Quick Guide for MTScaleKey

Quick Guide

July 2019



Contents

1	Wна	т is Scale Кеу	. 1
2	Wнy	DO WE NEED SCALE KEY	. 1
3	How	TO USE SCALE KEY	. 2
-	3.1	Installation	. 2
	3.2	Usaae	.3
	3.2.1	PLU Transmission	3
	3.2.2	Get Transaction Logs	4

1 What is Scale Key

Scale Key is a middle ware for communication between Mettler-Toledo scales and 3rd party software. Current version of Scale Key supports bPlus, FreshBase, RL00/RL00+/8442, bCom, bPro, bMobile, i15, bTwin, etc.



2 Why do we need Scale Key

From the 3rd party software provider's point of view, they only need to know the interfaces provided by Scale Key and do not need to be aware of the difference of each scales. This may save much integration effort.

Scale Key will keep the interfaces unchanged. This means the 3rd party software does not need to change even if MT scales are updated and have communication structure change.

Scale Key is designed to support all functionalities of legacy MT middle ware. It is the only official supported middle ware of MT. The 3rd party provider only need integrate with Scale Key.

Scale Key is optimized for the performance and it also provides debugging tool, which is helpful for the 3rd party software provider to root case issues.

3 How to use Scale Key

(x86)\ScaleKey\MTScaleAPI

i.

3.1 Installation

In Scale Key release, there is a <u>Lib</u> subfolder, as shown in below snapshot. There are a set of libraries files in this folder. They can be copied to any preferred folder for code-level integration, no installation in this case. The libraries export APIs for 3rd party software to invoke. Details are introduced in section 3.2.1.

y :	> 1.26.0710.00_AK > Setup > Windows	> Lib → Win32			
1	各称 ~	修改日期	类型	大小	
	S CheckOnline.dll	2019-07-10 15:51	应用程序扩展	267 KB	
	DeviceExplore.dll	2019-07-10 15:51	应用程序扩展	185 KB	
	Framework.dll	2019-07-10 15:51	应用程序扩展	664 KB	
	🗟 mfc100.dll	2019-07-10 15:51	应用程序扩展	4,295 KB	
	🗟 msvcp100.dll	2019-07-10 15:51	应用程序扩展	412 KB	
	🗟 msvcr100.dll	2019-07-10 15:51	应用程序扩展	756 KB	
	MTCommonScaleAPI.dll	2019-07-10 15:51	应用程序扩展	823 KB	
	MTConfiguration.dll	2019-07-10 15:51	应用程序扩展	72 KB	
	MTEncrypt.dll	2019-07-10 15:51	应用程序扩展	560 KB	
	MTLogger.dll	2019-07-10 15:51	应用程序扩展	187 KB	
	MTScaleAPI.dll	2019-07-10 15:51	应用程序扩展	134 KB	
	MTScalebCom.dll	2019-07-10 15:51	应用程序扩展	522 KB	
	MTScalebTwin.dll	2019-07-10 15:51	应用程序扩展	341 KB	
	MTScaleCaesar.dll	2019-07-10 15:51	应用程序扩展	819 KB	
	MTScale115.dll	2019-07-10 15:51	应用程序扩展	420 KB	
	MTScaleKey.config	2019-07-10 15:51	XML Configurati	1 KB	
	MTScaleKeyData.dll	2019-07-10 15:51	应用程序扩展	429 KB	
	MTScaleRL00.dll	2019-07-10 15:51	应用程序扩展	751 KB	
	MTSTDData.dll	2019-07-10 15:51	应用程序扩展	2,330 KB	
	MTXML.dll	2019-07-10 15:51	应用程序扩展	1,189 KB	
	MTZip.dll	2019-07-10 15:51	应用程序扩展	130 KB	
	Intermediate Platform.dll	2019-07-10 15:51	应用程序扩展	326 KB	
1	🔏 TestDll.exe	2019-07-10 15:51	应用程序	205 KB	
	🗟 TransferData.dll	2019-07-10 15:51	应用程序扩展	356 KB	
	TransferEth.dll	2019-07-10 15:51	应用程序扩展	242 KB	

In the <u>Setup</u> older, there are installation packages. After installation, there are services automatically running to really get connections from scales (for TLog, refer section 3.2.2). A subfolder named <u>MTScaleAPI</u> will be placed in the installation folder, which also contains the libraries files, which can be copied to destination folder as you wish.

