

# PROGRAMMATICALLY STARTING AND STOPPING AN SAP XMII UDS EXECUTABLE INSTANCE

## Summary

Some data sources run as executable programs which is true of most SCADA packages. In order for an SAP xMII UDS instance to connect to one of these data sources, the data source must be started prior to the SAP xMII UDS. This document describes several strategies for starting and stopping SAP xMII UDS instances without using the Admin Console.

**Created on:** 17 October 2006

## Author Bio



Rob Hallquist is an SAP Labs developer on the SAP xMII UDS development team. He has over ten years of experience designing and developing Microsoft-based applications targeted for the manufacturing plant floor.

## Table of Contents

Audience .....	3
Overview .....	3
Starting a UDS instance from a command window .....	4
Example .....	4
Starting a UDS instance from a Command Script or Batch File.....	5
Example .....	5
Starting/Stopping a UDS instance from a VB 6.0 application.....	6
Example .....	6
Starting/Stopping a UDS instance from a VBScript.....	8
Example 1 – Start UDS instance .....	8
Example 2 – Stop UDS instance.....	9
Starting/Stopping a UDS instance from a .NET application .....	10
Example 1 – C#.....	10
Example 2-VB.NET .....	11
Starting/Stopping a UDS instance from Wonderware InTouch .....	12
Example 1 – Start UDS instance .....	12
Example 2 – Stop UDS instance.....	12
Starting/Stopping a UDS instance from Rockwell RSView .....	13
Example 1 – Start UDS instance .....	13
Example 2 – Stop UDS instance.....	13
Copyright.....	14

## Audience

It is assumed that anyone reading this article is familiar with the Microsoft Windows operating system. It is also assumed that anyone attempting the examples listed have the correct background experience with the software used by the examples.

## Overview

The following documentation describes several different methods of starting and stopping an SAP xMII UDS instance. While it is possible to start an SAP xMII UDS instance that has its "Runtime Mode" property set to something other than "Exe", it does not make much sense. If you want to control what services an SAP xMII UDS instance requires before startup when its "Runtime Mode" is set to "Service", you would edit the "Service Dependencies" section. Therefore, this document will assume that all SAP xMII UDS instances that are to be started or stopped from an external program have their "Runtime Mode" property set to "Exe."

**NOTE:** For all examples the SAP xMII UDS instance name is case sensitive.

The SAP xMII UDSs do not support a built in shutdown feature without issuing a URL command via SAP xMII. Therefore a separate COM dll is included in this document to allow the programmatic shutdown of an SAP xMII UDS instance. In order to use this COM dll, it must be copied into the installation folder of the SAP xMII UDS which is usually C:\Program Files\SAP\xMII\UDS. Then the COM dll must be registered. To register the shutdown COM dll, do the following:

1. Open a Windows command prompt and change to the folder where the attached xMIIUDSShutdown.dll file is located.
2. Enter the following command:  
`regsvr32 xMIIUDSShutdown.dll`



xMIIUDSShutdown.zip

## Starting a UDS instance from a command window

The most basic way of starting an application is by use of a Windows command window. Almost any application can be started in this manner and the SAP xMII UDS instances are no different. NOTE: It is not possible to stop an SAP xMII UDS instance from the command window.

### Example

The following example will start an SAP xMII UDS instance named "MyUDS" from a Windows command line:

1. Make sure that the data source the SAP xMII UDS instance is using is started.
2. Open a Windows command window. There are several ways to open a command window, but the easiest is to select Start | Run from the Windows task bar, type "cmd" and click OK.
3. Change to the folder where the SAP xMII UDS is installed. If the SAP xMII UDS was installed to the default location type in the following:  

```
cd C:\Program Files\SAP\xMII\UDS
```
4. Type the command that will run the "MyUDS" instance (note: the quotations must be included):  

```
"xMIIUDSHost.exe" MyUDS
```
5. You can verify that the SAP xMII UDS instance is running by opening the Admin Console and viewing the runtime status.

## Starting a UDS instance from a Command Script or Batch File

Another popular method of running programs is by using a Windows NT Command Script. The example below shows how to start an SAP xMII UDS instance named "MyUDS" if the SAP xMII UDS has been installed in the default folder. NOTE: It is not possible to stop an SAP xMII UDS from a command script or batch file.

