STDF Workshop
使用手册

A Light and Intelligent Solution

STDF Workshop
(STDF Data Management Tool)
使用手册

http://www.nornion.com

Document Number: NL-009-01 Rev.K

目录

4 详解各种记录的修改

5 其他

1
简介

- **STDF Workshop** 是什么.
- **STDF** 数据格式简介.
- 典型 **STDF** 数据结构
- 一个 **Insertion** 的数据结构
- 数据记录之间的相关性

STDF Workshop 是什么？

STDF Workshop 是一个 STDF 管理工具，可以用于查看 STDF 的所有记录和每个记录的内容，同时可以修改每个记录的每个字段的内容，添加和删除记录，合并和修复 STDF。STDF Workshop 在修改记录内容的时候支持复制粘贴来批量更新多个记录的内容。

STDF Workshop 主要是为了数据管理员设计的，因为每个大的半导体公司都有专职的数据管理员负责检查 STDF 数据的完整性，记录内容是否正确，确保每个进入系统或者抛给客户的 STDF 数据都是完整有效的。

在 STDF Workshop 中数据都是按照原始的记录(Record)呈现的,我们同时也提供了把同类型的记录数据以表格形式呈现并可以导出到 Excel。由于 STDF Workshop 保留了 STDF 中的所有记录，并且为了数据完整性扩充了所有 STDF“短记录”为完整“长记录”，所以 STDF Workshop 解析 STDF 之后会占用比较高的内存(内存占用量可以按文件大小的 5 倍估算), 所以请在内存比较大的电脑上使用 STDF Workshop。(如果您只需要检查数据，而不需要修改 STDF，那可以选择“在线”解析模式，此模式速度快且节省内存。如果需要修改 STDF 数据，必须使用“离线”解析模式，此模式会一次性把所有数据加载到内存)。

STDF 数据格式简介

STDF 数据格式是半导体测试行业的通用数据格式，在半导体测试行业广泛被使用，市面上绝大多数的测试机平台都支持 STDF 格式。使用 STDF 格式具有非常多的有点：1)文件 size 比较小，相对于文本文件，二进制的 STDF 文件数据非常紧凑，而且可以去除很多重复的冗余数据进一步减小文件大小。2) 文件格式统一，后期分析非常方便，而不需要额外去了解不同测试机平台的数据格式差异，企业可以开发通用的数据系统来解析，存储和分析不同平台的数据，建立自己的数据分析系统。

STDF 数据是一种二进制的流式文件，不能直接阅读，需要借助特定的工具来解析和分析数据。STDF 数据是按照记录(Record)来存储的，可以认为记录是 STDF 数据的最小数据块，不同的记录类型用来存储不同的数据内容。

这里我们简单介绍一下常见的记录类型和作用，每个记录的详细信息可以去参考 STDF Spec V4.

FAR: 这个是 STDF 的第一个记录，如果一个 STDF 文件的第一个记录不是 FAR 则可能导致 STDF 无法解析。

MIR 和 MRR: 这一组记录表示一个 lot 的开始和结束，内部的所有记录就是当前 lot 的相关数据，一般一个 STDF 只有一对 MIR 和 MRR。MIR 是 lot 开始的记录，里面包含了许多开批的信息(如：批号，程序名，产品名，测试机编号，开批时间，操作员工号等)，它也是我们最常修改的数据记录。MRR 是 lot 结束的记录，主要包含 lot 的结束时间。

WIR, WCR 和 WRR: 这三个记录也是一组相关的记录，都是和 Wafer 测试相关的(FT 数据没有这三个记录)。其中 WIR 是 Wafer 的开始记录，包含 Wafer_ID, 和 wafer 测试开始时间等信息，WCR 是 wafer 测试的配置信息，包含 Wafer_Size, Die_Size, Wafer_Flat, CenterDie 坐标等信息，但是最常用的是 Wafer_Flat(晶圆 Notch 方向)。WRR 是 Wafer 的结束记录，主要包含 Wafer_ID 和测试数量统计信息(PART_CNT, GOOD_CNT 等)。一般一个 STDF 中只有片 wafer 的数据，也就是只有一对 WIR 和 WRR，但是也不排除有包含多片 Wafer 数据的 STDF 文件，那么在此类 STDF 中就会有多个 WIR 和 WRR 对。

SDR: 此记录主要是存储硬件相关信息的，比如 handler/prober 类型和编号，Load board 类型和编号，probe card 类型和编号以及 site (工位) 的信息。

PCR: Lot 测试数量的统计，这个记录主要用来统计整个 lot 测试结果(测试总数量，pass 的数量和 fail 的数量)。通常出现在 STDF 的末尾，在 STDF 结批的时候自动添加。

HBR 和 SBR: 这两个记录是 hardware bin 数量和 software bin 数量统计的记录，统计了每个 bin 的数量，这个其实就是测试 summary 里面的 bin 信息。通常出现在 STDF 的末尾，在 STDF 结批的时候自动添加。

TSR: 统计测试项执行数量和 fail 数量的记录，为每个不同测试项统计当前测试项执行了多少数量，在这些数量中有多少是 fail 的。一般每颗 Unit 会把每个测试项执行一遍，但是如果 stoponfail 并且当前 unitfail，那 fail 的测试项后面的就不被执行了，所以一般第一个测试项的统计数量就是 lot 的总数量。通常出现在 STDF 的末尾，在 STDF 结批的时候自动添加。

PIR 和 PRR: 这一对记录标记了每个 Unit (或者每一个 Insertion) 测试的开始和结束，包含在这一对记录里面的数据都是属于当前 Unit(或者当前 Insertion)。在 PIR 中只有 HEAD_NUM (测试头信息，默认为 1)和 SITE_NUM 标记。在 PRR 中包含了当前 Unit 的测试结果信息，如 PART_ID, HBin, SBin, Test Time 和 X，Y 坐标等信息。包含在这对记录之中的都是测试项的结果记录和一些附加信息(打印文字等)。

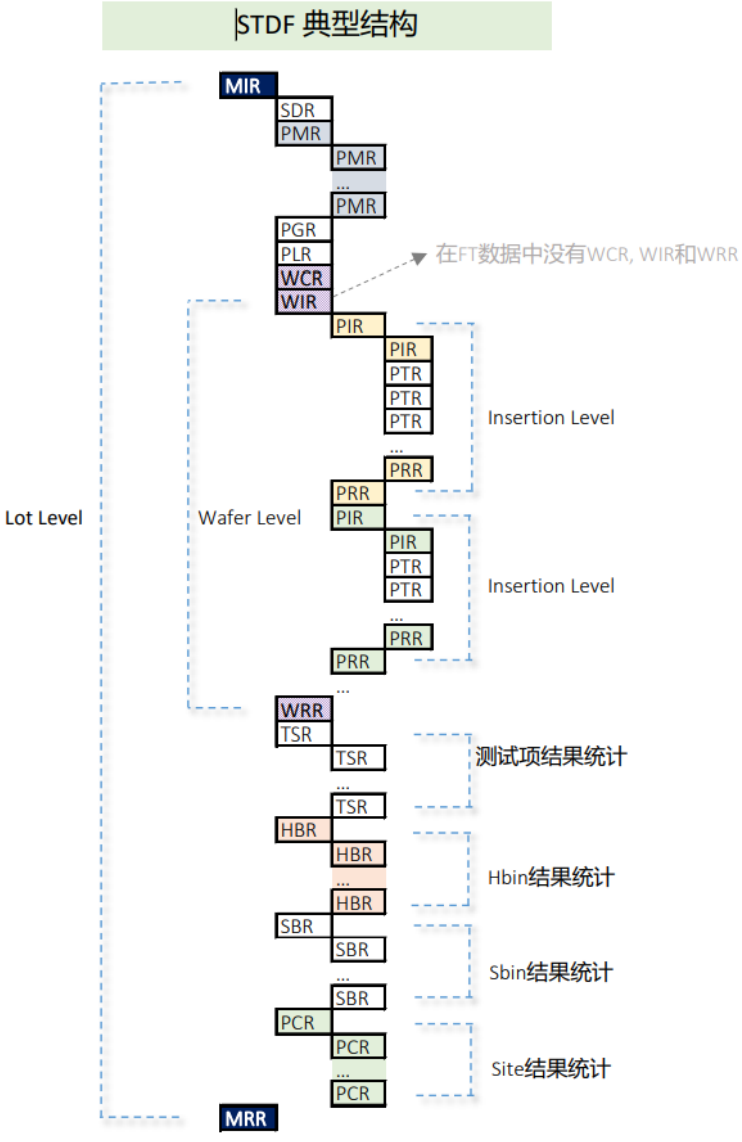
PTR: 这个就是最常见的参数测试项的测试结果记录，也是 STDF 中包含的最多的记录类型。通常都是在每个 PIR 和 PRR 对中的，包含测试头号，工位号，测试项编号和名称，测试结果(test measurement)，pass-fail flag 以及 limit 信息。

FTR: 这个记录类型对应测试中的 digit 测试项(跑 pattern 的结果)，此记录同样包含包含测试头号，工位号，测试项编号和名称，还有一些 Pattern Name, TimmingSet Name 等信息，但是没有测试结果，而只有 pass-failflag。

MPR: 这个记录也属于参数测试项，但是一个记录中可以包含很多个测试结果，它比 PTR 的数据要紧凑得多。在大型测试机上的硬件 pincount 比较多用来测试大型 Soc 芯片的时候，一般一个测试函数可以测很多个 pin, 比如 O/S 测试项，Leakage 测试项等等，这时候测试函数一样，测试 limit 也一样，而且很多个 pin 都是同时测试的，所以目前大型测试机也经常用 MPR 来存储数据，来减少文件的大小。MPR 也同样包含包含测试头号，工位号，测试项编号和名称，Limit 和测试结果信息。

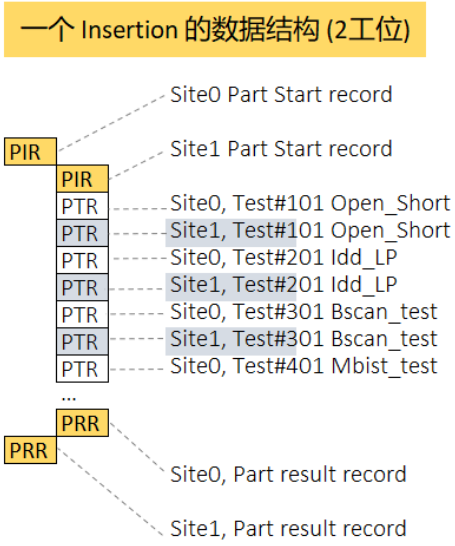
一般情况如果测试的时候没有正常结批，会导致 STDF 没有正常结束，数据会缺少 HBR, SBR, TSR, PCR 和 MRR 记录。

典型的 STDF 数据结构



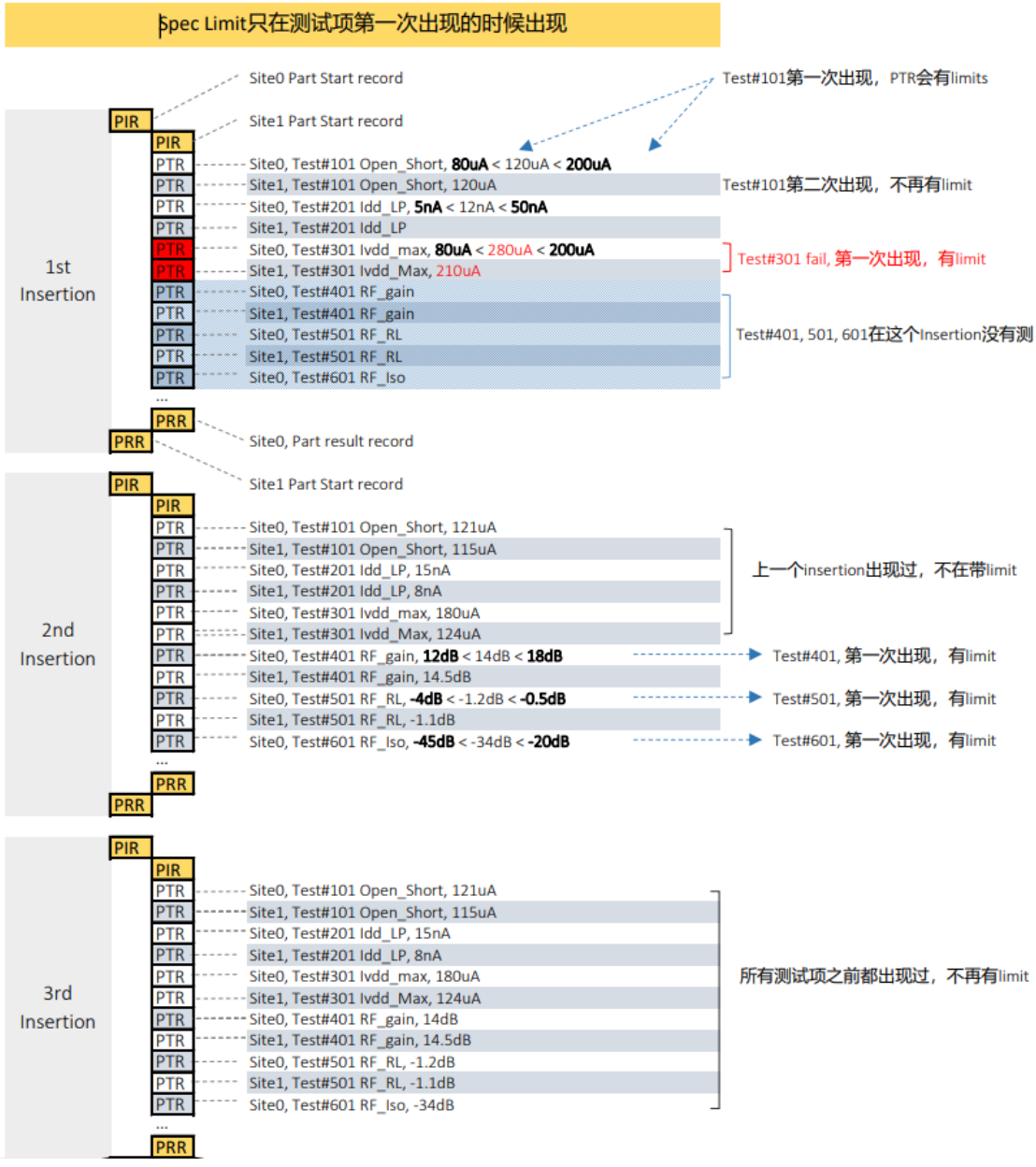
1. 这里我们用图形再展示一下典型的 STDF 数据结构，以提供更加直观的感受，了解各种记录在 STDF 中的位置和出现的频率。可以看出一个 lot 的数据被 MIR 和 MRR 记录包括，开头有一些 SDR, PMR 等信息，末尾是 HBR, SBR, TSR, PCR 等 summary 记录信息，而主题都是很多个由 PIR 和 PRR 包括的 Insertion 的测试数据块。

