:

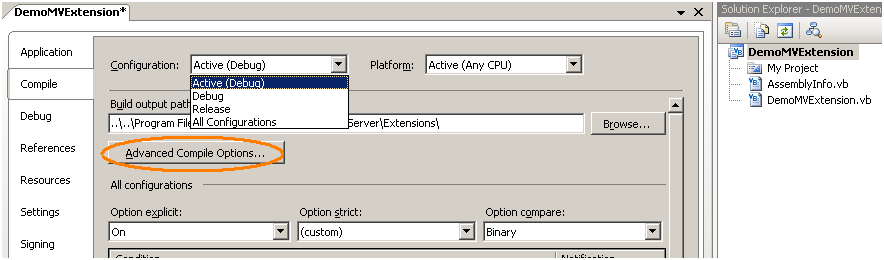


Fig. 5 Advanced Compile options

The following screenshot shows the options Visual Studio displays:

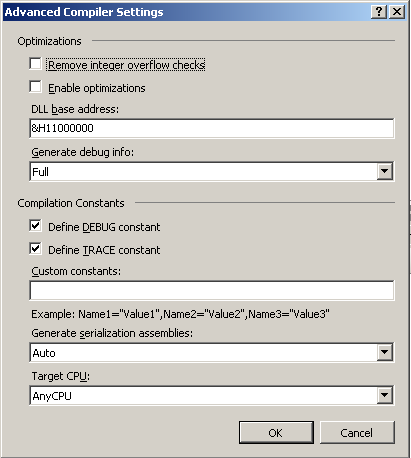


Fig. 6 Advanced Compile Options - Debug info

Make sure the ‘Generate debug info’ option is set to ‘Full’.

Save the extension and recompile it.

Let us take a look at the difference between the debug and release mode.

In the following example I’ve created a small FIM/MIM setup (source AD, target ADAM). Next I enabled provisioning in FIM/MIM and activated an empty template MV extension as created by default by FIM/MIM. After enabling the MV extension I ran an import and a synchronization cycle.

At the end of this step, I got an “extension-entry-point-not-implemented” error.

The following screenshots show an example for this.

### Example Debug mode vs. Release mode

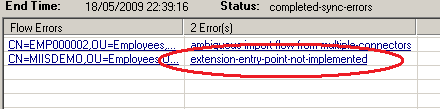


Fig. 7 Synchronization error

You can get more details about the error by clicking the link of the error message. The following screenshot shows the dialog that opens in this case.

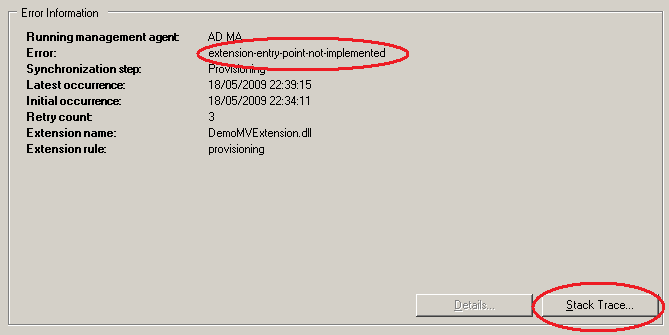


Fig. 8 Error detail

You can get to the root cause of an issue by clicking the **Stack Trace** button. Clicking this button, opens a dialog that contains the exact exception information including the line number where the error occurred in your code.

The following screenshot shows an example for this:

If you click the ‘Stack Trace…’ button, FIM/MIM shows the screen below:

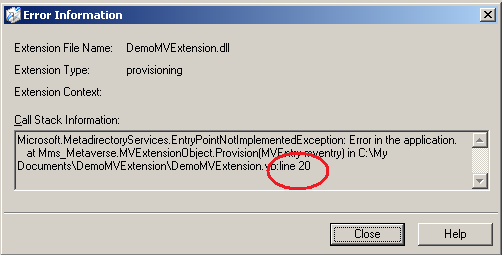


Fig. 9 Stack Trace

The Call Stack Information in the previous screenshot refers to line 20 in the extension source code.

