

Spry Software

QuickLoad-JMP

User Guide

1. Preliminaries

Introduction.....	3
Saving the QuickLoad-JMP License File.....	3
Installation.....	4

2. Basic Functions

Opening QuickLoad-JMP and Getting Started.....	5
Add File(s).....	5
Remove File(s).....	6
Remove All Files.....	6
Load Selected Files to JMP.....	6
Exit.....	6
Exit Quickload After Loading Files.....	6
About QuickLoad-JMP.....	6

3. Customizing QuickLoad-JMP

Select Data Type.....	7
Select Data Presentation.....	8
Define Parametric Column Name Format.....	9
Set Percentage of Parametric Rows to Include.....	10
Optional DTR Pattern for Last DTR Col.....	10
Saving Configurations.....	10

4. Advanced Options

Finding your QuickLoad-JMP home directory.....	11
Adding Customized Configuration Scripts.....	12
Adding Parametric or Equivalents Scripts.....	12
Running QuickLoad-JMP as a Command Call.....	13

Preliminaries

Introduction

Spry Software provides low-cost components that help you to analyze Automated Test Equipment (ATE) data for semiconductor products, including the easy to use QuickLoad-JMP Add-In that enables you to access STDF files from within JMP.

QuickLoad-JMP is an Add-In that provides fast access to STDF data directly from JMP. Use it to load STDF parametric data with limits and bin summary data directly into JMP and eliminate intermediate scripts, multi-step processes, missing limits or unmaintained scripts of questionable origin. QuickLoad-JMP is now available for JMP versions 10, 11, 12, 13 and 14 on Microsoft Windows and Apple OS X for 32 and 64 bit JMP.

For any questions not answered by this User Guide, please send e-mail to support@sprysoftware.net.

Saving the QuickLoad-JMP License File

If you download a trial version of QuickLoad-JMP, a temporary license file will be automatically generated.

If you purchase a QuickLoad-JMP license, Spry Software will provide you with a license file.

The QuickLoad-JMP license file should be saved in a directory ending in AppData\Roaming\spry directory. Note that the file should be named **.quickLoadJmp.license** with a leading "."

If you are using Windows XP, save the file in a directory such as:

```
C:\Documents and Settings\John\AppData\Roaming\spry
```

If you are using other versions of Windows, save the file in a directory such as:

```
C:\Users\John\AppData\Roaming\spry
```

If you are using OS X save the license file in your home directory.

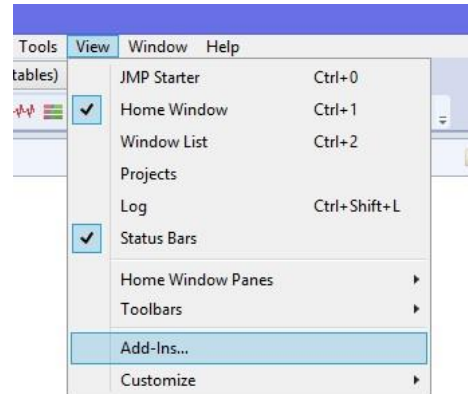
Installation

Download QuickLoad-JMP from the Spry Software website (<http://sprysoftware.net/download-quickload-jmp/>), then save the file.

Some browsers may change the name of the JMP 9 QuickLoad-JMP Add-In to "QLJmp9.zip". If this occurs, change the name to "QLJmp9.jmpaddin" so that the program will load properly.

After saving the file, run it or open it with JMP. QuickLoad-JMP will already appear as an Add-In when the program is opened.

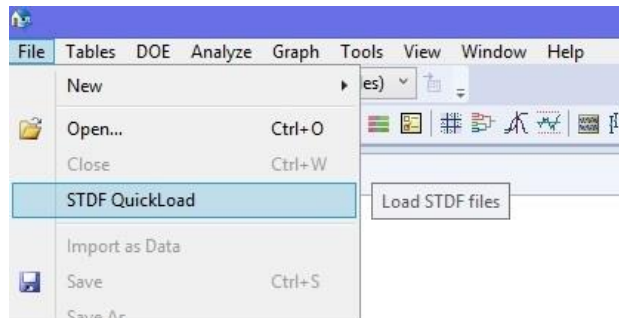
Click **View > Add-Ins** to see information about QuickLoad-JMP. By default, QuickLoad-JMP will be enabled.