2. 下面是一个 Insertion 的数据结构。一个 Insertion 代表一次并行测试(handler 一次下压)，如果有多个工位，那么一次测试就有多颗数据。下面是一个 2 工位测试的一个 Insertion 数据示意图。STDF 数据的主体就是非常多个 Insertion 数据块。



3. 下面介绍一下测试项结果记录 PTR(和 MPR)的 limit 信息。STDF 规范规定，limit 信息只需要在同一个测试项第一次出现的时候出现就可以，后面都可以省略，这样可以节省数据空间。这些重复的 Limit 数据也确实没有必要重复存储到数据中。所以在修改 STDF 数据的时候，如果需要修改 Limit 信息，请找到 limit 出现的记录，也就是测试项第一次出现的地方，大多数情况都在第一个 Insertion 里面。

同时，在修改测试结果的时候需要用户自行判断测试项 Pass/Fail 结果是否改变；这个时候需要用户自己必须知道当前修改测试项的 Limit 范围。


Spec Limit只在测试项第一次出现的时候出现
Site0 Part Start record
Test#101第一次出现，PTR会有limits
Site1 Part Start record
1st Insertion
PIR
PIR
PTR Site0, Test#101 Open_Short, 80uA < 120uA < 200uA
PTR Site1, Test#101 Open_Short, 120uA
Test#101第二次出现，不再有limit
PTR Site0, Test#201 Idd_LP, 5nA < 12nA < 50nA
PTR Site1, Test#201 Idd_LP
PTR Site0, Test#301 Ivdd_max, 80uA < 280uA < 200uA
PTR Site1, Test#301 Ivdd_Max, 210uA
Test#301 fail, 第一次出现，有limit
PTR Site0, Test#401 RF_gain
PTR Site1, Test#401 RF_gain
PTR Site0, Test#501 RF_RL
PTR Site1, Test#501 RF_RL
PTR Site0, Test#601 RF_Iso
Test#401, 501, 601在这个Insertion没有测
...
PRR
PRR
Site0, Part result record
Site1 Part Start record
2nd Insertion
PIR
PIR
PTR Site0, Test#101 Open_Short, 121uA
PTR Site1, Test#101 Open_Short, 115uA
PTR Site0, Test#201 Idd_LP, 15nA
PTR Site1, Test#201 Idd_LP, 8nA
PTR Site0, Test#301 Ivdd_max, 180uA
PTR Site1, Test#301 Ivdd_Max, 124uA
上一个insertion出现过，不在带limit
PTR Site0, Test#401 RF_gain, 12dB < 14dB < 18dB
Test#401, 第一次出现，有limit
PTR Site1, Test#401 RF_gain, 14.5dB
PTR Site0, Test#501 RF_RL, -4dB < -1.2dB < -0.5dB
Test#501, 第一次出现，有limit
PTR Site1, Test#501 RF_RL, -1.1dB
PTR Site0, Test#601 RF_Iso, -45dB < -34dB < -20dB
Test#601, 第一次出现，有limit
...
PRR
PRR
3rd Insertion
PIR
PIR
PTR Site0, Test#101 Open_Short, 121uA
PTR Site1, Test#101 Open_Short, 115uA
PTR Site0, Test#201 Idd_LP, 15nA
PTR Site1, Test#201 Idd_LP, 8nA
PTR Site0, Test#301 Ivdd_max, 180uA
PTR Site1, Test#301 Ivdd_Max, 124uA
PTR Site0, Test#401 RF_gain, 14dB
PTR Site1, Test#401 RF_gain, 14.5dB
PTR Site0, Test#501 RF_RL, -1.2dB
PTR Site1, Test#501 RF_RL, -1.1dB
PTR Site0, Test#601 RF_Iso, -34dB
所有测试项之前都出现过，不再有limit
...
PRR

2
使用 STDF Workshop

- 软件界面
- 记录统计
- 记录筛选和批量修改
- 树形结构
- 修改和保存
- 复制和粘贴
- 合并和修复 **STDF**
- 重新计算 **Summary** 记录
- 重新序列化 **PART_ID**

软件界面

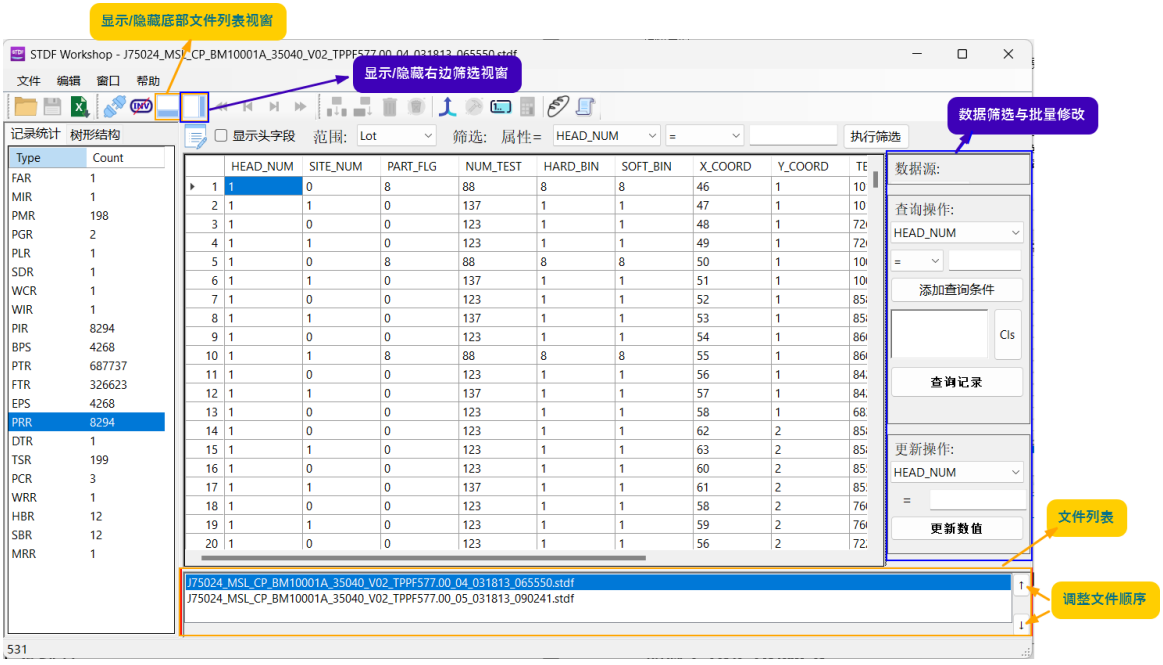
STDF Workshop 的界面非常简洁，使用方便。界面主要分位三大块，上面为菜单栏和工具栏。左边为记录导航区域，右边主区域是记录内容展示和修改区域。在数据展示区域上方还有一个筛选工具条。

在工具栏上有三个按钮：STDF 打开，STDF 保存，数据表格导出到 Excel。STDF Workshop 是针对单个 STDF 操作的，所以每次只能打开一个文件。STDF 的保存，是在修改了 STDF 一个或者多个记录之后，把内容保存到 STDF 中去，需要注意的是在每次打开 STDF 文件后，第一次保存时，需要指定新的 STDF 文件名，必须与源文件名不一致，STDF Workshop 不容许覆盖原文件。数据导出到 Excel，是指把记录数据展示区域的表格导出到 Excel 文件，这个一般是供其他分析使用的。

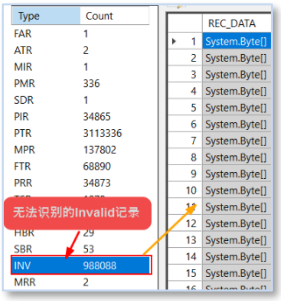
在左边的记录导航栏，主要是用来让用户可以快速找到对应的记录。主要有两种展示模式：按记录类型(Summary)，按记录顺序的树形结构(Tree)。

打开文件

STDF Workshop 可以同时打开单个或者多个 STDF 文件，打开多个文件的时候窗口底部的文件列表就会显示出来，可以用来切换当前文件，也可以通过菜单[窗口]来切换。在文件列表中可以调整文件的顺序，这个顺序对于合并 STDF 至关重要，最上面的文件默认为合并时的主文件。



在解析 STDF 的时候，如果遇到 Invalid 记录类型，默认会停止解析并弹窗提示。如果想要继续解析，可以点击工具栏的 按钮来包含 Invalid 记录，然后重新打开文件即可。*注意：如果包含了 Invalid 记录，但是 Invalid 记录并没有遵循 STDF 的规范，则会导致数据解析错乱进而导致软件卡死。*



STDF Workshop 的 “在线模式” 也是一个非常有用的模式，在这个模式下，STDF 解析速度会大幅度提高，因为这个模式下仅解析记录的**头部**而跳过记录的**内容**，而当用户在记录导航中点击指定记录的时候才会解析当前记录的内容进而展示。*注意：如果需要修改任何记录的内容或者增/删记录，请使用 “离线模式”* 。

记录统计

在按照记录类型的 Summary 展示时，同时统计了每个记录类型的数量，可以用来了解 STDF 的记录状况，也可以了解 STDF 是否完整(包含 HBR, SBR, PCR, MRR 等记录)。在点击每个记录类型的时候，会在右边的数据展示区域看到所有此类型的记录的数据展示，可以通过筛选工具条筛选展示的记录，也可以直接修改任何记录的内容。

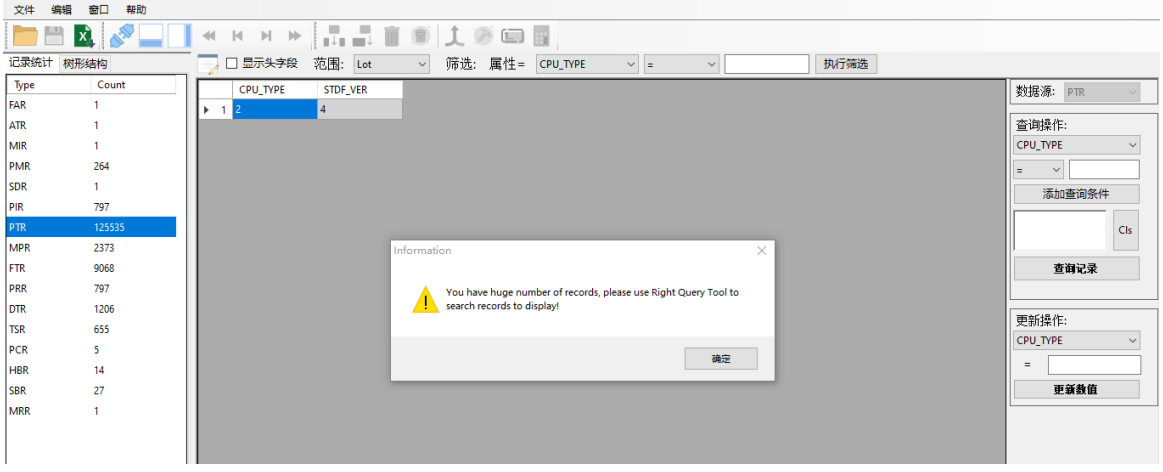
加载 PTR/FTR/MPR

如果您选择的数据记录类型对应的数目非常多的时候，右边表格加载需要比较长的时间(主要时 PTR, MPR 和 FTR)，这些记录尽量在树形导航中去查找。

如果您在记录统计 tab 里面直接点击 PTR 记录，STDF Workshop 会提示您 PTR 记录不会直接展示，如果您真的需要显示所有 PTR 记录，或者部分 PTR 记录，可以在右边的筛选框里面设置筛选条件来显示 PTR 记录。如果没有设置任何筛选条件而直接点击 "查询记录" 按钮，所有的 PTR 记录都会被加载并显示出来。

当表格视图中有记录被展示出来之后，顶部的筛选工具栏才可以被用来做快速筛选。

视图右侧的记录筛选工具的详细使用方法请参看下面的说明。

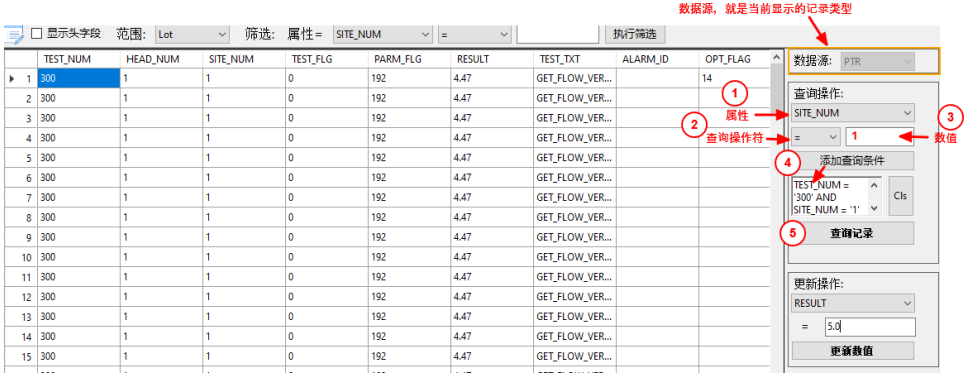


记录筛选和批量修改

在 STDF Workshop 的视图中，右侧部分是记录的筛选和批量修改功能块，用户可以自定义筛选条件来筛选出自己想要的记录，然后通过更新操作一次性更新当前显示的所有记录的某个字段的内容。