The following screenshot shows the related source code:

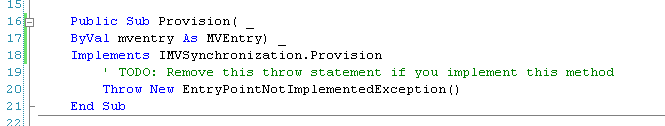


Fig. 10 Root Cause

As you can see, the code always throws an error because of the ‘Throw New …’ statement.

Now, let’s disable the debug information generation.

* In Visual studio
  + Set the extension compilation to release mode
  + Set the debug information generation to none
  + Save and recompile the solution
* In FIM/MIM
  + Click the extension errors in the management run profile details or run history.

The following screenshot shows an example for this.

Set the ‘Generate debug info’ to ‘None’.

.

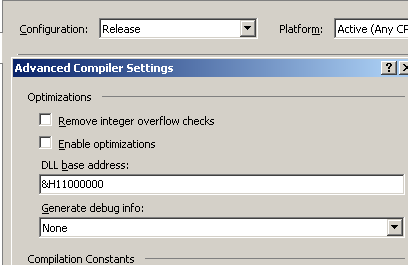


Fig. 11 Release mode

After saving the new settings, you need to recompile the extension.

When you run another synchronization run, you will not find detailed information about the error anymore.

The following screenshot shows an example of this.

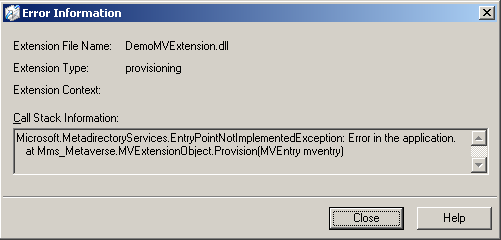


Fig. 12 No debug info in release mode

With the debug configuration disabled, the call stack information is very limited. FIM/MIM does not tell you why, when and where the error occurs.

|  |
| --- |
| **Important** |
| When you switch to release mode, remove the pdb debug file. |

## Information on configuration circumstances

In the previous section I’ve discussed components strictly related to the configuration basics of FIM/MIM components, which you can retrieve in real-time.

It’s clear that the complete FIM/MIM infrastructure exists of different parts, each depending on each other: the FIM/MIM application, the operating system, the SQL server, the .NET framework, the Visual Studio extensions, and more …

It’s a complex balance, even a small change in one of these components, might cause trouble.

Ideally you manage the complete FIM/MIM infrastructure yourself. But in my experience, this is rarely the case, as most live server setups are managed by a group of people.

And, not every member of the team knows the in and outs of every part.

So, when your FIM/MIM system starts behaving badly, just suddenly or intermittently, while it has been in operational mode for quite a while, it’s not always easy to determine the root cause.

When you’re troubleshooting your FIM/MIM setup, also take a wider look on the technical infrastructure.

I can guarantee it is worthwhile to log the changes you make on your system configuration from the very start.

Additionally, make sure the events and errors are logged accurately.

Change logs and events logs allow you to answer the following list of questions:

* Has there been a change to the system configuration recently, like an update/upgrade, hotfixes,...?
* **When** exactly does the error condition occur?
* Are you able to reproduce the error condition?
* Is the error occurring intermittently?
* Any particular configuration on your environment, like highly secured networking, server lockdown,…?

You should post this kind of information in your initial post, as it is very helpful.

# Optional data parameters

Another important type of information you should provide in your post, is NOT of a technical kind.

It’s not about the FIM/MIM configuration, neither.

It concerns the actions you have done when troubleshooting, before you posted your question.