### Example

1. Create a text document named StartMyUDS.cmd.
2. Enter the following text (note: the quotations must be included):  
`"c:\Program Files\SAP\xMII\UDS\xMIIUDSHost.exe" MyUDS`
3. Save and run the script.
4. You can verify that the SAP xMII UDS instance is running by opening the Admin Console and viewing the runtime status.

## Starting/Stopping a UDS instance from a VB 6.0 application

There are a lot of Microsoft Visual Basic 6.0 applications and application developers that work in the manufacturing field. The following example will illustrate how to start and stop an SAP xMII UDS instance using this development tool. **NOTE:** In order to access the xMIIUDSShutdown.dll, you must add a reference to it in VB. Refer to the Microsoft Visual Basic 6.0 documentation on how to add a reference to a COM dll.

### Example

First create a form that is similar to the following:



Where the attributes of the objects on the form are:

Object	Caption	Name
Form	UDS Start/Stop Example	Form1
TextBox		Instance
Button	Shutdown	Shutdown
Button	Start	Start

Paste the following text into the form:

```
Private Sub Shutdown_Click()  
On Error GoTo catch  
  
    Dim UDSShutdown As New xMIUOSShutdown.xMIUOSStopInstance  
    UDSShutdown.ShutdownInstance (Instance.Text)  
  
Exit Sub  
catch:  
    MsgBox Err.Description  
End Sub  
Private Sub Start_Click()  
On Error GoTo catch  
    Dim dbl As Double  
    Dim CommandLine As String  
  
    CommandLine = Chr(34) & "C:\Program Files\SAP\xMII\UDS\xMIUOSSHost.exe"  
& Chr(34) & " " & Instance.Text  
    dbl = Shell(CommandLine, vbHide)  
    If (dbl = 0) Then  
        MsgBox "Could not start UDS instance"  
    End If  
  
Exit Sub  
catch:  
    MsgBox Err.Description  
End Sub
```

## Starting/Stopping a UDS instance from a VBScript

Many different applications as well as the Windows scripting host allow running VBScript commands. The following example describes how to create a VBScript that will start an SAP xMII UDS instance.

### Example 1 – Start UDS instance

The following script will start a named SAP xMII UDS instance from the command line:

```
option explicit

Dim CommandLine
Dim RetCode
Dim WshShell

'Make sure that the UDS instance name is provided
If Wscript.Arguments.Count = 0 Then
    Wscript.Echo "Too few arguments. Please specify an SAP xMII UDS
instance to start."
Else
    CommandLine = Chr(34) &
"C:\Perforce\xMII\LHDSFramework\dev\bin\debug\xMIIUDSHost.exe" & Chr(34) &
Wscript.Arguments(0)
    'Create the scripting shell object
    set WshShell = WScript.CreateObject("wscript.shell")
    'Start the uds instance, hide the window, wait for the function to return
    RetCode = WshShell.run (CommandLine, 0, TRUE)
    If (RetCode = 0) Then
        Wscript.Echo "Could not start SAP xMII UDS '" & Wscript.Arguments(0)
& "' "
    End If
End If
```

## Example 2 – Stop UDS instance

The following script will stop a running SAP xMII UDS instance from the command line:

```
option explicit

Dim WshShell
Dim UDSShutdown

'Make sure that the UDS instance name is provided
If Wscript.Arguments.Count = 0 Then
    Wscript.Echo "Too few arguments. Please specify an SAP xMII UDS
instance to stop."
Else
    'Create the SAP xMII UDS Shutdown object
    set UDSShutdown =
WScript.CreateObject("xMIIUDSShutdown.xMIIUDSStopInstance.1")
    UDSShutdown.ShutdownInstance(Wscript.Arguments(0))
End If
```

## Starting/Stopping a UDS instance from a .NET application

It may be desirable to create a program in another language to start an SAP xMII UDS instance. Here is an example of code that will run an SAP xMII UDS instance named "MyUDS" if the SAP xMII UDS is installed in the default location. **NOTE:** For each of the .NET examples, the xMIIUDSShutdown.dll must be added as a reference. Refer to the appropriate Microsoft documentation for information on how to add a COM reference.