筛选的步骤 (如 SITE_NUM=1)：

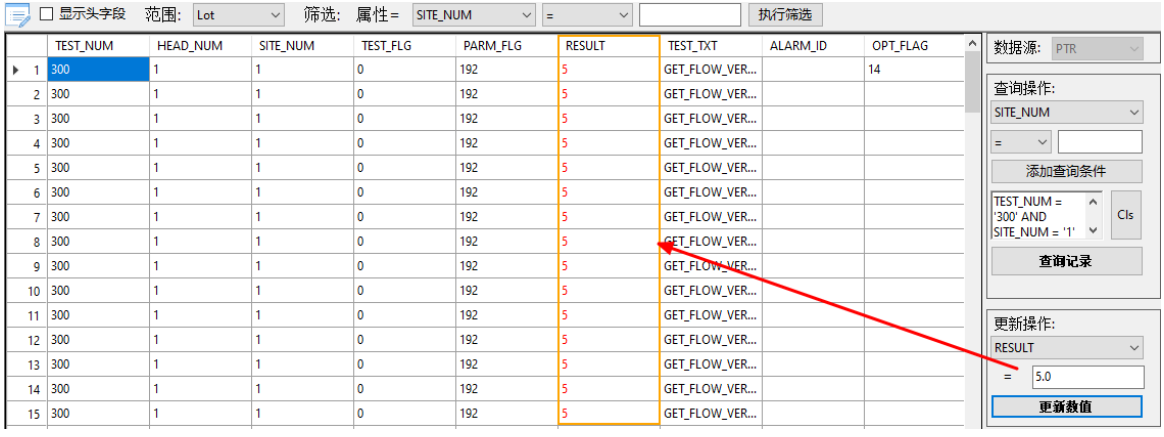
1， 选择你需要根据什么属性来筛选 (如 SITE_NUM)
2， 选择查询操作符 (如 “= “)
3， 设置查询数值 (如 “1 “)
4， 点击 添加查询条件 按钮，查询条件会被添加到文本框中。可以添加多个查询条件，这些查询条件会自动被用“AND”相连
5， 最后点击 查询记录 按钮，符合条件的记录都会被列出来



批量修改的操作 (如，把所有查询出来的记录的 RESULT 修改为 5.0)：

1， 选择更新操作的目标属性
2， 输入新的数值
3， 点击 更新数值 按钮

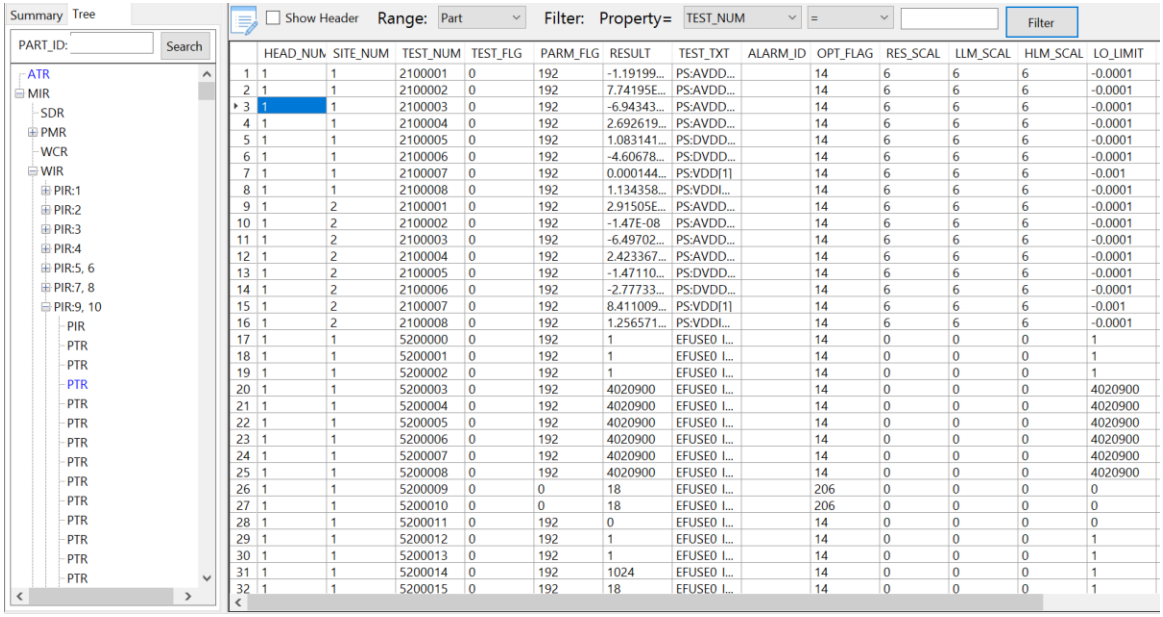
注意：更新操作对当前表格视图的所有记录有效



树形结构

树形结构导航从另一个维度展示了数据记录，完全按照每个记录在 STDF 中的文件顺序展示的，同时又按照 Lot，Wafer 和 InsertionLevel 对记录进行了折叠，可以让用户快速找到需要的记录进行查看和修改。同时我们还在 Insertion Level 的 PIR 记录上附加了当前 Insertion 的 unit 的 PART_ID 信息，并且提供根据 PART_ID 搜索的功能，以方便用户快速找到需要的记录。

在树形导航中，当你输入 PART_ID 进行搜索时，软件如果找到对应的 Insertion，会自动展开

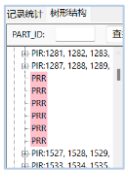


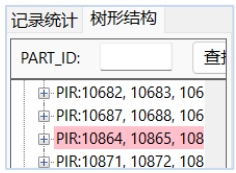
PIR-PRR 节点，而折叠其他打开的 Insertion 节点，让你可以直接看到定位到对应的 Insertion 数据。

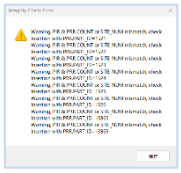
在你点选任何数据记录节点的时候，会在右边的数据展示区域展示对应的记录的数据内容。默认是展示当前 Insertion 的同类型的所有记录。例如点击了一个 PTR 记录，则在右边会展示当前 Insertion 的所有 PTR 记录，同时在右边的树形结构中和右边的表格中用蓝色高亮标记您当前选择的记录。如果您不想软件展示所有 PTR 记录，而只想展示当前您选中的单个 PTR 记录，则选择筛选工具上的 Range = Record 即可。

有时您想展示某些测试项的所有记录，比如您想仅展示当前 Insertion 内的 Test#210001 的所有 PTR，这时候你只需要使用筛选功能即可(TEST_NUM=210001)。您可以根据当前记录类型的任何树形进行筛选。

STDF Workshop 在创建树形结构的时候会对 PIR/PRR 做数据完整性检查，如果有任何 touch-down 的 PIR/PRR 的数量或者 SITE_NUM 不匹配，都会弹框提示，并且在属性结构中用粉色背景高亮显示。



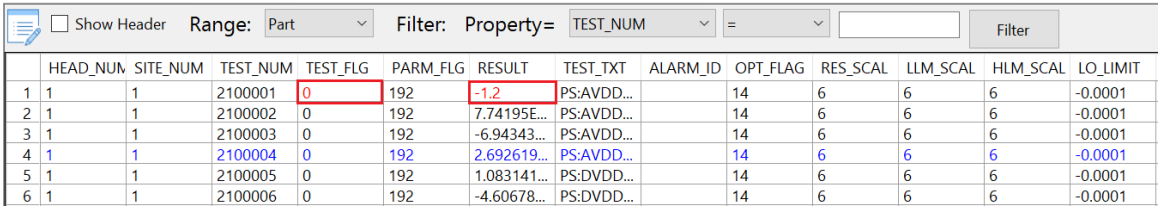




数据修改和保存

修改数据是直接在数据展示表格里面完成的。在你需要修改的单元格上直接双击数据进入编辑模式，编辑完直接回车确认，更新的数据会被记录的内存中。修改后的数据会以红色突显出来。

在 STDF 修改过之后可以直接点击保存  按钮把内存中的数据写入到 STDF 文件中，第一次保存的时候，需要设置不同于源文件的文件名。

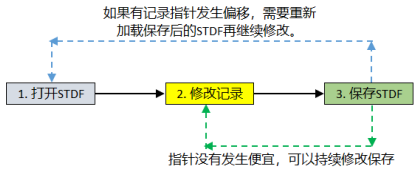


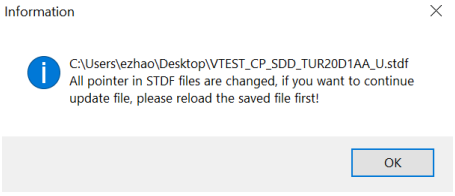
	HEAD_NUM	SITE_NUM	TEST_NUM	TEST_FLG	PARM_FLG	RESULT	TEST_TXT	ALARM_ID	OPT_FLAG	RES_SCAL	LLM_SCAL	HLM_SCAL	LO_LIMIT
1	1	1	2100001	0	192	-1.2	PS:AVDD...		14	6	6	6	-0.0001
2	1	1	2100002	0	192	7.74195E...	PS:AVDD...		14	6	6	6	-0.0001
3	1	1	2100003	0	192	-6.94343...	PS:AVDD...		14	6	6	6	-0.0001
4	1	1	2100004	0	192	2.692619...	PS:AVDD...		14	6	6	6	-0.0001
5	1	1	2100005	0	192	1.083141...	PS:DVDD...		14	6	6	6	-0.0001
6	1	1	2100006	0	192	-4.60678...	PS:DVDD...		14	6	6	6	-0.0001

由于 STDF 是一种以记录为基础的流式文件，所以每个记录在 STDF 中的位置都依赖于它前面的记录，如果任何它前面的记录长度发生改变，那它在文件的位置也会随之变化。所以如果修改记录后带来任何记录的位置发生变化，则在保存后就不能继续保存了(系统会给出下面的提示框)。如果继续修改，需要重新加载新保存的文件，在继续修改和保存。(可以一次性修改多个记录，然后一起保存)

什么情况下会导致记录位置的变化？主要是修改变长字段的时候，特别是字符串内容的长度变化修改了或者列表数据的长度变化了，一定会带来记录位置的偏移。

如果修改记录后没有带来任何记录位置偏移，则保存后可以继续修改继续保存，而不需要重新加载最新保存的文件。

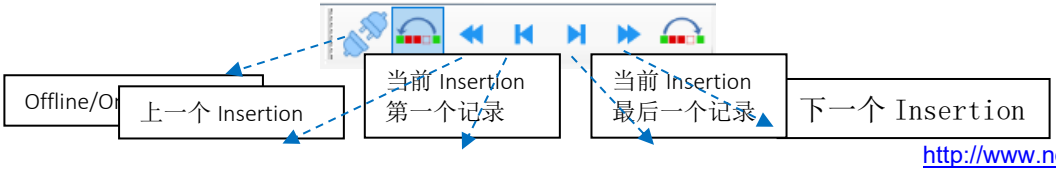




导航工具栏，下面这个工具栏是在树形结构的导航中使用的，可以用来跳到下一 Insertion 或者上一个 Insertion，在同一个 Insertion 内跳到开头或者结尾。

Offline Mode: 一次性把文件的所有数据加载入内存，如果需要修改 STDF，必须用这种模式，这种模式下解析 STDF 比较占内存(可以按照 STDF 文件大小的 5 倍估算需要的内存)。

Online Mode: 只加载每个记录的位置信息，每次点击记录时在实时解析对应的记录。这个解析速度非常快，并且占用内存少。适合仅检查/浏览数据内容时使用。



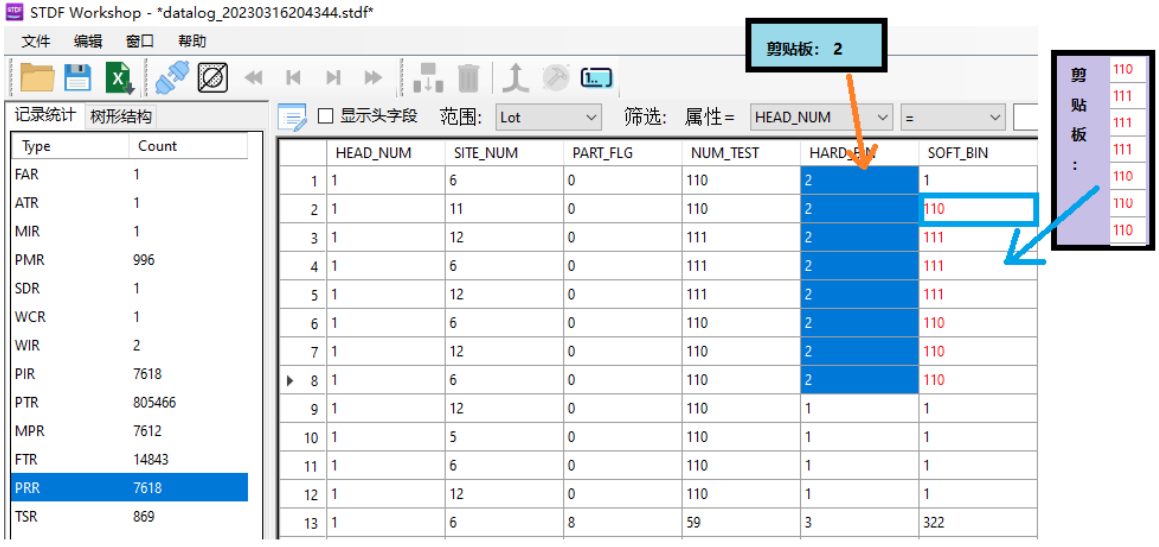
复制和粘贴字段内容

在主窗口的表格中修改数据时可以使用复制和粘贴功能实现批量修改，可以从 Excel，文本或者其他软件中复制数据然后粘贴到 STDF Workshop 中。

粘贴技巧：

如果你只复制了一个数据，在 STDF Workshop 中需要粘贴到多个单元格，你只需要在 STDF Workshop 中选中指定的单元格(可以是不连续的单元格), 然后按下 Ctrl+V 快捷键粘贴即可。