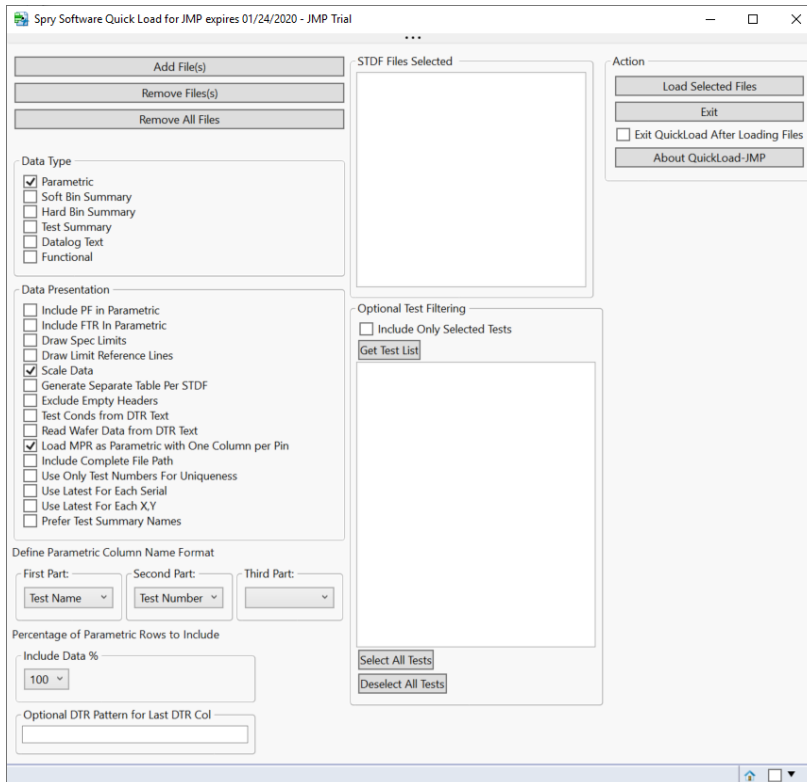
Basic Functions

Opening QuickLoad-JMP and Getting Started

Once JMP is running, click **File > STDF QuickLoad**. This will open the **Spry Software QuickLoad for JMP** window.



Add File(s)



To add a file, click **Add File(s)**. The **Select the STDF files to Use** window will open. Navigate to the directory where your STDF files are stored. Click on the file name and click **Open**. The file name to be loaded will appear in the **STDF Files Selected** box.

In JMP 10 or greater, you can select multiple files before clicking **Open**.

In earlier versions, you must open each file separately by clicking **Add File(s)** and repeating the process described above for each file.

Note that files can optionally be gzipped (.gz) or zip archives (.zip). Files with those suffixes will be included in the list of files in the selection window.

Remove File(s)

To remove a file from the **STDF Files Selected** box, click the file name and click **Remove File(s)**.

Remove All Files

To remove all the files from the **STDF Files Selected** box, click **Remove All File(s)**. The **STDF Files Selected** box should no longer contain any file names.

Load Selected Files to JMP

Once one or more files have been selected, they may be loaded into JMP by clicking **Load Selected Files to JMP**. The file data will be opened in a new data table in JMP.

	Data File	Lot id	Sub Lot id	Tester	Date	Time	Wafer id	Site	Part id	X	Y	HBIN	SBIN	Device	Device
1	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	1	5	5	0	0	SRM9	
2	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	2	6	5	2	0	SRM9	
3	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	3	7	5	14	0	SRM9	
4	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	4	9	6	1	0	SRM9	
5	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	5	8	6	2	0	SRM9	
6	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	6	7	6	2	0	SRM9	
7	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	7	6	6	1	0	SRM9	
8	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	8	5	6	2	0	SRM9	
9	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	9	4	6	0	0	SRM9	
10	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	10	3	7	1	0	SRM9	
11	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	11	4	7	0	0	SRM9	
12	C:\Users\rober_000\Desktop\pw02.std	A525642	2	TERADYNEJ750	09/10/2005	8:38:40 AM	2	0	12	5	7	1	0	SRM9	

Exit

To manually close the **Spry Software QuickLoad for JMP** window at any time, click **Exit**.

Exit Quickload After Loading Files

To automatically close the **Spry Software QuickLoad for JMP** window after files are loaded into JMP, check the box labeled “Exit QuickLoad After Loading Files”. If this option is not selected, then the **Spry Software QuickLoad for JMP** window will remain open after files are loaded.

About QuickLoad-JMP

To view the software version and copyright information, click About QuickLoad-JMP.



Click OK to close the information message.

Customizing QuickLoad-JMP

QuickLoad-JMP can be customized by using the configuration fields in the **Spry Software QuickLoad for JMP** window, or by running custom scripts on the imported data after they are loaded into JMP.