### Example 1 – C#

```
using System.Diagnostics;
using System.Threading;

namespace UDSExample
{
    class Program
    {
        private static void StartUDS(string instanceName)
        {
            Process MyUDSProcess = Process.Start(
                @"C:\Program Files\SAP\xMII\UDS\xMIIUDSHost.exe",
                instanceName);
        }

        private static void StopUDS(string instanceName)
        {
            xMIIUDSShutdown.xMIIUDSStopInstanceClass UDSShutdown = new
xMIIUDSShutdown.xMIIUDSStopInstanceClass();
            UDSShutdown.ShutdownInstance(ref instanceName);
        }

        static void Main(string[] args)
        {
            string instanceName = "MyUDS";
            StartUDS(instanceName);
            //Wait for the UDS to start.
            //You would not put a sleep in your code, but it is here
            //to simulate that the server will be running for quite
            //some time before it is shutdown.
            Thread.Sleep(5000);
            StopUDS(instanceName);
        }
    }
}
```

## Example 2-VB.NET

```
Imports System.Diagnostics
Imports System.Threading
Module Module1
    Private Sub StartUDS(ByVal instanceName As String)
        Dim MyUDSProcess As Process
        MyUDSProcess = Process.Start( _
            "C:\Program Files\SAP\xMII\UDS\xMIIUDSHost.exe", _
            instanceName)
    End Sub
    Private Sub StopUDS(ByVal instanceName As String)
        Dim UDSShutdown As New xMIIUDSShutdown.xMIIUDSStopInstance()
        UDSShutdown.ShutdownInstance(instanceName)
    End Sub
    Sub Main()
        Dim instanceName As String = "MyUDS"
        StartUDS(instanceName)
        'Wait for the UDS to start.
        'You would not put a sleep in your code, but it is here
        'to simulate that the server will be running for quite
        'some time before it is shutdown.
        Thread.Sleep(5000)
        StopUDS(instanceName)
    End Sub
End Module
```

## Starting/Stopping a UDS instance from Wonderware InTouch

This example assumes that the InTouch application is running as an executable and not a Windows service. In order to access data from Wonderware InTouch via an SAP xMII UDS instance, an InTouch application must be running first. It is possible to start and shutdown an SAP xMII UDS instance when the InTouch application starts and exits by following the examples below:

### Example 1 – Start UDS instance

1. Create an On Startup application script within InTouch.
2. Add the following text to the script (note that there must be a space before the MyUDS UDS name.):

```
StartApp StringChar(34) + "C:\Program Files\SAP\xMII\UDS\xMIIUDSHost.exe" +  
StringChar(34) + " MyUDS";
```

### Example 2 – Stop UDS instance

1. Create an On Shutdown Application script within InTouch.
2. Add the following text to the script:  

```
OLE_CreateObject(%UDSShutdown, "xMIIUDSShutdown.xMIIUDSStopInstance.1");  
%UDSShutdown.ShutdownInstance("MyUDS");  
OLE_ReleaseObject(%UDSShutdown);
```

## Starting/Stopping a UDS instance from Rockwell RSView

This example assumes that the RSView application will be running as an executable and not a Windows service. In order to access data from RSView application via an SAP xMII UDS instance, an RSView application must be running first. It is possible to start and shutdown an SAP xMII UDS instance when the RSView application starts and stops by following the examples below:

### Example 1 – Start UDS instance

1. Create a Macro and add the following text to the macro:  
`AppStart "C:\Program Files\SAP\xMII\UDS\xMIIUDSHost.exe" MyUDS`
2. In the RSView startup macro add the name of the macro created above so that the UDS startup macro above will be started when the RSView application is started.

### Example 2 – Stop UDS instance

1. Create the following subroutine in the ThisProject VBA object:

```
Sub ShutdownUDS()  
On Error GoTo catch  
    Dim InstanceName As String  
    InstanceName = "MyUDS"  
    Dim UDSShutdown As Object  
    Set UDSShutdown =  
CreateObject("xMIIUDSShutdown.xMIIUDSStopInstance.1")  
    UDSShutdown.ShutdownInstance (InstanceName)  
    Set UDSShutdown = Nothing  
Exit Sub  
catch:  
    gActivity.Log "RSView32 VBA Error - Could not shut down UDS instance  
[" & InstanceName & "] ->" & Err.Description, roActivityError  
End Sub
```
2. Create a macro and add the following text to the macro:  
`VBAExec ShutdownUDS`
3. In the shutdown macro, add the name of the macro created above which will be called during the shutdown of RSView and will shutdown the SAP xMII UDS instance.

## Copyright

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

Rockwell and RSView are registered trademarks of Rockwell Automation, Inc.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves information purposes only. National product specifications may vary.

Wonderware and InTouch are registered trademarks of Invensys Systems, Inc.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.