如果你复制了多个数据(如：Excel 中多个单元格的数据)，在 STDF Workshop 中，选中你需要粘贴区域的第一个单元格，然后按下 Ctrl+V 快捷键粘贴即可。此操作用于复制粘贴一列的多个连续单元格。由于 STDF Workshop 中展示的记录的字段是列，所以都是按列复制粘贴数据的。



GDR 记录的修改

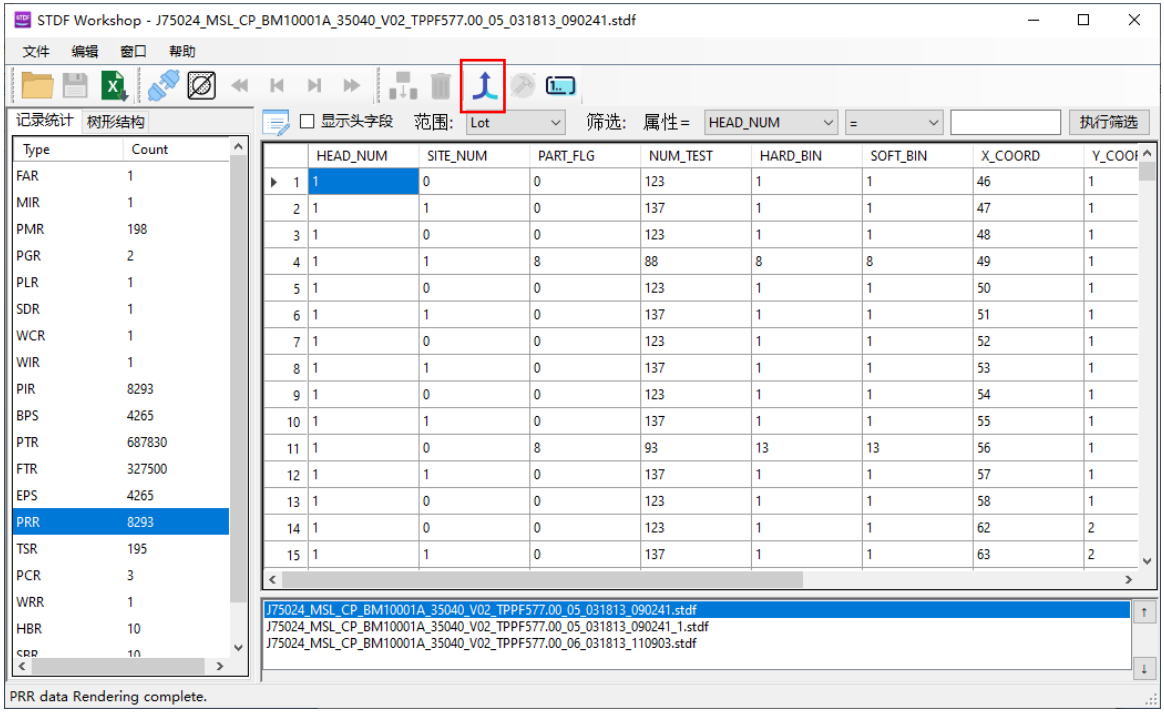
GDR 是一种非常特殊的记录，它的长度不固定，同时数据类型也不固定，所以在 STDF Workshop 中，它的内容展示为如下格式，编辑的时候请务必保证**内容**和**类型**匹配 (参看 STDF SPEC) 。*注意：FLD_CNT 不需要编辑，STDF Workshop 会自动根据 GEN_DATA 的内容更新 FLD_CNT。*

GEN_DATA: *10=_STEPP_PRARM* | *1=2* | *2=0*

GEN_DATA: Type=Value | Type=Value | Type=Value

合并 STDF

当打开多个 STDF 文件之后并解析完成后，工具栏的合并按钮 就变为可用状态，点击之后可以把当前打开的 STDF 文件合并成一个文件，其中 HBR, SBR, TSR, PCR 以及 WRR, MRR 中信息会被重新计算并更新。



修复 STDF

当打开的 STDF 文件不完整的时候，工具栏上的修复按钮 就会变为可用状态，点击之后会自动在 STDF 末尾添加缺少的记录，完成后点击保存按钮即可。

注：在修复之前可以检查一下最后一个 Insertion 的数据，如果不完整且包含 PRR，那么需要手动删除最后一个 Insertion 的数据，保存为一个新的 STDF，然后再导入进来修复。

如果最后一个不完整的 insertion 不包含 PRR，那么 STDF Workshop 在修复的时候会自动舍弃最后一个 Insertion 的数据。

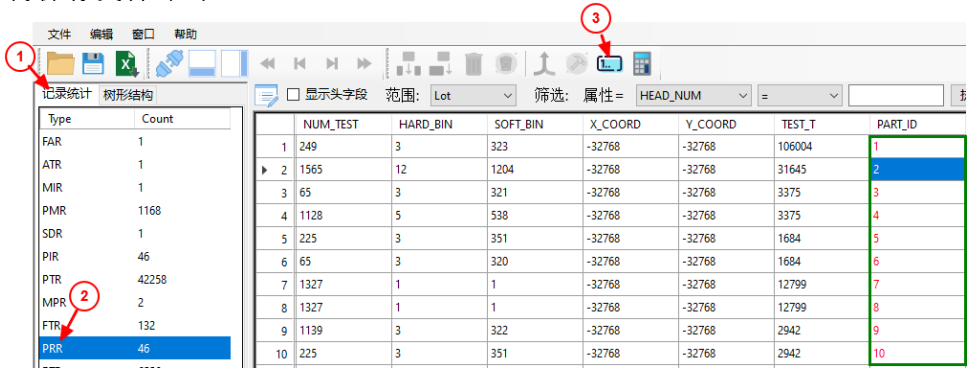
重新计算 Summary 记录

在删除和添加 touch down 记录或者修改过 PRR 的 Bin 数据之后，Summary 记录(HBR, SBR, PCR, TSR)不会自动更新，这会导致数据的前后不一致。STDF Worshop 提供了工具可以重新计算并更新 Summary 记录，点击工具栏上的重新计算按钮 即可，你会发现 HBR/SBR/PCR/TSR 会相应更新并被标记为红色，然后保存到新的 STDF 文件就可以了。

重新序列化 Part_ID

我们在合并多个 STDF 文件到一个文件后，或者删除了某些 Die 的数据之后，STDF 中每颗 die 的 PART_ID 将变得不连续，合并后的 STDF 中 PART_ID 还会有重复。这时候你可以使用 STDF Workshop 的 PART_ID 序列化工具来重新设置 PART_ID, 是他们连续。

操作方式，打开 STDF 之后，在[记录统计]视图里面点击 PRR 来显示所有 PRR 内容，然后点击工具栏的按钮 :重新编号所有的 PART_ID(留意 PART_ID 列内容的变化), 然后再保存文件即可。



3
记录的创建/选择/复制/删除

- 创建记录
- 选择记录和记录组
- 选择直到当前位置的连续记录
- 选择当前 **SITE** 的所有记录
- 复制选中的记录到当前位置
- 移动选中的连续记录到当前位置
- 复制选中的 **PART** 记录到当前 **Insertion**
- 删除记录和记录组
- 删除选中的记录
- 在 **Insertion** 中删除指定 **SITE** 的所有记录
- 在表格视图中选择和删除记录

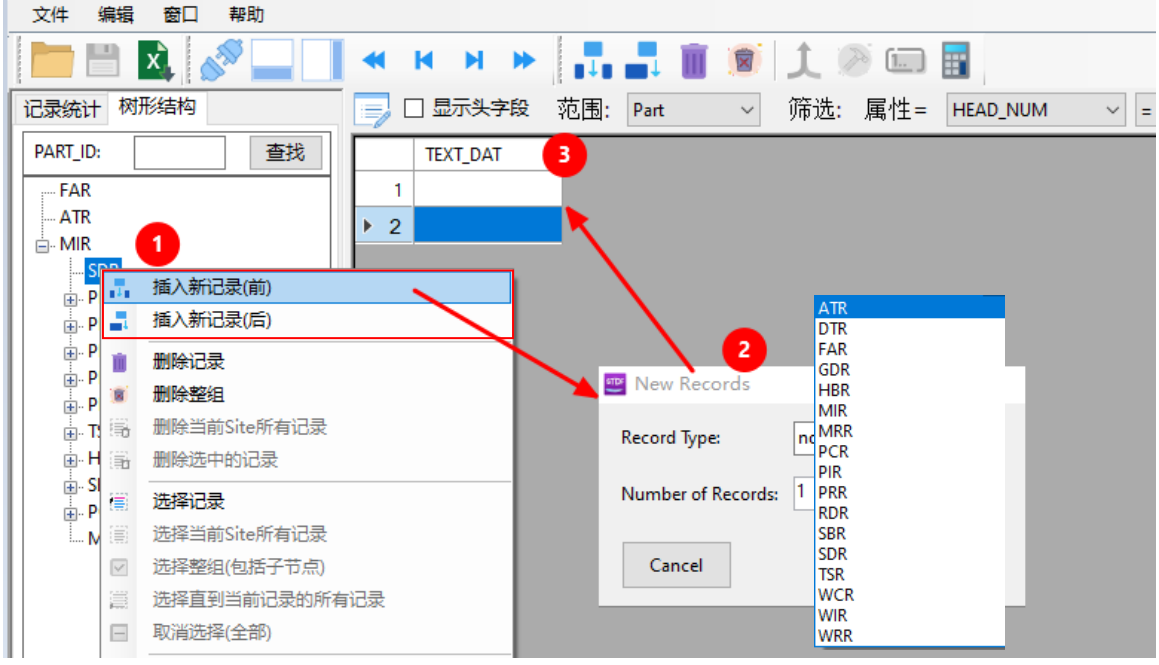
说明

此章节主要介绍记录的创建，选择，复制和删除，本章节的操作主要集中在树形结构的右键菜单中。

新建记录

新建记录的时候，你需要在树形结构中定位需要添加记录的位置，然后在右键菜单中点击[插入新纪录]，在弹出的窗口中选择记录类型和数量。确定后，会在当前记录的前面创建指定数量的特定记录类型并以红色凸显，此时记录的内容都是空的，需要点击记录然后修改每个字段的内容。同类型的记录可以同时显示在主窗口的表格中，可以通过复制粘贴的功能实现批量更新。

注：右键菜单中可以选择在当前记录的前面还是后面添加新记录。目前可插入的记录不包括数据记录(PTR/FTR/MPR)。

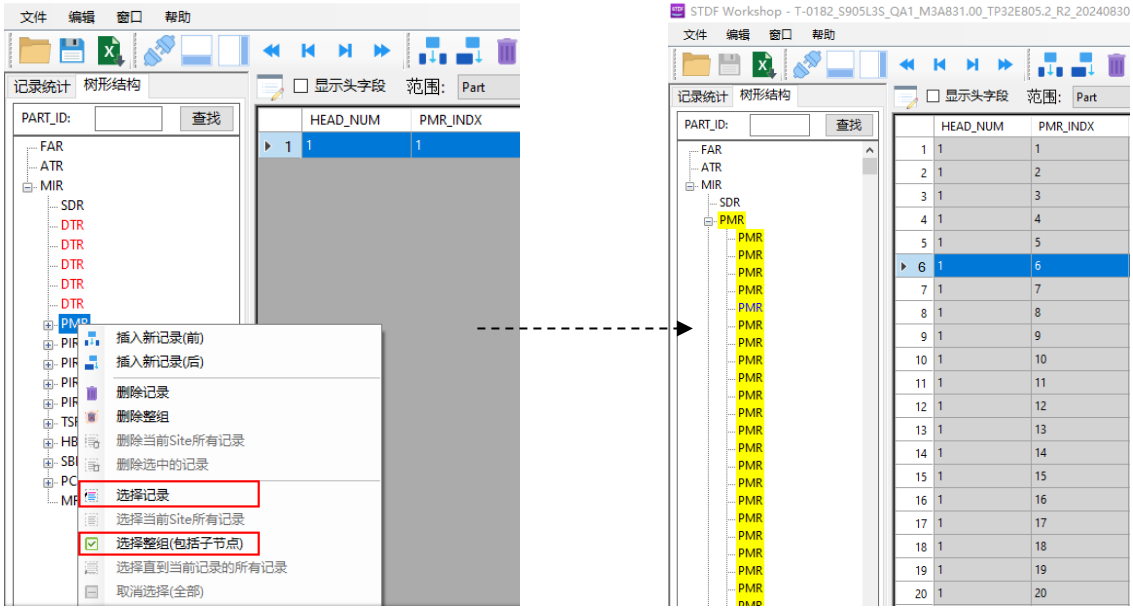


选择记录和记录组

在记录或者记录组上右击，可以选择当前记录或者选择整组记录(当前记录+所有子节点记录)，选中的记录会以黄色高亮显示。

选中记录或者记录组之后可以进行的下一步操作：

- 删除选中的记录
- 复制到指定位置 [把选中的记录复制到其他位置，这个操作可以在不同 stdf 中进行，当然多个 STDF 需要同时被加载到 STDF Workshop 中]
- 移动到指定位置