The **Spry Software QuickLoad for JMP** window features a set of check boxes and drop-down windows that customize the way JMP parses STDF files and imports their data.

The four types of settings that can be customized this way are labeled Data Type, Data Presentation, Parametric Column Name Format, and Percentage of Parametric Rows to Include.

Select Data Type

Up to 6 different data types may be selected to be viewed in JMP: Parametric, Multiple Parametric, Soft Bin Summary, Hard Bin Summary, Test Summary, and Datalog Text.



Data Type	
<input checked="" type="checkbox"/>	Parametric
<input type="checkbox"/>	Multiple Parametric
<input type="checkbox"/>	Soft Bin Summary
<input type="checkbox"/>	Hard Bin Summary
<input type="checkbox"/>	Test Summary
<input type="checkbox"/>	Datalog Text
<input type="checkbox"/>	Functional

Each data type selected will open as a separate data table in JMP. Each data table will be labeled by file name and data type, such as “filename.stdparametric.data”. This way, if different data types from several files are opened, they can be easily distinguished from each other.

Data Types:

Parametric: The Parametric table includes the results of each parametric test executed by the test program. This data type will include additional information about the test, such as the test limits, units of measurement, and the scale of the data. Rows are identified by data file, lot ID, and part ID.

Soft Bin Summary: The Software Bin table contains a count of the parts associated with a logical bin or category as a result of testing.

Hard Bin Summary: The Hardware Bin table contains a count of the parts physically placed in a particular bin as a result of testing.

Test Summary: The Test Summary table contains the contents of the input files’ Test Synopsis Records. These include counts of executed tests and failures with one row per parametric test in the test program. It also includes indexing information such as test name, test number, head number, and site number. Columns for statistical summary fields such as Sums and Squares may be left empty if they were not populated in the Test Synopsis Record of the STDF file.

Datalog Text: The Datalog Text Record (DTR) contains text information that has been recorded through the testing process. This information is stored as comments in the datalog listing.

Functional: This format contains the results of Functional Test Records (FTR) including pass or fail test results plus additional data. This data table contain a column for each field in the FTR, and a row for each test performed.

The Window List in the JMP Home Window view will list each data table that is currently open. One way to view a data table is to double-click on the data table name in the Window List. **Spry Software QuickLoad for JMP** will also appear in the Window List, unless **Exit QuickLoad After Loading Files** was selected in the **Spry Software QuickLoad for JMP** window.

Select Data Presentation

There are several options for customizing the way data from the STDF file will be presented in JMP. To select an option, click on the check box next to it. All of the options can be selected, or they can all be left unselected. Unless a configuration script is used, selections made for the previous file loaded will be automatically selected the next time QuickLoad-JMP is opened.

Data Presentation Options:

Include PF in Parametric: A Pass/Fail column for each parametric or functional test will be included in the Parametric table if this choice is selected.

Include FTR In Parametric: A column for each functional test will be included in the Parametric table if this choice is selected. In each of the functional test columns, a 1 indicates that the unit for that row passed the test, a 0 indicates that it failed, and no value indicates that the test was not executed.

Draw Spec Limits: Vertical lines representing the lower specification limit and the upper specification limit will be included in the data table.

Scale Data: The scale used to label data in the file will be used in the data table.

Generate Separate Table Per STDF: This feature will open each STDF file in a separate data table.

Exclude Empty Headers: Some columns in the STDF file may contain no test data. This setting will omit them from the data table.

Test Conds from DTR Text: The test conditions from the Datalog Text Record (DTR) will be displayed in the data table.

Data must adhere to the following format to be read as test conditions:

COND:c1=y1,c2=y2,c3=y3

In this example, COND: is a fixed value and ci=yi are one or more pairs of condition name=value, separated by commas.

Data Presentation

- Include PF in Parametric
- Include FTR In Parametric
- Draw Spec Limits
- Scale Data
- Generate Separate Table Per STDF
- Exclude Empty Headers
- Test Conds from DTR Text
- Read Wafer Data from DTR Text
- Load MPR as Parametric with One Column per Pin
- Include Complete File Path
- Use Only Test Numbers For Uniqueness

Read Wafer Data from DTR Text: The Wafer ID, X_LOC and Y_LOC values from DTR records will be displayed in the data table.

The DTR value must have three space-delimited fields, indicating type of data, test site and value, as in the following example:

```
WAFER 1 18
X_LOC 2 14
Y_LOC 1 -7
```

In this example, the first line indicates wafer 18, tested with test site 1. The second line indicates an X_LOC of 14, tested with test site 2 and the third line indicates a Y_LOC of -7, tested with test site 1. The test site is used to tie the DTR to one of the parts currently being read from the STDF file.

