



Search our Knowledge base, Community, and more...



什么是推荐的异常和压缩设置

适用于: Data Archive- 1,3.4,3.1.287,3.1.288,3.1.290,3.2.0.0,3.2.332,3.2.357.11,3.2.357.17,3.2.357.8,3.3.361.96,3.3.361.98,3.3.362.47,3.3.362.49,3.3.362.5,3.3.362.55,3.3.362.63,3.3.362.79,3.4.363.68,3.4.363.74,3.4.363.75,3.4.363.97,3.4.363.98,3.4.364.32,3.4.370,3.4.370.52,3.4.370.54,3.4.370.76,3.4.370.88,3.4.370.96,3.4.375,3.4.375.0,3.4.375.38,3.4.375.79,3.4.375.80,3.4.375.84,3.4.375.91,3.4.375.99,3.4.376.01,3.4.380,3.4.380.36,3.4.380.70,3.4.385,3.4.385.59,3.4.385.77,3.4.390,3.4.390.16,3.4.390.18,3.4.390.28,3.4.395,3.4.395.72,3.4.395.80,3.4.400.1162,3.4.405.1198,3.4.410.1213,3.4.410.1256,3.4.410.1284,3.4.415,3.4.415.1188,3.4.420.1182 减去

👍 3 🗨️ 0

✅ **已验证的解决方案** - 更新于 2019年 7月9日 - 英语

问题

内容来自3226OSI8。
什么是推荐的压缩和例外设置？

环境

PI Data Archive (所有版本)
Windows

解

异常报告

在将值发送到PI Data Archive之前，会在接口节点上进行异常报告。此过程过滤掉噪声，从而减少PI Data Archive与接口节点之间的通信 (I/O) 负担。

OSIsoft建议将异常偏差设置为略低于仪器精度的值。另一个准则是将Exception Deviation (ExcDev) 设置为Compression Deviation (CompDev) 的一半。

- **异常最大值 (ExcMax)** : 异常之间的最大时间跨度，以秒为单位表示。
- **异常最小值 (ExcMin)** : 异常之间的最小时间跨度，以秒为单位表示。
- **异常偏差 (ExcDev)** : ExcDev属性指定值在被视为重要值之前可能与先前值的差异。这是一个死区，当超出时会导致异常。您可以使用ExcDev或ExcDevPercent属性为PI标记点指定异常偏差。

压缩测试

快照子系统在PI Data Archive上执行压缩测试，以提高数据存储效率并节省磁盘空间。但是，如果您使用的是PI Buffer Subsystem，则在将数据发送到PI Data Archive之前，会在接口节点上执行压缩（或者在PI Collective的情况下将其展开到多个PI Data Archives）。

您的目标是设置压缩以有效地存储数据，而不会牺牲其准确性。更高效的数据存储可以在同一磁盘空间上捕获更长时间的在线数据，并实现更快的分析和数据检索。这里的想法不是存储PI可以通过外推周围事件实质上重建的事件。

要检查压缩比，可以使用pi \adm命令提示符 (1) 中的“piartool-ss”命令，或者检查Windows性能监视器 (2) 中以下两个PI性能计数器的比率：

- (1) “快照事件”：“发送到队列的事件”
- (2) PI Snapshot Subsystem计数器：“Snapshot / sec: ”Queued Events / sec”

理想情况下，此比率应介于10: 1到100: 1之间，这意味着每10-100个事件进入队列，一个发送到存档。最低限度应该在3: 1到10: 1之间，特别是在点数较多（超过50,000点）的系统或频率数据非常高的系统上。

如果比率为1: 1到2: 1，则系统在给定时间段内存档磁盘使用率可能非常低效。

关闭压缩

建议不要关闭压缩。但是，它适用于实验室或手动输入的点或累计点，其中每个事件都很重要。如果将Compressing设置为0 (OFF)，则归档所有异常（无压缩）。

关闭压缩的另一种方法是将Compressing属性设置为1 (ON)，将Compression Deviation (CompDev) 属性设置为0 (OFF)。在这种情况下，不会存档连续的相同值（或沿着线完美对齐的值）。这在归档使用方面比关闭压缩要高效得多。

例如，如果将值“高”发送到存档，后跟另一个“高”值，则不会添加第二个值。作为另一个例子，如果值“1”，“2”，“3”，“4”的序列以偶数时间间隔出现，则2和3不归档，因为它们的值可以从1到4之间的斜率计算。

注：在压缩中= 1和CompDev = 0的与在很长一段时间内的非常小的偏差的情况下，可能的是，一个稍微偏离事件将不被存档，因为压缩斜率是如此之小。

“CompMin”和“CompMax”

“CompMin”和“CompMax”是指存档中事件之间的时间限制。如果自上次记录的事件以来的时间小于该点的压缩最小时间，则不记录新事件。

其他资源

- OSIsoft YouTube视频 - 例外和压缩快速摘要 (<http://www.youtube.com/watch?v=6-scv3oQ7Kk>)
- OSIsoft YouTube视频 - 例外和压缩全部详细信息 (<http://www.youtube.com/watch?v=89hg2mme750>)
- PI Live Library - PI Data Archive - 异常报告和压缩测试 (<https://livelibrary.osisoft.com/LiveLibrary/content/en/server-v5/GUID-96C7E7C1-F5A9-441F-B0C7-Q32DE1DEF65C>)

- [PI Data Archive系统管理指南 \(https://www.osisoft.com/myosisoft/\)](https://www.osisoft.com/myosisoft/)

原因

异常报告和压缩测试的设置允许您调整PI标记点以实现最高效率。异常报告会从一个点的接口节点到服务器的数据流，压缩测试会影响一个点的归档中的数据存储效率。

PI Data Archive可以存储大量数据，但存储有意义的数据非常重要。在PI Data Archive中存储过多数据会影响性能。当客户端在很长时间内调用检索压缩数据或执行汇总计算时，很可能在第一次调用时从磁盘读取大部分归档数据。与从存储器读取相比，这是一种昂贵的操作。如果只有有意义的数据存储在存档中，则可以将更大的数据时间范围存储在存储器（读缓存）中，以便更快地访问

没找到你要找的东西?



[打开一个新案例](#)



[查看联系选项](#)