A			
各称	修改日期	类型	大小
Framework.dll	2019-03-26 13:10	应用程序扩展	650 KI
msvcp100.dll	2018-12-10 10:44	应用程序扩展	412 K
imsvcr100.dll	2018-12-10 10:44	应用程序扩展	756 KI
MTCommonScaleAPI.dll	2019-03-26 13:17	应用程序扩展	823 Ki
MTConfiguration.dll	2019-03-20 11:02	应用程序扩展	72 Ki
MTEncrypt.dll	2019-03-20 11:02	应用程序扩展	560 Ki
🚳 MTLogger.dll	2019-03-20 11:02	应用程序扩展	187 Ki
MTScaleAPI.dll	2019-03-26 13:17	应用程序扩展	134 K
MTScaleAPI.h	2018-12-10 10:43	UltraEdit Docum	8 KI
MTScaleAPI.lib	2019-03-26 13:17	LIB 文件	279 Ki
Directory MTScaleKey.config	2018-12-10 10:44	XML Configurati	1 K
MTScaleKeyData.dll	2019-03-26 13:09	应用程序扩展	429 K
MTSTDData.dll	2019-03-20 11:09	应用程序扩展	2,330 KI
MTXML.dll	2019-03-20 11:02	应用程序扩展	1,189 Ki
MTZip.dll	2019-03-20 11:01	应用程序扩展	130 Ki
Platform.dll	2019-03-20 11:01	应用程序扩展	326 K
🙈 TestDil.exe	2019-03-26 13:17	应用程序	204 KI
TransferEth.dll	2019-03-26 13:17	应用程序扩展	242 K

3.2 Usage

3.2.1 PLU Transmission

Step 1. Prepare PLU data. For example, if we have below PLU data to transfer to scales,

PLU Number	Name	Article Number	Unit of Measure	Unit Price	Label Number	ET Number
1	Apple	1	KGM	11.3	1	1
2	Banana	2	PCS	20	1	2

We need have a XML file to represent above data, as shown in <u>Data1.xml</u>. (IMPORTANT: This and other files as the input to Scale Key should be in UTF-8 format.)

Step 2. Create another XML file to describe the operation. In this example, we want to "update and transmit" the PLU data in Data1.xml. The operation maps <u>Command1.xml</u>.

Step 3. Create the third XML file to list destination scales. If we have below 2 scales

Device Number	Scale Number	Scale Type	IP	Port	Communication Type
1	1	bCom	172.30.8.82	3001	Ethernet
2	2	bPlus	172.30.8.85	2305	Ethernet

The corresponding XML file is shown as DeviceList.xml.

Step 4. When above 3 XML files are ready, we need the final XML file to describe the task, as shown in <u>Task.xml</u>. All the 4 XML files must be saved in the same folder.

Step 5. Now we can invoke the ExecuteTaskInFile API in MTScaleAPI.dll. This DLL contains all external interfaces of Scale Key. The declaration of the API is:

-ANSI version:

extern "C" bool __stdcall ExecuteTaskInFile(const char* szTaskID, const char* szInputFile, const char* szOutputFile, bool bSynch)

-UNICODE version:

extern "C" bool __stdcall ExecuteTaskInFileW(const wchar_t* szTaskID, const wchar_t* szInputFile, const wchar_t* szOutputFile, bool bSynch)

Parameters of the APIs are:

szTaskID: The unique task ID

szInputFile: The full path of the task file (above Task.xml)

szOutputFile: The full path of the result file

bSynch: Execute way, true is synchronized, and false is asynchronous.

The API returns a boolean value to indication if it succeeds.

Here we have a sample code in C# to invoke the API, referring TaskExecute.cs.