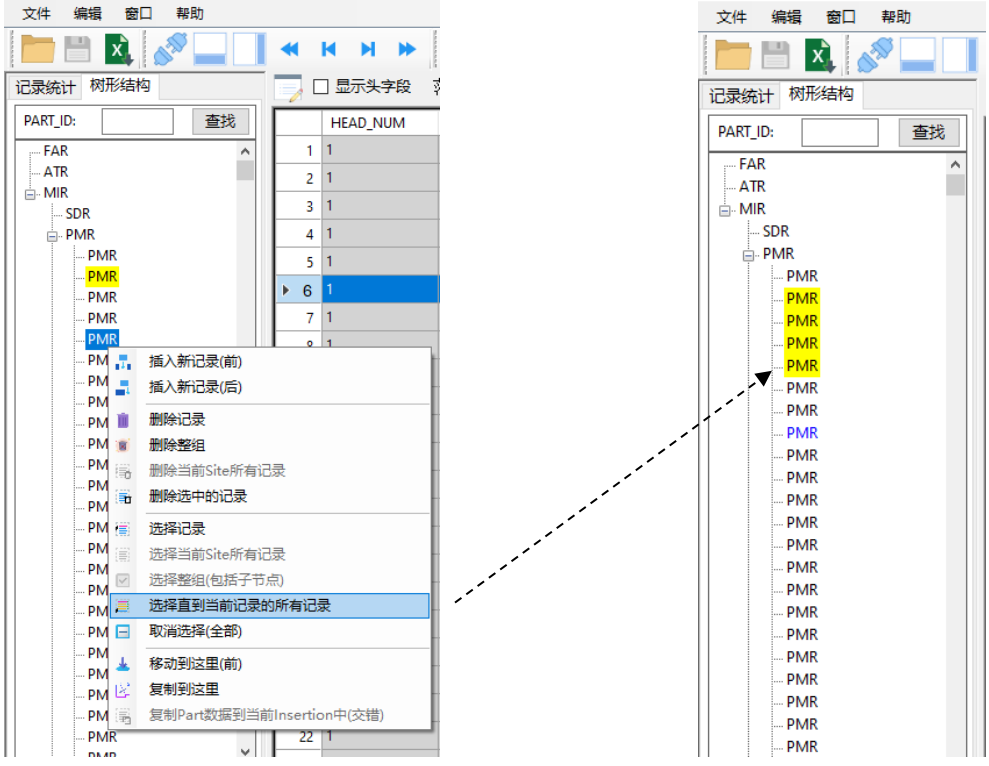


选择直到当前记录位置的所有连续记录

当你选中一个记录之后，再在后面的记录点击右键时，“选择直到当前记录的所有记录”按钮变成可用状态，点击这个按钮之后会选中从第一个选中的记录到当前位置的所有记录。

注意：这个操作只可以在同一级节点中进行，选中记录或者记录组之后可以进行的下一步操作：

- 删除选中的记录
- 复制到指定位置 [把选中的记录复制到其他位置，这个操作可以在不同 stdf 中进行, 当然多个 STDF 需要同时被加载到 STDF Workshop 中]
- 移动到指定位置



选择当前 SITE 的所有记录

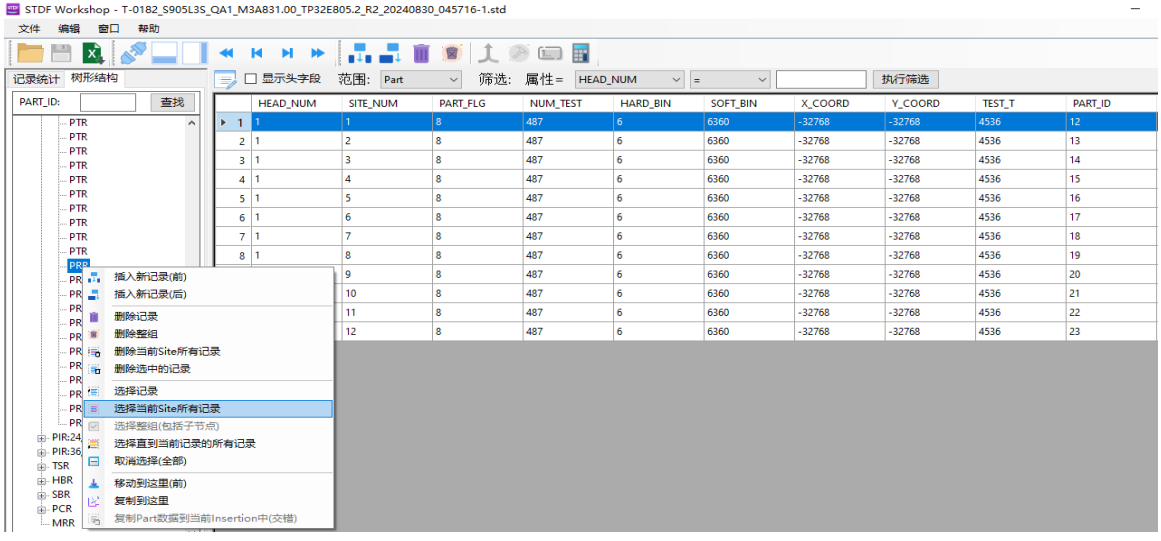
有时候我们需要选择某个 Die 的所有记录，但是目前大多数测试机的多工位测试的数据都是交错的，这就让我们选择某个 site 的所有记录(某个 die 的所有记录)变得非常困难，所以我们开发了这个选择工具，可以让用户点击一次鼠标就可以选中当前 SITE 的所有记录。

这个操作需要先找到对应的 insertion/touch down, 然后在对应的 SITE 的 PIR 或者 PRR 上右击，然后这个选项才会在右键菜单中可用。在 PRR 上使用这个功能的时候可能会更加方便一些，因为 PRR 中有 PART_ID 内容，在 PRR 上单击然后在表格视图中查看内容，如果当前 PRR 是所查找的 DIE 的记录然后再在 PRR 上右击，点击“选择当前 SITE 的所有记录”，然后就可以看到对应 SITE 的记录(PIR/PTR/FTR/MPR/PRR)都已经被选中，并以黄色高亮显示。

注意：由于 DTR/GDR/BPS/EPS 等记录没有 SITE_NUM 信息，所以无法被选中，如果需要这些记录，则需要额外的选中和复制操作

选中当前 SITE 的所有记录之后可以进行的下一步操作：

- 删除选中的记录
- 复制 PART 数据到当前 Insertion [这个就是把选中的那个 die 的数据复制到其他 Insertion 中去，这个操作可以在不同 stdf 中进行，当然多个 STDF 需要同时被加载到 STDF Workshop 中]



取消选择

当我们选中单个或者多个记录之后，再在任意节点上右键时，“取消选择”变得可用，点击这个右键菜单之后，所有之前选中的记录都会恢复到未选中的状态。

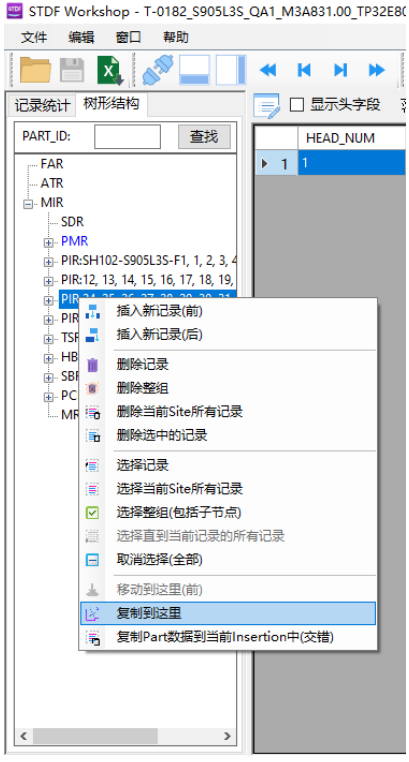
复制选中的记录到当前位置

之前我们讲的各种选择记录的方式，主要目的就是为了把记录复制到其他位置，当有记录被选中之后再到其他记录上右击时，“复制到这里”就会变得可用，点击之后之前选中的所有记录就会被复制到当前记录的前面。

注意：这个操作适用于以下记录选择

[这个操作可以跨 STDF 文件操作，也就是说可以在一个 STDF 文件中选中单个或者多个记录，然后复制到另一个 STDF 文件中的指定位置。注：多个 STDF 需要同时被加载到 STDF Workshop 中]

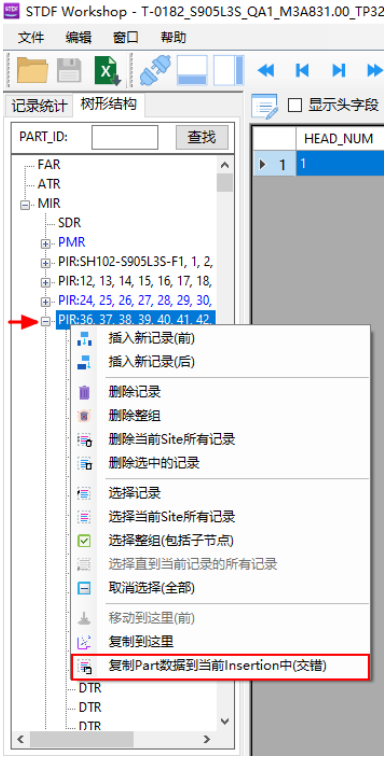
- 选中单个记录
- 选中记录组
- 选中连续的记录块
- 选中不连续的多个记录
- 选中某个 Die 的所有记录



复制 PART 数据到当前 Insertion(交错)

如果你已经选中某个 die 的所有记录，可以通过这个右键菜单工具把它们复制到指定的 Insertion/touch down 中。这个操作会把选中的 die 的 PIR 和 PRR 按照 SITE_NUM 插入到当前 insertion 的 PIR 记录组和 PRR 记录组中去，PTR/FTR/MPR 会把插入到第一个 PRR 的前面，而没有按照 SITE_NUM 去交错插入，主要是为了后续您可能需要复制 DTR/GDR/BPS/EPS 等记录过来。

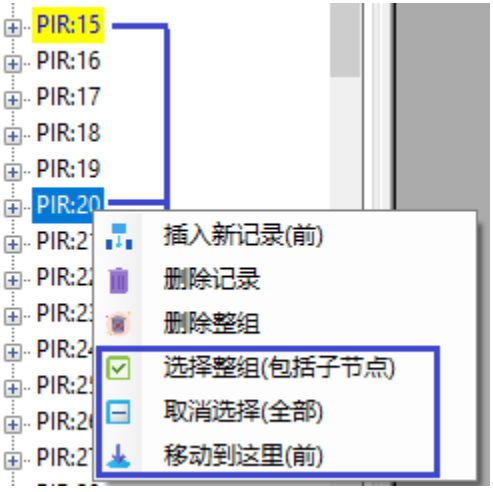
注意：这个右键菜单只有在 insertion 的第一个 PIR (PIR 父节点)点击时才会被启用。这个功能也可以复制其他打开的 STDF 文件中的某个 DIE 的数据到当前 STDF 中的某个 insertion 中。



移动记录

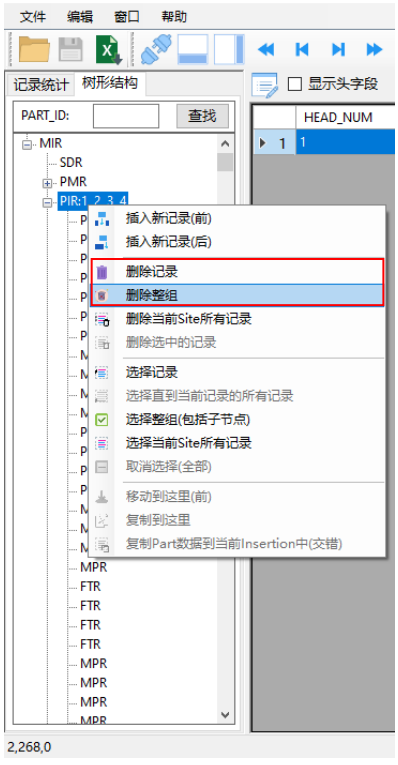
如果你当前选中的是连续的记录或者记录组，就可以通过右键菜单“移动到这里”把它们移动到指定的位置。然后 STDF Workshop 会自动提示保存到新文件，保存之后 STDF Workshop 会自动把新的 STDF 文件加载进来供后续的查看和编辑(这是因为记录/记录组的移动打乱了原先的 STDF 记录序列，需要重新加载并重新创建序列)。

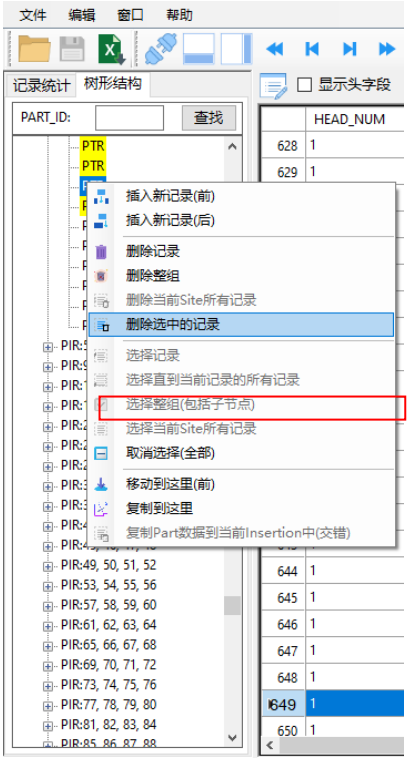
注意：移动记录只在当前 STDF 中进行，并且只能移动选中的连续记录/记录组，如果不连续的记录被选中，这个右键菜单将不可用。