Load MPR as Parametric with One Column per Pin: This loads the multiple values from the Multiple-Result Test Record into the Parametric table in a separate column for each pin. The MPR channel name is appended to the column name.

Include Complete File Path: When this box is checked, the file name column in the data table will contain the entire file path, including the file name. When it is unchecked, the file name column will only contain the file name.

Use Only Test Numbers For Uniqueness: When this box is checked, only the test number is used to identify a test as unique from other tests. When it is unchecked, the combination of test name and test number is used to determine uniqueness.

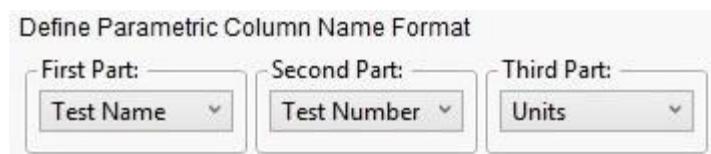
Use Latest For Each Serial: When this box is checked, only the latest entry in each STDF file for each part identifier will be included in Parametric and Functional output tables.

Use Latest For Each X,Y: When this box is checked, only the latest entry in each STDF file for each X,Y pair will be included in Parametric and Functional output tables.

Prefer Test Summary Names: When this box is checked, test names from TSR records will be used rather than test names from PTR, MPR and FTR records, if TSR records are present.

Define Parametric Column Name Format

Parametric columns can be ordered by Test Name, Test Number, and Units. Data will be sorted according to the order selected for the First Part, Second Part, and Third Part.



Define Parametric Column Name Format

First Part:

Second Part:

Third Part:

It may be practical in some cases to order the data by Test Name first, then Test Number, and finally by Units. In other cases, it may be better to order the data by Units first.

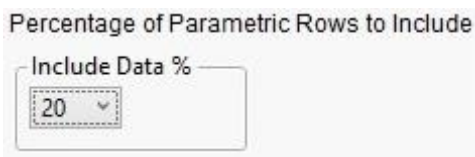
Use the drop-down menus to choose Test Name, Test Number, or Units for each part. If no second or third part is selected, then the columns for tests will be labeled by the first part only.

When all the tests are numbered zero, QuickLoad-JMP will treat them as the same test and hide some of them. If this happens, set First Part to "Test Name" and have the other two parts blank (or set Second Part to Units). You should then see the full set of tests.

Select Percentage of Parametric Rows to Include

Sometimes a particularly large data file may cause JMP to load more slowly than desired. Selecting a lower percentage of the data to analyze will allow data to be loaded faster.

Use the drop-down menu to select the percentage of the data to be loaded into JMP.



If less than 100% is selected, QuickLoad-JMP will take a random sample of the data in the file based on the selected percentage. For example, if 20% is selected, then QuickLoad-JMP will randomly select 20% of the data in the file and only load this selection of the data into JMP.

Optional DTR Pattern for Last DTR Col

Enter the pattern that a last DTR record must match in order to be included in the Last DTR column.

The parametric data table generated by QuickLoad-JMP includes a Last DTR column. This column displays the last DTR record contained in the file within the range of records with the test results for a device. Only records matching the pattern entered in the "Optional DTR Pattern for Last DTR Col" field will be included in the Last DTR column.

Saving Configurations

Each time the QuickLoad-JMP data import window is opened, the configuration fields will be populated with the same values used in the previous session. This function may be overridden by a configuration script (see **Adding Customized Configuration Scripts** below).

Advanced Options

Finding your QuickLoad-JMP home directory

For basic usage of QuickLoad-JMP you should not have to find your QuickLoad-JMP home directory. However, you will need to find your home directory if you want to add custom scripts because the scripts will be stored in a subdirectory under the home directory as described for

each script type.

Your QuickLoad-JMP home directory will normally be under your Windows home directory. If your Windows home directory is c:\users\john, then your QuickLoad-JMP home will be:
C:\Users\John\AppData\Local\SAS\JMP\Addins\net.sprysoftware.quickloadjmp

Adding Custom Configuration Scripts

QuickLoad-JMP will execute any JSL scripts found in the config subdirectory of your QuickLoad-JMP home directory. These scripts can be used to override previously saved default values in the visible configuration fields. However, these scripts are mainly used to set minor configuration fields that are not available in the **Spry Software QuickLoad for JMP** window.

These scripts will be executed when QuickLoad-JMP starts up. To set the field, call the JSL function **updateoption** with the name of the configuration field and the value to use like this:

```
updateoption("<OptionName>",1);
```

You must substitute <OptionName> with the name of the option you want to change.

