xMII UDS Overview and Troubleshooting

Applies to:

SAP xMII Universal Data Servers (UDS) version 4.0 and higher.

Summary

This article describes the SAP xMII UDS line of products and methods for debugging issues related to the product.

Author(s): Martin Kreibe

Company: SAP Labs, LLC

Created on: 7 February 2007

Author Bio

Martin Kreibe is a Software Developer for SAP Labs. Martin is currently part of the xMII Connectivity Team and has worked on the xMII UDS project since 2004, where he participated in a leading role in the architectural, design and development stages of the project.

Table of Contents

Overview of the SAP xMII UDS Architecture

Why are the xMII UDSs Needed?

xMII uses many mechanisms for extracting data from 3rd party applications, some examples of these "Data Servers" are the JDBC, OLAP and XML connectors. This article deals with a specific type of connectivity called the xMII UDSs. The xMII UDSs are unique to the other xMII Data Servers because they connect to the 3rd party systems by using APIs and other tools developed by the 3rd party vendors and require customized applications to extract this data, also the xMII UDSs are a COM based application architecture and can run on a different machine then the xMII Server.

As of the publication of this article there are the following xMII UDSs available:

- SAP xMII OLE DB UDS
- SAP xMII OPC DA UDS
- SAP xMII OPC HDA UDS
- SAP xMII InTouch UDS
- SAP xMII RSView UDS
- SAP xMII Citect UDS
- SAP xMII Proficy Historian UDS
- SAP xMII Proficy iFIX UDS
- SAP xMII PI UDS

What are the System Requirements for xMII UDSs?

The xMII UDS product runs on Windows 2000 Server Service Pack 2 or greater. Though the xMII UDS product can run on Windows XP it is not supported for production usage due to the Operating Systems networking limitations.

How do the xMII UDSs Work?

The xMII UDSs have three major components that work in conjunction to deliver the functionality required to extract third party data from the back end data sources. The first element is the xMIIUDSAdminConsole.exe application which is used to configure the xMII UDS instances. Next the xMIIUDSHost.exe application is the host for the xMII UDS instance. Last the xMII UDS, which itself is a COM DLL, is the particular logic needed to extract data from the specific back end data source.
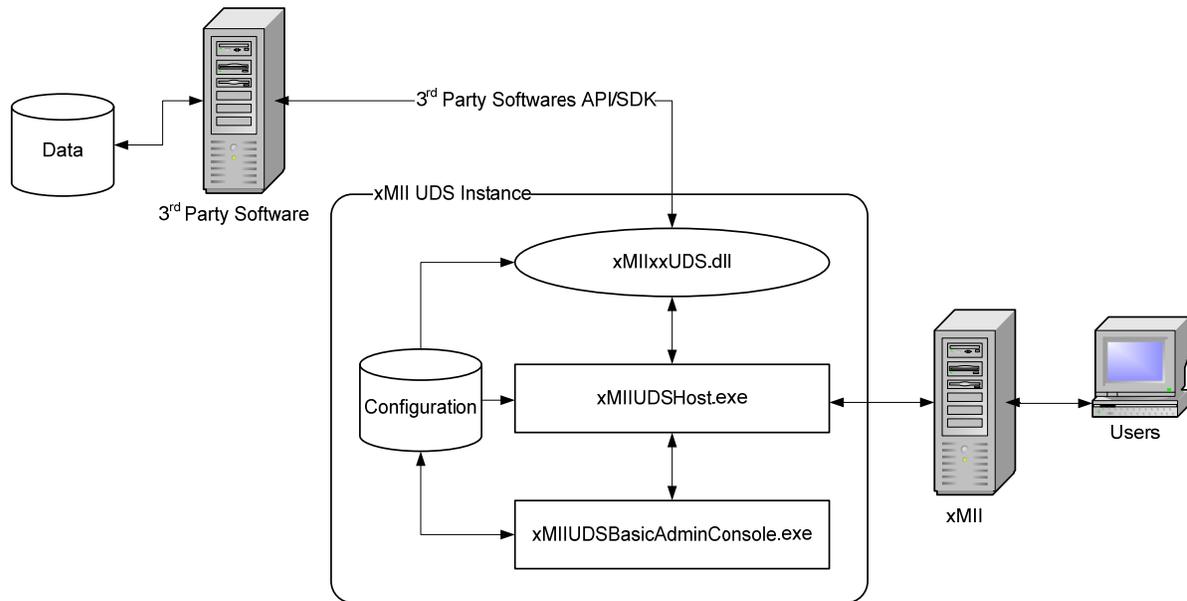
What Data do the xMII UDSs Expose?

The data exposed to xMII from the xMII UDSs is dependent on the xMII UDS. The xMII UDSs typically retrieve information from database and tag (current, historical and aggregated) data sources.

What is an xMII UDS Instance?