删除记录

如果需要删除某个或者某组记录，直接到树形结构里面找到对应的记录或者记录组，然后在右键菜单中选择 删除记录/删除整租。 被删除的记录会显示为带有删除线的红色标记。此时只是标记了哪些记录需要删除，在你保存 STDF 的时候，新生成的 STDF 中就不会再有这些被删除的记录了。





删除选中的记录

如果当前你已选中了任何记录，右键菜单的“删除选中的记录”就被启用，你可以直接删除刚刚选中的所有记录。

注意：

如果你删除了某些记录之后，你可以继续删除其他记录，也可以修改其他记录的内容。但是如果你做下面的影响记录序列操作之前建议先保存数据并重新加载之后再进行操作。

- 在当前 STDF 文件中移动记录
- 在当前文件中复制记录到不同的位置 (包括复制 PART 记录)
- 从其他文件中复制记录或者复制 PART 记录

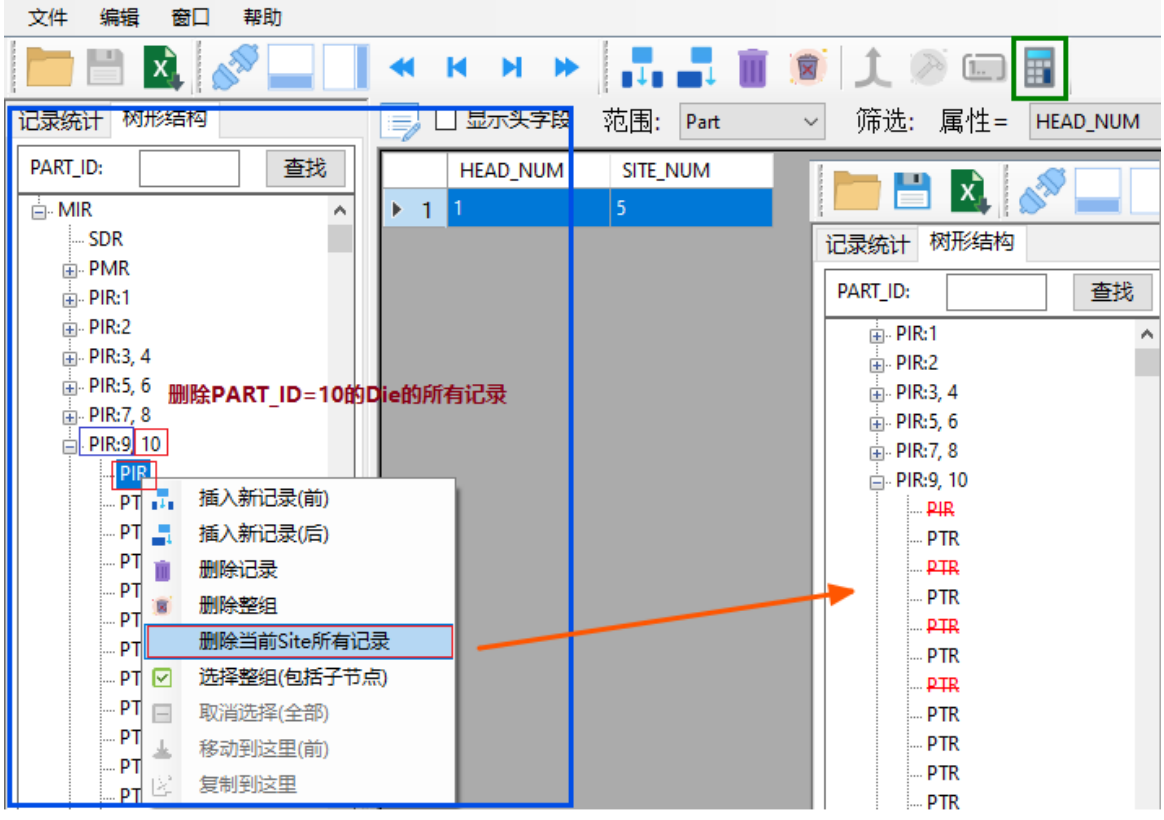
在 Insertion/Touch-down 中删除特定 Site 的所有记录

如果一个 Insertion/Touch-down 中有多个 site 的数据，并且这些数据是交错的， 这是要删除某个 die(site)的所有记录将变得非常困难，你需要手动找到对应 site 的 PIR, PTR, FTR, MPR, PRR 等所有记录，当你的数据有几百个或者几千个测试项的时候，每个 die 就会有成百上千个记录，这样手动删除将变得不太可能。

我们考虑到了这种需求，所以添加了自动删除某个 site 的所有记录的方法。您只需要通过 PART_ID 找到对应 die 所在的 Insertion/Touch-down, 然后在对应的 PIR/PRR 上右击，在右键菜单中点击“删除当前 Site 的所有记录”，STDF Workshop 就会在当前 Insertion 中寻找 SITE_NUM 一样的数据并删除。

不过 DTR/GDR/BPS/EPS 此类记录没有 SITE_NUM 信息，所以不会被删除，需要再次手动删除。

如果你删除了一个或者多个记录，记得在保存数据之前重新计算一下 Summary 记录，点击工具栏的计算器图标即可。(当然你也可以重新序列化一下 PART_ID 让 STDF 的所有 die 的 PART_ID 连续)。



在表格视图中选择/标记/删除记录

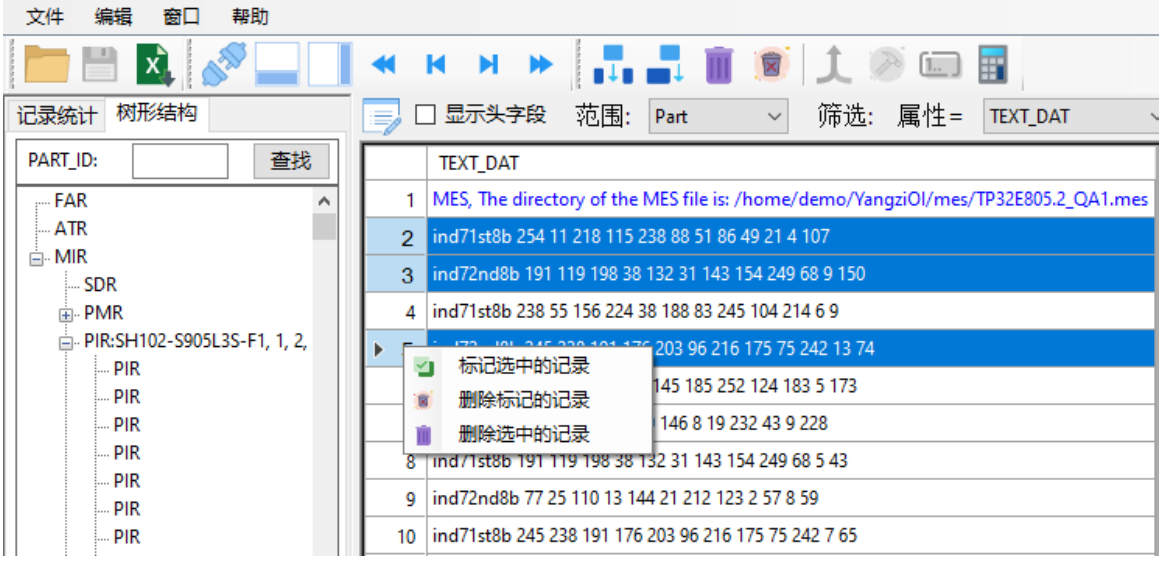
由于树形结构只展示记录的类型，对于 DTR 这类的记录操作的时候需要了解其数据内容才会比较方便，所以 STDF Workshop 同时提供了在表格主视图中进行选择，标记和删除的操作。(当然任何记录都可以在表格视图中进行选择和删除操作的)

选择对应的行(可以连续也可以不连续)，在行标的地方右键，右键菜单就会显示出来，主要包括三个菜单：标记选中的记录，删除标记的记录，删除选中的记录。

- 选择记录 –> 删除选中的记录
- 选择记录 -> 标记记录 -> 删除标记的记录
- 选择记录 -> 标记记录 -> 标记记录(取消标记)

当在表格视图中标记记录时，被标记的记录会自动在树形结构中被选中并以黄色高亮显示，这时候就可以在树形结构中进行复制到指定位置和移动到指定位置(必须选择的时连续记录)的操作。

注意：在表格视图中标记记录以选择对应节点时，如果之前有选择了记录会自动被取消选择。



4
详解记录修改

- **MIR, MRR 和 SDR 记录**
- **WIR, WRR 和 WCR 记录**
- **PIR 和 PRR**
- **PTR, FTR 和 MPR**
- **HBR 和 SBR**
- **TCR 和 PCR**

详解记录修改

在这一章，我们详细介绍一下各种记录如何修改，每个记录的每个字段的含义和数据类型。每各字段的数据都有严格的规定，所以我们会介绍每个字段的的数据的合规性。用户在修改 STDF 的时候必须保证数据的合规性，否则数据就无法更新到对应的记录中去。

MIR 和 MRR

MIR 记录是 lot 的开始记录，包含了所有的 Start_Lot 的信息，每个字段的解释如下图。MRR 是 lot 的结束记录，主要包含 lot 的结束时间。

附：其实 MIR 就是对应测试程序中的 Start_Lot(), 而 MRR 则对应测试程序中的 End_Lot().

SDR 是工位描述记录，存储有工位数量和各工位编号，同时还有各种硬件的类型和编号信息。如：分选机类型和编号，Load board 类型和编号，针卡的编号和类型等等。

MIR记录字段说明		
字段名	字段值	说明
SETUP_T	1639520445	时间，STDF Workshop提供
START_T	1639520569	时间编辑器，不需要直接修改数值
STAT_NUM	1	测试头号：数字，默认为 1
MODE_COD		字符类型，只能一个字符或者空格
RTST_COD	N	
PROT_COD		
BURN_TIM	65535	老化时间，一般不修改
CMOD_COD		字符类型，只能一个字符或者空格
LOT_ID	0	批号，字符串
PART_TYP		产品名，字符串
NODE_NAM	TTK-72	测试机名，字符串
TSTR_TYP	F3	测试机类型，字符串
JOB_NAM	CPx4_V93K_A102	程序名，字符串
JOB_REV	CpA4_93K.flow	程序版本，字符串
SBLOT_ID	SDDEF47C02	子批号，字符串
OPER_NAM	demo	操作员工号，字符串
EXEC_TYP	V93K_A102.flw	测试机软件系统类型，如IGXL
EXEC_VER	7.3.2.10	软件系统版本
TEST_COD	CP1	字符串
TST_TEMP	25	字符串
USER_TXT		
AUX_FILE		
PKG_TYP		
FAMLY_ID		
DATE_COD		
FACIL_ID		
FLOOR_ID		
PROC_ID		
OPER_FRQ		
SPEC_NAM		
SPEC_VER		
FLOW_ID	RP0	
SETUP_ID		
DSGN_REV		
ENG_ID		
ROM_COD		
SERL_NUM		
SUPR_NAM		

MRR记录字段说明		
字段名	字段值	说明
FINISH_T	1639526009	结批时间
DISP_COD		单个字符，一般不修改
USR_DESC		字符串，一般不修改
EXC_DESC		

SDR记录字段说明		
字段名	字段值	说明
HEAD_NUM	1	测试头号：数字，默认为 1
SITE_GRP	2	工位组号，数字，一般不修改
SITE_CNT	4	工位数量，数字
SITE_NUM	1,2,3,4	工位号列表，数字的数组(列表)
HAND_TYP		各硬件类型和编号，字符串
HAND_ID	POS-162	
CARD_TYP		
CARD_ID		
LOAD_TYP		
LOAD_ID		
DIB_TYP		
DIB_ID		
CABL_TYP		
CABL_ID		
CONT_TYP		
CONT_ID		
LASR_TYP		
LASR_ID		
EXTR_TYP		
EXTR_ID		

WIR, WCR 和 WRR

WIR 是 Wafer 的开始记录，存储了 Wafer_ID 和 Wafer 测试开始时间等重要信息。WRR 是 Wafer 结束的记录，同样存储有 Wafer 测试结束时间和 WaferID，还有 Wafer 测试的数量统计信息。

WCR 是 Wafer 测试的配置信息，包含 Wafer Size, Die Size, Wafer Notch, 中心 die 位置等信息。但是绝大多数工厂的 STDF 中只有 WaferNotch 的信息是有效的，其他字段都是空的。

附：其实 WIR 就是对应测试程序中的 Start_Wafer(), 而 WRR 则对应测试程序中的 End_Wafer().

具体每个字段的解释请见下图。

WIR记录字段说明		
字段名	字段值	说明
HEAD_NUM	1	测试头号：数字，默认为 1
SITE_GRP	2	工位组号，数字，一般不修改
START_T	1639520569	开始时间
WAFER_ID	UL8305-01-F7	Wafer ID, 字符串

WRR记录字段说明		
字段名	字段值	说明
FINISH_T	1639526009	结束时间
WAFER_ID	UL8305-01-F7	Wafer ID, 字符串
PART_CNT	4766	Wafer总数量，数字
RTST_CNT	0	复测的数量，数字
ABRT_CNT	4294967295	中断测试的数量, 数字，一般不修改
GOOD_CNT	4537	良品的数量，数字
FUNC_CNT	4294967295	功能测试的数量，数字，不修改
FABWF_ID		其他字符串信息
FRAME_ID		
MASK_ID		
USR_DESC		
EXE_DESC		

WCR记录字段说明		
字段名	字段值	说明
WAFR_SIZ	0	Wafer大小，数字，一般为空
DIE_HT	0	Die高度，数字，一般为空
DIE_WID	0	Die宽度，数字，一般为空
WF_UNITS	0	wafer大小的单位，一般不修改
WF_FLAT	R	Notch方向，单个字符L, U, R, D
CENTER_X	-32768	中心die的X坐标，数字，不修改
CENTER_Y	-32768	中心die的Y坐标，数字，不修改
POS_X		一般为空，不需要修改
POS_Y	D	

PIR 和 PRR

PIR 是 PRR 是成对出现的，表示一个 Unit 的数据块(或者一个 Insertion 的数据块)。在多工位的时候在开头会同时出现多个 PIR，在末尾会出现多个 PRR 记录。在整个数据块中，通过 SITE_NUM 确定当前数据属于哪个 Site (对应哪个 Part_ID)。在 PIR 中只有 HEAD_NUM 和 SITE_NUM, 表示当前记录属于哪个工位；在 PRR 中则保存了当前 unit(当前 site 的 unit) 的测试结果信息，主要包括 PART_ID, Pass/Fail Flag, HBin#, SBin#和 Test time 信息。

附：其实 PIR 就是对应测试程序中的 Test_Start(), 而 MRR 则对应测试程序中的 Bin_Out().