Be aware that all files in the config directory will be removed if you upgrade the QuickLoad-JMP Add-In, so keep an original of your scripts in another location. Spry Software has requested a change in this behavior in the next version of JMP.

Adding Parametric or Equivalent Scripts

You can also run a script that is specific to the type of data in a table after the table is loaded. This type of script will be stored in a **parametric** subdirectory or an **equivalents** subdirectory. The following example is a JSL script that is used to remove test names starting with ": :".

```
dt = CurrentDataTable();
For( i = 1, i <= Ncol(dt), i++,
    nextColName = ColumnName(i);
    nextColStr = char(columnname(i));
    nextCol = column(nextColStr);
    colonIndex = Contains(nextColStr, " : ");
    if ((colonIndex > 0),
        newStr = left(nextColStr, colonIndex - 1);
    // print("replace " || nextColStr || " with " || newStr);
        nextCol<<setName(newStr);
    );
);
For( i = 1, i <= Ncol(dt), i++,
    nextColName = ColumnName(i);
    // show(nextColName);
);
```

This particular script would be stored in a parametric subdirectory such as:

C:\Users\John\AppData\Local\SAS\JMP\Addins\net.sprysoftware.quickloadjmp\parametric

Running QuickLoad-JMP as a Command Call

You can run QuickLoad-JMP from the command line and use the software's core functionality without launching the user interface. The following example is a JSL script used to run QuickLoad-JMP with several options selected.

```
// *** Copyright Spry Software 2015 ***
// Function to run Quickload-JMP as a command call
runDirectParametric = Function( {fileNames, dllLoc},
    {default local},

    Try(
        dll = Load DLL( dllLoc );
        dll << Declare Function("startParametricReport",
            Convention(CDECL),
            Alias("startParametricReport"), Arg(AnsiString),
            Returns(Int32));
        dll << DeclareFunction("setOption", Convention(CDECL),
            Alias("setOpt"), Arg(AnsiString), Returns(Int32));
        dll << Declare Function("addFileToReport", Convention(CDECL),
            Alias("addFileToReport"), Arg(AnsiString, "stddfFile"),
            Returns(Int32));
        dll << Declare Function("writeReport", Convention(CDECL),
            Alias("writeReport"), Arg(Int32, "option"), Returns(UIInt64));
        loaded = 1;
        //Show( "Loaded dll with add-in path" );
        ,
        Warning(
            "Error during loading of the QuickLoad-JMP dll.
            "
        )
    );
    dll << setOpt( "first name part=2" );
    dll << setOpt("second name part=1" );
    dll << setOpt("third name part=0" );
    dll << setOpt("Include PF in Parametric=0" );
    dll << setOpt("Test Conds from DTR Text=0" );
    dll << setOpt("Scale Data=0" );
    dll << setOpt("draw spec limits=1" );
    dll << setOpt("Generate Separate Table Per STDF=0" );
    dll << setOpt("use_selected_tests=0");
    show(fileNames);
    makeReport( dll, fileNames, "startParametricReport", "parametric",
        1, 0, 0 );
);

paths = Associative Array();

//Location of JMP quickload DLL
//addinsDir = Get Path Variable( "$HOME" );
//Substitute Into( addinsDir, "/14/", "/Addins/" );
addinsDir = GetPathVariable("ADDIN_HOME(net.sprysoftware.quickloadjmp)");
dllLoc = addinsDir ||
    "quickLoadJmp.dll";
```

```

show(dllLoc);
//Get list of files from somewhere else in user's script
fileList = {"c:/stdf/generated/BZ_copy.stdF"};
If( N Items( fileList ) != 0,
finalFileList = {});
Try(
  show(addinsDir);
  limitScripts = Convert File Path( addinsDir ||
    "setSpecLimits.jsl", absolute, Windows);
  try(
    Include( limitScripts );
    show("after include limitscripts"),
    show("problem during include limit scripts", exception_msg);
    show(limitScripts);
  );
  engine = Convert File Path( addinsDir ||
    "qljEngine.jsl", absolute, Windows );
  try(
    Include( engine ),
    show("problem during include engine ",exception_Msg);
    show(engine);
  );
  For( i = 1, i <= N Items( fileList ), i++,
    nameOnly = storeFilePath( fileList[i] );
    show(nameOnly);
    Insert Into( finalFileList, nameOnly ); //Generate List of
filenames
  );

  runDirectParametric( finalFileList, dllLoc );
  , //If error occurs....
  Show( "Problem during run", exception_msg )
);
);

```