Step 6. To query the task execution status, we can invoke the *QueryTask* API in *MTScaleAPI.dll*. The declaration of the API is:

-ANSI version:

extern "C" char *__stdcall QueryTask(const char* szInput)

-UNICODE version:

extern "C" wchar_t *__stdcall QueryTaskW(const wchar_t* szInput)

Parameters of the APIs are:

szInput: XML represented in string

The API returns a string of XML to represent the task's status. If to query multiple tasks in batch, only status of existing tasks are returned.

Below is an example of input XML string:

<MTTask>

<TaskID>20160222111817081</TaskID> <!-- Task ID -->

<TaskType>98</TaskType> <!-- "98" means to query task status -->

</MTTask>

Sample code can also be found in TaskExecute.cs.

If *TaskStatus* in the return XML string is Complete, it means the task is finished. A sample return XML can be found in <u>TaskResult.xml</u>.



You can read more from the MTScaleKey SDK document.

3.2.2 Get Transaction Logs

If you need receive transaction logs (TLog) from scales, you should install Scale Key other than only copying the DLLs (refer section 3.1). After installing, there is a service automatically running to receive TLogs. There are 3 steps to receive TLogs with Scale Key.

Step 1. Setup scales to make them automatically upload TLog.

There are two aspects of settings. You can find details in the manual of scales or ask for service technicians.

- Enable "save TLog and upload".
- Set the same server IP and port value as what in Scale Key.

Step 2. Create database tables.

There is a folder named "Database" in Scale Key installation folder, which contains scripts to create the TLog tables for different databases. You can select corresponding script for your database.

brary 🔻 Share with 💌	Compatibility files New folder		
Name	Date modified	Туре	Size
MySQL.sql	2017/2/22 22:58	Microsoft SQL Ser	6 KB
MySQL_New.sql	2017/2/22 22:58	Microsoft SQL Ser	8 KB
📄 Oracle.sql	2017/2/22 22:58	Microsoft SQL Ser	5 KB
Dracle_New.sql	2017/2/22 22:58	Microsoft SQL Ser	8 KB
PostgreSQL.sql	2017/2/22 22:58	Microsoft SQL Ser	6 KB
PostgreSQL_New.sql	2017/2/22 22:58	Microsoft SQL Ser	9 KB
🛐 Sqlite.sql	2017/2/22 22:58	Microsoft SQL Ser	5 KB
Sqlite_New.sql	2017/2/22 22:58	Microsoft SQL Ser	8 KB
SqlServer.sql	2017/2/22 22:58	Microsoft SQL Ser	5 KB
SqlServer_New.sql	2017/2/22 22:58	Microsoft SQL Ser	8 KB
SqlServer2000.sql	2017/2/22 22:58	Microsoft SQL Ser	5 KB

The scripts with "_New" in the file name are designed for customers out of China.

Step 3. Setup database connection parameters.

Launch ConfigTool.exe in the ConfigTool subfolder in Scale Key installation folder to setup database connection.

Local Disk (C:) Program Files (x86) So	aleKey 🕨 ConfigTool I	•	▼ ∮g	Search Config
ibrary Share with New folder				
Name	Date modified	Туре	Size	
퉬 zh-CN	2017/4/18 11:44	File folder		
ConfigTool.exe	2017/4/10 23:10	Application	1,540 KB	
🗟 ConfigTool.exe.config	2017/2/20 15:20	XML Configuratio	1 KB	

You need specify database type, database server IP and port, database name, database user name and password, and then click the "Test Connection" button.

ScaleKey Config Tool 30213860 1.12.17.0407	Exit			
Time Sender Task II)	Task Status	Return Code	Memo
Save Start Stop Refresh Serve	a (3) er Address Exit	Connection Name	Postgre SQL	
General Settings Log Settings User Settings ScaleList Settings TransferData Settings General Settings	PostgreS Postg	Database Type Server Server Port Database User Password	Postgre SQL 127.0.0.1 5348 scalemanager scalemanager	• •
		System Authentication		Test Connection

After test connection succeeds, click the save button on top left. Scale Key then will automatically save transaction logs into customer database.