An xMII UDS Instance is the particular configuration and xMII UDS needed to extract data from a given back end data source. To be more specific, an xMII UDS Instance is an instance of the xMIIUDSHost.exe application with a given name that is associated with a configuration setting.
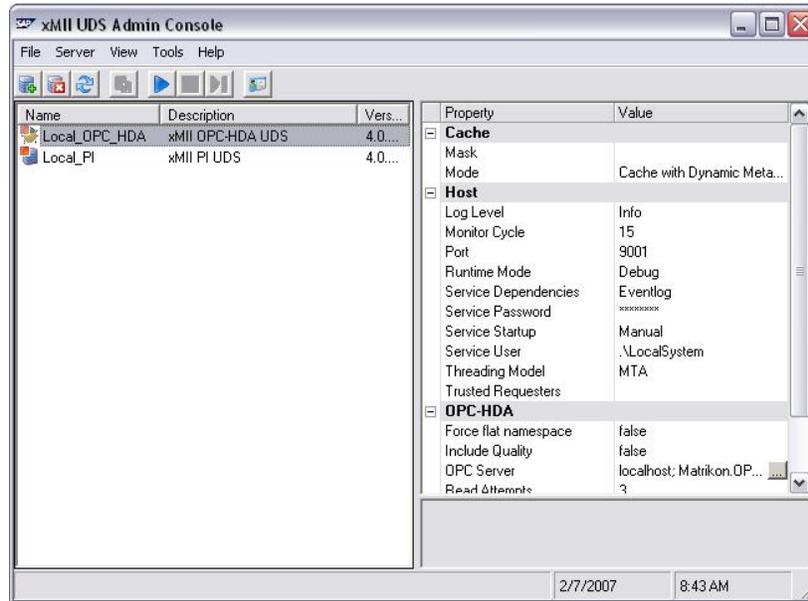

Figure 1: Admin Console Instances

The 'Local_OPC_HDA' entry under the Name column of the Admin Console is an example of a UDS instance.

Note: The 'Port' setting under the '**Host**' aspect is the TCP/IP port that the xMII UDS Instance will listen on, waiting for incoming xMII Server requests. It is not a setting for communicating to the backend.

Configuration Related Issues

How do the xMII UDS Instances Run?

There are three runtime modes that an xMII UDS Instance can run under. Each mode has its advantages and some back end data sources may not communicate with the xMII UDS correctly under some modes. The first is the "Exe" mode. This mode allows for the program (though hidden) to run under the current users account. "Service" mode allows the Windows Service system to manage the actual process. Service mode will allow you to manipulate the user credentials and other process parameters, including the ability to launch the application when the machine boots up. The last mode is a "Debug" mode. The debug mode is not supported in production environments and is only intended to help with troubleshooting runtime related issues.

Note that some UDSs can only be run in "Exe" mode like RSView and InTouch. Refer the UDS help documentation for more information of which xMII UDSs need to be run in "Exe" mode.

For ease of maintenance of the xMII UDS Instances, it is recommended, if the xMII UDS supports it, for the instances to be set up as a "Service" with proper settings and credentials. So that if the machine is restarted, the instances will correctly startup and connect without any administrator/user interaction.

How can an xMII UDS Instance Run as a Windows Service?

The Admin Console has many options available to configure the xMII UDS Instance for Service Mode. First the Runtime Mode in the Host aspect should be selected, this will tell the instance it is going to run as a service. Other properties allow for enhanced modes of operation like user name and password.

Note that the service name in the 'Services' Administration tool will be in the following format:

xMII UDS Instance (<xMII UDS Instance Name>)

What is DCOM and do the xMII UDSs use it?

DCOM stands for Distributed COM or Distributed Component Object Model. For more information on DCOM please refer to Microsoft's MSDN Articles on DCOM.

The xMII UDSs do *not* expose interfaces to DCOM. That said, the xMII UDSs do use DCOM to connect to some of the back end data sources like OPC DA and OPC HDA. In order to get some of these interfaces to work correctly you may have to modify the security settings to allow for the xMII UDS to connect.

Modifying DCOM Settings.

The Component Services MMC snap is used to configure DCOM settings (this is usually found in the "Control Panel > Administrative Tools" Folder). Look for the desired Application or Application ID. Make sure the xMII UDSs User has the ability to launch and access the Applications.

Accessing xMII UDS Logs and Cache Snapshots

Where are the xMII UDS Logs?

All log files are stored near the application installation directory in a folder called "Logs". All log files are named as follows.

For xMII UDS Instances:

<xMII UDS Instance Name>_<4 Digit Year>_<2 Digit Month>_<2 Digit Day>.log

For the xMII UDS Admin Console:

xMII UDS Admin Console_<4 Digit Year>_<2 Digit Month>_<2 Digit Day>.log

What about Startup logs?

There is an extremely small possibility that the UDS Instance will not startup and no log files are generated by the errors. Make sure that your configuration is correct before contacting technical support for further assistance.

In these cases it is still possible to see where the process failed. The xMII UDSs use that Debug Stream to log startup issues if the "-debug" flag is added to the command line. In order to use the debug flag the instance must be run in Exe mode. The process then must be started manually by starting a command line, navigating to the installation location of the xMII UDSs and the "xMIIUDSHost.exe" application. Following is the command line arguments for the "xMIIUDSHost.exe" application:

"xMIIUDSHost.exe" <xMII UDS Instance Name> [-debug]

Be sure to include the ".exe" part of the application name.

To read the Debug Stream a Debug Stream Reader must be used. An example of a Debug Stream Reader is DebugView 4.6.

Where are the Cache Snapshot Files?

All tag based xMII UDS Instances have an internal cache to help speed up runtime performance. The cache is loaded at the startup of the xMII UDS Instance and consists of the Alias file (if the Cache Mode is set to "Alias") or the entire 3rd parties namespace (if the Cache Mode is set to "In-Memory").

For those xMII UDSs that have a caching mechanism, it is possible to take a snapshot of the cache upon startup of the xMII UDS Instance. The snapshot files can be retrieved by using the debug flag [-debug] as described above. All cache snapshot files are located in the Logs directory under the installation location. The cache snapshot file names will be as follows:

<xMII UDS Instance Name>_<4 Digit Year>_<Month Name>_<2 Digit Day>[<2 Digit Hour [0-23]>_<2 Digit Minute>_<2 Digit Second>]_Cache.xml

Copyright