### What kind of steps have you already done to get an answer?

* Which posts did you already look at?
* Did you already install certain hotfixes?
* Which run profiles did you run?
* …

|  |
| --- |
| **Note** |
| This kind of questions should allow the community to spend time on making progress instead of replying to questions already addressed previously. |

# System information you should not show at all

It might be obvious, but it wouldn’t be the first time someone posted personal, secret, identifiable data on the web. It would not take too much effort to use it maliciously.

Make sure you mask:

* Passwords and secrets
* Personal identifiable information
* Live machine and domain info
* Service Account names
* IP addressing
* …

# Architecture and design questions

The forum is an excellent resource if you just need some assistance solving a problem on architectural level.

The following list of hints and questions should allow you to provide the basics in your post:

* Provide some basic data on your setup.
* What did you implement already?
* Describe what you actually aiming at. What are you trying to achieve?

Provide as much detail as possible to make your case with examples and/or (anonymized) screenshots.

# Summary

When troubleshooting your setup:

* Take time to gather sufficient and **detailed** info.
* Provide **basic information** on your **environment.**
* Be specific, describe the **error conditions** in detail.

When you need help or an insight on a design issue:

* Provide some basic data on your setup.
* Describe what you actually aiming at. What are you trying to achieve?

In both cases provide as much detail as possible to make your case.

Providing examples or (a link to) screenshots, help to understand.

# Check list

The following check list summarizes the topics I have discussed in this article and should help you to make sure that you include all details that are relevant to your issue.

|  |  |  |
| --- | --- | --- |
|  | **1. Basic Server info (general)** | **Results / notes / remarks** |
| ❑ | - FIM/MIM build version |  |
| ❑ | - SQL Server build version |  |
| ❑ | - SQL installed locally or remotely? |  |
| ❑ | - License type (MSDN, eval, production, …) |  |
|  |  |  |
|  | **2. Specific server info (details on configuration)** |  |
| ❑ | - Which MA's configured? |  |
| ❑ | - MA & MV object count? |  |
| ❑ | - Particular attributes, attribute flows, extensions? |  |
| ❑ | - Provide code bits of your extensions  (when applicable) |  |
|  |  |  |
|  | **3. Error conditions** |  |
| ❑ | - Provide **exact** error codes\* |  |
| ❑ | - Drill down to the **detailed content** of the error message\* |  |
| ❑ | - Check the **event viewer** |  |
| ❑ | - If applicable, check logs created by extensions or services (e.g. PCNS) |  |
|  | \*(HINT: copy/paste info into post) |  |
|  |  |  |
|  | **4. Circumstances & particular details** |  |
| ❑ | - Any particular configuration on your setup? |  |
| ❑ | - Particular conditions or features? |  |
| ❑ | - Has the system **changed** recently? Like update/upgrade, hotfixes,...? |  |
| ❑ | - When exactly does the error condition occur? |  |
| ❑ | - Are you able to **reproduce** the error condition? |  |
| ❑ | - Is the error occurring intermittently? |  |
|  |  |  |
|  | **5. What actions have you already performed?** |  |
| ❑ | - Have you already followed certain procedures? |  |
| ❑ | - Any articles or posts you looked at? |  |
| ❑ | - Results achieved so far? Details? |  |

# Acronyms & abbreviations

|  |  |
| --- | --- |
| **Code** | **Description** |
| ADAM | Active Directory Application mode (=ADLDS) |
| ADLDS | Active Directory Lightweight Directory Services (aka ADAM) |
| CS | (FIM/MIM) Connector Space |
| FIM | Forefront Identity Manager 2010 |
| GALSync | Global Address List Synchronization |
| IIFP | (MIIS) Identity Integration Feature Pack (retired) |
| ILM | Identity Lifecycle Manager |
| MA | (FIM/MIM) management agent |
| MIIS | Microsoft Identity Integration Server 2003 |
| MIM | Microsoft Identity Manager 2016 |
| MSDN | Microsoft Developer Network |
| MV | (FIM/MIM) Metaverse |
| RC | Release Candidate |
| XMA | (FIM/MIM) Extensible Management Agent |

# 