注意：在修改 PRR 数据的时候需要特别留意数据的关联性。如果修改了 PART_FLG 则当前 Unit 会从 Pass 变成了 Fail，或者从 Fail 变成了 Pass，会导致整个 lot 的 Pass/Fail 数量发生改变。需要对应修改文件末尾的 PCR 里面 GOOD_CNT，如果是 CP 数据还需要同时修改 WRR 中的 GOOD_CNT。

如果修改了 HARD_BIN 的信息，则同时需要修改 HBR 对应记录的 HBIN_CNT。例如: HARD_BIN 从 3 改到了 1；则需要找到 HBIN_NUM=3 的 HBR 记录，把 HBIN_CNT 减掉 1；再找到 HBIN_NUM=1 的 HBR 记录，把 HBIN_CNT 增加 1。如果修改了 PRR 的 SOFT_BIN 信息，同样需要修改对应的 SBR 的信息。

PIR记录字段说明		
字段名	**字段值**	**说明**
HEAD_NUM	1	测试头号，数字
SITE_NUM	1	工位号，数字

PRR记录字段说明		
字段名	**字段值**	**说明**
HEAD_NUM	1	测试头号，数字
SITE_NUM	1	工位号，数字
PART_FLG	0	Unit结果的Pass/Fail Flag, 我们提供Flag编辑器，如果改变则需要同时修改PCR和WRR里面的数量信息
NUM_TEST	309	测试项执行的数量，数字
HARD_BIN	1	当前Unit的Hbin #, 数字，如果改变则需同时修改HBR的数量，需要减少原来Bin的数量和增加新的Bin的数量
SOFT_BIN	1	当前Unit的Sbin #, 数字，如果改变则需同时修改HBR的数量，需要减少原来Bin的数量和增加新的Bin的数量
X_COORD	28	当前Unit的X坐标, 数字
Y_COORD	4	当前Unit的Y坐标, 数字
TEST_T	5920	测试时间(ms), 数字
PART_ID	1	Unit的序列号，字符串，一般不修改，除非有特殊需求
PART_TXT	000000_0_0_0	一般用来存储CHIP ID, 字符串
PART_FIX		一般不需要修改

PTR, FTR 和 MPR

PTR，FTR 和 MPR 都是测试项结果的记录，分别表示参数测试项结果的记录 (parametric test)，功能测试项结果的记录(digital test) 和多 pin 参数测试项的结果记录。

其中 FTR 比较简单，因为功能测试都是执行 pattern，而在 STDF 中一般只保存了测试结果 pass/fail 的 flag 和测试项名等信息。我们这里着重介绍 PTR 的修改，MPR 修改可以参照 PTR，只是 MPR 的结果记录是一个数组，需要用户修改到数组中对应的数据，有些复杂，我们以后会提供工具辅助用户修改。

注意：修改 PTR 的 RESULT 字段(测试结果)时，需要用户自行根据 Limit 判定测试项的 Pass/Fail 状态是否发生改变，如果发生了改变需要同时修改 TEST_FLG 信息。同时还需要修改 PRR 对应的 PART_FLG, HARD_BIN 和 SOFT_BIN 信息。由于软件无法知道测试项 fail 的时候对应的 BIN#，就需要用户自己非常了解测试程序，确保修改成正确的 HARD_BIN 和 SOFT_BIN。同时如果 PRR 的记录修改了，还需要同时修改对应的 PCR, WRR, HBR 和 SBR。

数据记录的关联性特别复杂，非常容易改错或者漏掉，所以一般不建议修改。

PTR记录字段说明		
字段名	**字段值**	**说明**
TEST_NUM	2100001	测试项编号，数字，一般不同测试项不可以有相同TEST_NUM
HEAD_NUM	1	测试头编号，数字
SITE_NUM	1	工位号，数字
TEST_FLG	0	测试结果的Flag, 包含Pass/Fail信息，数字，STDF Workshop提供flag编辑器
PARM_FLG	192	一般无需修改
RESULT	4.72039E-07	测试结果，float类型。修改之后，需要结合Limit检查测试项是否Pass/Fail状态发生变化，若变化了，需要同时修改当前PTR的TEST_FLG，同时需要修改对应的PRR的Pass/Fail状态和对应的Hbin, Sbin.
TEST_TXT	AVDD3318_MIPIRX	测试项名称，字符串
ALARM_ID		一般无需修改
OPT_FLAG	14	此flag和limit信息相关，无需修改 (STDF Workshop会自动生成)
RES_SCAL	6	测试结果单位转换因子，数字，一般无需修改
LLM_SCAL	6	下限Limit单位转换因子，数字，一般无需修改
HLM_SCAL	6	上限Limit单位转换因子，数字，一般无需修改
LO_LIMIT	-0.0001	测试项的下限Limit, float类型
HI_LIMIT	0.0001	测试项的上限Limit, float类型
UNITS	A	测试项的单位，字符串
C_RESFMT	%7.9f	此5项不需要考虑，一般为空。
C_LLMFMT	%7.9f	
C_HLMFMT	%7.9f	
LO_SPEC	0	
HI_SPEC	0	

这些额外信息只出现在测试项第一次出现的PTR中

HBR 和 SBR

HBR 和 SBR 是 STDF 末尾的 Summary 信息，如果 STDF 没有正常结束，一般 HBR 和 SBR 是缺少的。HBR 和 SBR 的结构非常类似，所以我们这里就只介绍 HBR。

这里要特别注意 HEAD_NUM=255 的 Bin 记录是所有 site 的汇总记录。

HBR记录字段说明		
字段名	**字段值**	**说明**
HEAD_NUM	1	测试头编号，数字。HEAD_NUM=255时代表当前是所有site汇总信息
SITE_NUM	1	工位号，数字 (有时SITE_NUM也可能是255， 所有site汇总的情况)
HBIN_NUM	1	HBIN的编号，数字
HBIN_CNT	1163	HBIN的数量，数字
HBIN_PF	P	HBin的Pass/Fail Flag, 单个字符 (P/F)
HBIN_NAM	PASS	HBin名称，字符串

HEAD_NUM	SITE_NUM	HBIN_NUM	HBIN_CNT	HBIN_PF	HBIN_NAM
1	1	1	1163	P	PASS
1	2	1	1152	P	PASS
1	3	1	1115	P	PASS
1	4	1	1107	P	PASS
255	**0**	**1**	**4537**	**P**	**PASS**
1	1	2	4	F	OS_FAILED
1	2	2	21	F	OS_FAILED
1	3	2	32	F	OS_FAILED
1	4	2	44	F	OS_FAILED
255	**0**	**2**	**101**	**F**	**OS_FAILED**
1	2	3	2	F	DC_FAILED
1	3	3	1	F	DC_FAILED
255	**0**	**3**	**3**	**F**	**DC_FAILED**
1	1	4	23	F	DFT_FAILED
1	2	4	14	F	DFT_FAILED
1	3	4	35	F	DFT_FAILED
1	4	4	34	F	DFT_FAILED
255	**0**	**4**	**106**	**F**	**DFT_FAILED**
1	1	5	2	F	IP_FAILED
1	2	5	4	F	IP_FAILED
1	3	5	7	F	IP_FAILED
1	4	5	6	F	IP_FAILED
255	**0**	**5**	**19**	**F**	**IP_FAILED**

TSR 和 PCR

TSR 记录是用来统计测试项的总执行数量(EXEC_CNT)和失效数量(FAIL_CNT)，可以通过 FAIL_CNT/EXEC_CNT 来计算每个测试项的失效比率，从而知道 Majorfailure。TSR 有按照每个 site 统计的记录，也有所有 site 的汇总记录(HEAD_NUM=255)。TSR 与 HBR, SBR 一样是 summary 的一部分，也出现在 STDF 的末尾位置，如果 STDF 没有正常结束，可能缺少 TSR.

TSR记录字段说明		
字段名	**字段值**	**说明**
HEAD_NUM	1	测试头编号，数字。HEAD_NUM=255时代表当前是所有site汇总信息
SITE_NUM	1	工位号，数字 (有时SITE_NUM也可能是255， 所有site汇总的情况)
TEST_TYP	P	测试项类型，单个字符， P/F/M分别对应PTR, FTR, MPR
TEST_NUM	0	测试项编号，数字，对应PTR/FTR/MPR的TEST_NUM
EXEC_CNT	14166	当前测试项的执行总数量，数字
FAIL_CNT	2358	当前测试项Fail的数量，数字，这个可以看Major failure
ALRM_CNT	4294967295	一般都是无效值，不用管
TEST_NAM	DVS_READ:DVS_FLAG	测试项名称，字符串
SEQ_NAM		测试项的Sequencer名称，字符串
TEST_LBL		测试项的标签，一般不用
OPT_FLAG	200	一般不需要修改
TST_TIM	4.56291E-05	这几项都是统计参数，很多STDF都为无效值，一般不用管。
TST_MIN	0	
TST_MAX	1	
TST_SUMS	11782	
TST_SQRS	1981.47839	

PCR 是统计每个工位的总数(PART_CNT)和良品数(GOOD_CNT)，可以根据 GOOD_CNT/PART_CNT 来计算每个工位的良率。当 HEAD_NUM=255 时代表所有 site 的汇总记录。PCR 也出现在 STDF 的末尾，如果数据未正常结束，可能缺少 PCR。

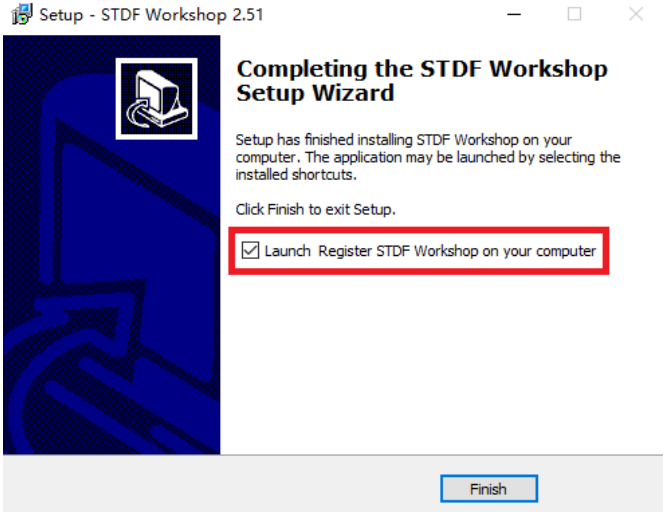
PCR记录字段说明		
字段名	**字段值**	**说明**
HEAD_NUM	1	测试头编号，数字。HEAD_NUM=255时代表当前是所有site汇总信息
SITE_NUM	1	工位号，数字 (有时SITE_NUM也可能是255， 所有site汇总的情况)
PART_CNT	1192	当前工位的总数量，数字
RTST_CNT	0	当前工位的复测数量，数字，一般不用修改
ABRT_CNT	4294967295	当前工位异常停止测试的数量，数字，一般不用修改
GOOD_CNT	1163	当前工位良品数量，数字，结合PART_CNT计算工位的良率
FUNC_CNT	4294967295	当前工位功能测试数量，数字，一般都是无效值，不用管

5
其他

- 软件安装与激活
- **SWS** 自动脚本
- 利用计划任务自动执行 **SWS** 脚本

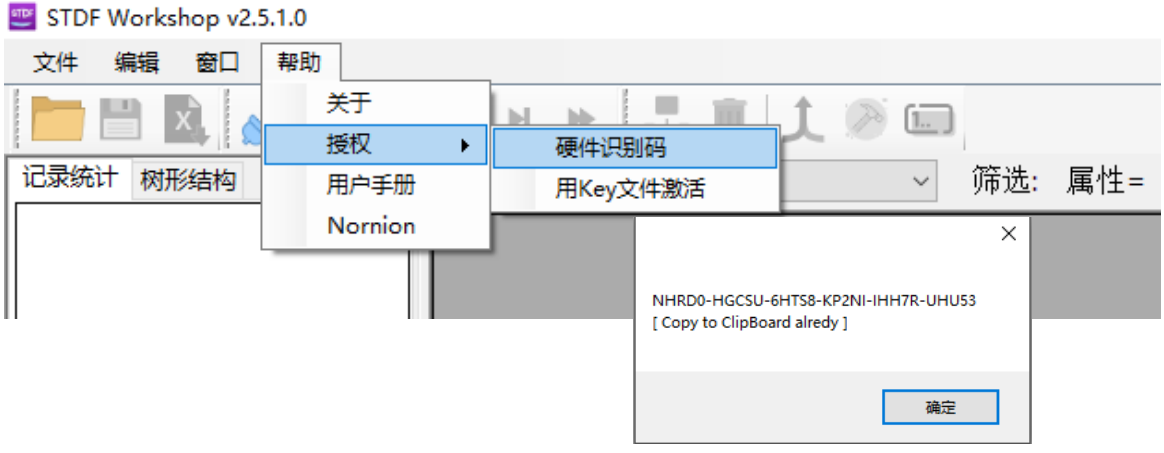
软件安装

请到 http://www.nornion.com/download.aspx 下载 STDF Workshop 最新安装软件。安装时需要有管理员权限，安装结束时需要在您的电脑上注册软件，请勾选运行。

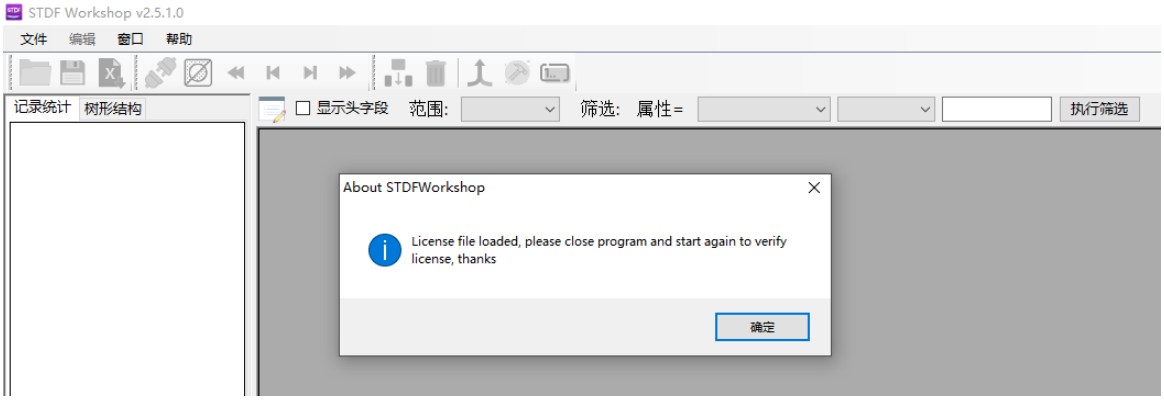


软件激活

安装好软件之后，打开 STDF Workshop, 通过菜单 [帮助] – [授权] – [硬件识别码] 来获得硬件识别码，并发送到 support@nornion.com 获取授权文件。每个用户都可以获得 30 天的使用授权来了解 STDF Workshop 的功能和特性。



获得授权文件后(.dat 文件)， 按照下面步骤激活。打开 STDF Workshop 菜单 [帮助] – [授权] – [用 key 文件激活]， 然后在打开的窗口中选择.dat 授权文件。加载好授权文件之后，关掉 STDF Workshop 然后重新打开即可。



自动更新

每次打开 STDF Workshop 之后，如果检测到新版本发布会自动提示您是否需要更新最新版本，确定之后等待下载完成会自动运行安装程序，和首次安装一样点击下一步完成安装。

SWS 自动脚本

SWS (STDF Workshop Script)脚本是 STDF Workshop 提供的简单脚本功能，可以用来编程并实现批量修改 STDF。在脚本编辑区修改脚本时，编辑器会自动设置不同的颜色显示。

紫色：内置命令

蓝色：内置对象/方法/属性

黑色：变量/常数等

SWS 使用流程如下：

- 编辑 SWS 脚本
- 编译 SWS 脚本并确认没有错误
- 打开文件(可以选择多个文件)执行编辑好的 SWS 脚本
- 或者打开文件夹执行编辑好的 SWS 脚本

注意：所有编辑好的 STDF 文件都会被另存到新的目录

可以将编辑好的 SWS 脚本另存为.sws 文件，然后用 exec_sws.exe 和 windows 计划任务后台定时执行 sws 脚本来实现自动 monitor 目录并把目录中的 STDF 自动修改并另存到指定的目录。(参看下一节内容)

SWS 语法介绍：

SWS 是一种简单脚本，通过提供的有限命令来实现自动打开文件，修改属性，执行方法和另存为的功能，同时提供了 FOR_EACH 循环和 IF 条件判断。SWS 的格式要求非常严格，所以需要用户严格按照要求编写脚本，建议每次都是在示例脚本的基础上修改而不是从完全从头编写。

重要：所有 SWS 命令都是大写字母，并且在命令、赋值符号、比较符号、数值之间必须要有空格。

FOR_EACH_FILE f

文件循环命令，循环每个 STDF 文件，f 代表 STDF 文件 (建议不要修改这个循环变量)，最后会以 **END_FOR_EACH** 结束。这个命令是 SWS 的最外层命令，整个 SWS 脚本必须以“**FOR_EACH_FILE f**”开始，并以“**END_FOR_EACH**”结束。在这个循环内部更新需要修改的记录的属性和执行相应的方法。

EXECUTE_METHOD

执行方法命令，在 SWS 中 STDF 文件“ f “ 提供可执行方法，用来做文件解析和文件另存为的操作等。每一种 Record 也有一个 r.Delete()方法可以执行。

f.Extract()　解析 STDF 文件，这个方法没有参数，这个方法一般是 FOR_EACH_FILE 循环里面的第一条需要执行的命令。

f.SplitFileName(“_”) 拆分文件名并把文件名拆分的内容保存在 f.FileNameFields 数组中，以便可以把文件名的特定字段的内容写入 STDF 的 MIR 或者其他的记录字段中。需要把文件名的分隔符通过参数传入方法。

注意：在调用 f.FileNameFields[3] 数组时，需要用户自己考虑 index 指向正确数组位置，确保 index 不会超出索引范围。

~~f.Repair() 方法可以用来修复不完整的 STDF, 然后执行 SaveAs()方法把修复好的 STDF 文件保存到新的目录中。~~

f.RepairAndSave(“C:\share”) 是用于修复 STDF 的新的方法，f.Repair()已经不可用了。新的方法直接修复并保存到目标目录，不需要再调用 f.SaveAs()方法。

f.UpdateSummaryRecords() 方法用来重新计算 HBR/SBR/PCR/TSR 的数量，并且更新到 STDF 中去，然后执行 SaveAs()方法把修复好的 STDF 文件保存到新的目录中。

f.SaveAs(“folder_name”) 把修改后的 STDF 另存到新的目录中(原来的 STDF 文件不会被修改), 需要把目录通过参数的形式传入。也接受没有引号的参数 f.SaveAs(folder_name)。确保在 FOR_EACH_FILE 循环内的最后一条命令是 f.SaveAs 命令，否则修改好的 STDF 不会被保存。

所有 RECORD 都支持 r.Delete()方法，用来删除当前记录。

调用外部接口获取 lot 信息并写入 STDF → f.GetInfoFromExternalInterface()

有时候需要根据 lot_id 或者其他信息从外部接口(例如：数据库，web server 等)查询并获取相关信息，比如产品名，handler id 等信息。再写入 STDF 文件。

这时候需要准备 External Interface DLL, 用户可以参看 STDF Workshop 安装目录下的“dev”目录的源代码，更新 SEIL.dll 从外部获取信息，然后返回 Dictionary<string, string>字典。再 sws 脚本中可以调用接口并传递参数来获取外部返回的字典，然后用这些信息更新 STDF.

在 sws 中调用编辑好的外部接口 SEIL.dll (位于 interface 目录下)，需要类似下面的代码，先传递参数(多个参数需要传递多次)，然后再调用 GetInfoFromExternalInterface()方法从外部获取信息字典，信息会保存在 f.ExternalInfoDict 中，再用字典里面的信息更新 STDF 相关记录的字段。由于返回信息是字典，所以在指定 Key 时请自行确认 key 是否存在，如果不存在的话，sws 执行会报错。

```
EXECUTE_METHOD f.AddExternalInterfacePara("EXLOT")       #Arg1 defined in external interface lib
EXECUTE_METHOD f.AddExternalInterfacePara("HOTCOLD") #Arg2 defined in external interface lib
EXECUTE_METHOD f.GetInfoFromExternalInterface()          #Get information from External interface
UPDATE_PROPERTY f.MIRs[0].LOT_ID = f.ExternalInfoDict["LOT"]
UPDATE_PROPERTY f.MIRs[0].PART_TYP = f.ExternalInfoDict["PRODUCT"]
```

UPDATE_PROPERTY

更新属性值，这个命令用来更新指定记录(Record)的属性值。STDF 解析之后在 STDF Workshop 中都是以记录数组的方式存储的，所以修改 STDF 其实就是修改指定记录的属性。

一般情况下每个 STDF 只有一个 MIR 和一个 MRR，所有修改 MIR 记录的属性时，可以直接指定 index 为 0 (也就是数组中的第一个记录)，比如：f.MIRs[0].LOT_ID=”new lot id”。

STDF 中有很多记录的数量非常庞大，所以在修改属性的时候需要用到 FOR_EACH_REC 循环和 IF 条件判断来实现。

设置记录属性的值，可以直接在 SWS 中使用字符串常数或者也可以指定其他属性值(目前只支持使用文件名拆分数组 f.FileNameFields[3])

Note: 关于 STDF 有哪些记录，每种记录有哪些属性，需要用户对 STDF 的结构有一些了解，可以同时参照此帮助文件的第一章 STDF 结构和第四章 详解记录修改。

在 STDF Workshop 中同类型的记录会被保存在同样的记录数组中，记录数组的名称都是记录类型后面加一个 ”s”, 例如：MIRs, PIRs, PTRs, HBRs 等。

FOR_EACH_REC

记录循环命令，这个命令用来循环记录数组中的每一条记录，在这个循环中可以修改每条记录的属性，或者借助 IF 条件判断来修改符合条件的记录的属性。建议用 r 来作为循环变量代表每个记录。这个循环命令同样以 END_FOR_EACH 命令结束.

记录循环嵌套，FOR_EACH_REC 可以嵌套以实现复杂的功能，每个 FOR_EACH_REC 的循环变量必须不一样，否则编译会出错。

示例代码（scripts\tsr_255_updates.sws）：用来把 TSR 记录中 SITE_NUM!=255 的记录的 EXEC_CNT 和 FAIL_CNT 复制到同样 TEST_NUM 的 SITE_NUM==255 的记录中去，并且把 SITE_NUM==255 的记录的 TEST_NAM 复制到 SITE_NUM!=255 的对应记录中(注意循环变量 r，t)。

```
FOR_EACH_REC r IN TSRs
        IF r.SITE_NUM == "255"
            FOR_EACH_REC t IN TSRs
                IF t.TEST_NUM == r.TEST_NUM && t.SITE_NUM != "255"
                    UPDATE_PROPERTY t.TEST_NAM = r.TEST_NAM
                    UPDATE_PROPERTY r.EXEC_CNT = t.EXEC_CNT
                    UPDATE_PROPERTY r.FAIL_CNT = t.FAIL_CNT
                    UPDATE_PROPERTY r.HEAD_NUM = "255"
                END_IF
            END_FOR_EACH
        END_IF
END_FOR_EACH
```

IF 条件判断命令，这个命令用来判断条件，一般是用来检查记录的属性是否符合特定条件，来筛选记录，以便修改符合条件的记录的属性。请务必在条件判断符号左右都留空格，支持的条件判断符号如下：== , != , > , >= , < , <=。 **IF** 命令最终以 **END_IF** 命令结束。

```
IF  f.MIRs[0].RTST_COD != "R"
```

多条件 IF 语句，IF 语句支持多个条件可以用 AND(&&)或者 OR(||)连接，其中&&的优先级高于||的优先级，所以在多条件语句中&&会优先计算，最后在计算||来完成所有的条件判断。

优先级示例：A && B || C && D　*（实际代码）*

→　*(A && B) || (C && D)*　*（执行顺序，实际代码中不支持括号）*

实际代码演示（*scripts/default.sws*）：

```
IF f.MIRs[0].MODE_COD == "D" && f.MIRs[0].PROT_COD == "E"
    UPDATE_PROPERTY f.MIRs[0].RTST_COD = "R"
END_IF
```

说明：SWS 的编译功能并不能检查出所有的语法和逻辑错误，需要用户在编写 SWS 脚本的时候特别留心。如果有任何问题可以联系 support@nornion.com. 在 STDF Workshop 的安装目录下的 scripts 文件夹中可以找到我们提供的 sws 常用实例脚本。

利用计划任务自动执行 SWS 脚本

既然 STDF Workshop 可以执行 SWS 脚本自动修改 STDF，是否可以设置 Windows 计划任务自动执行 SWS 脚本呢？答案是肯定的，只需要利用我们提供的 exec_sws.exe (SWS 脚本执行器)就可以实现这个功能。

exec_sws.exe 是一个命令行工具，调用它的时候需要指定一些参数，可以使用 windows 计划任务执行这个命令配上相应的参数即可。也可以把调用的命令保存在 BAT 文件中，然后用计划任务执行这个 BAT 文件 (可以参看 STDF Workshop 安装目录下的 exec_sws.bat)

exec_sws.exe -s .\scripts\example.sws **-d** c:\temp\input **-a** c:\temp\archive **-e** c:\temp\error **-g** c:\temp\log

参数介绍：

-s 用来指定 sws 脚本文件，路径中如果有空格请使用 “ ”

-d 用来指定 STDF 输入目录，脚本执行器会从这个目录寻找文件并为每个文件执行 sws 脚本

-a 用来指定 STDF 文件 archive 目录，原来的 STDF 文件会被备份到此目录

-e 用来指定错误目录，STDF 执行 sws 时如果遇到错误，STDF 会被移动到这个目录

-g 用来指定 log 的目录，这个参数可选，如果指定，sws 执行器会把详细 log 写入这个目录

注意：

STDF 输出目录(修改后的保存目录)是在 sws 脚本中定义的 *f.SaveAs(“c:\temp\output”)*。在调用 exec_sws.exe 的时候参数指定的顺序可以随意。

在创建计划任务的时候，请务必设置其“**起始目录**”为 STDF Workshop 的安装目录，exec_sws.bat 也需要放在这个目录。当然 sws 脚本可以放在任何地方，只要用 ”-s” 指定完整路径即可。