

综合对比表格增加实况数据。

增加准误。

准误是对修订预报而言。

当修订预报的温度与实况数据的温度相差绝对值小于等于1摄氏度时，为“准”，否则为“误”。

天气现象不能直接拿来与实况中的降水量比较。所以需要取降雨量与实况中的降水量进行比较

—

* 修订预报不可以使用ProductTime（即报文中的YYGGgg字段），虽然这个时间是固定的。汤军义说需要使用文件名上的创建时间最接近06：15的那个报文。

```
<Group table="XDYB_Data" source="{SpanForecast}" ignoreCondition="product
Time>' {...ProductTime}' and modifyFlag>='{...ModFlag}' and StationID='{..Stati
onID}' and StartTime='{...StartTime}' and ForecastSpan={TimeSpan}" updateCo
ndition="productTime='{...ProductTime}' and StationID='{..StationID}' and Start
Time='{...StartTime}' and ForecastSpan={TimeSpan}">
```

从DZB.Database.xml配置文件上看，如果站点、起报时间、预报时效相同的时候，这时就看报文里的ProductTime了。1.如果ProductTime也相同，那么就是更新数据。相当于，多次修改修订报，但只有最后一次修订报文件数据会记录在数据库中，Info表中只记录了报文文件名相关信息。（Bug：在抓取到的报文顺序混乱的情况下，有可能用较早的报文数据覆盖最新的报文数据！）；2.如果报文中的ProductTime比数据库的ProductTime小并且ModifyFlag也比数据库中的ModifyFlag小或相等，那么直接忽略。

也就是说，数据库Data表中保存的数据总是最新的报文数据，历史报文数据仅在Inof表中记录文件名相关信息。

所以按照产品时间来取是OK的。

综合对比表格中的实况数据展示为雨量（虽然列名叫xx小时/晴雨）。

实况数据的雨量转成两级：有雨（大于等于0.1毫米），无雨（小于0.1毫米）；

同样地天气现象也需要转成这两级：

0:晴,1:多云,2:阴,3:阵雨,4:雷阵雨,5:雷阵雨并伴有冰雹,6:雨夹雪,7:小雨,8:中雨,9:大雨,10:暴雨,11:大暴雨,12:特大暴雨,13:阵雪,14:小雪,15:中雪,16:大雪,17:暴雪,18:雾,19:冻雨,20:沙尘暴,21:小雨-中雨,22:中雨-大雨,23:大雨-暴雨,24:暴雨-大暴雨,25:大暴雨-特大暴雨,26:小雪-中雪,27:中雪-大雪,28:大雪-暴雪,29:浮尘,30:扬沙,31:强沙尘暴。

归类为有雨的（阵雨、雷阵雨、雷阵雨并伴有冰雹、雨夹雪、小雨、中雨、大雨、暴雨、大暴雨、特大暴雨、阵雪、小雪、中雪、大雪、暴雪、小雨-中雨、中雨-大雨、大雨-暴雨、暴雨-大暴雨、大暴雨-特大暴雨、小雪-中雪、中雪-大雪、大雪-暴雪），无雨的（晴、多云、阴、雾、冻雨、浮尘、扬沙、强沙尘暴），

然后进行比较。

* 实况数据站可能与预报站不一样。根据“各市预报技巧评分代表站一览表”中的对应站点使用正确站点。

AWSBase表字段名解析

列名	数据类型	允许NULL值	含义	单位
Station	nvarchar(50)	否	站点编号	编码
Time	datetime	否	发报时间	UTC
Wind2D				

Bug沿伸：由于产品时间为06：45与10：30的报文都是在11：00统一抓取，有可能先入库了10：30的报文。这时后入库06：45的报文会因为达到ignoreCondition条件而忽略06：45的报文。造成缺失。

但是观察FileTree2的GetUnprocessedItem，在返回前做了排序。因此不会出现所担心的问题。

TransferLog表中是有Success字段的，用于判断是否成功传输。但是在GetUnprocessItem中使用的db.HasProcessed方法里面却没有考虑Success字段。只要TaskType与Source字段一致就认为是已经处理过的任务。

按理说，Success为0时表示该Item尝试处理过但是失败了。所以下一次理应再次尝试，直到成功为止。

因此，db.HasProcessed方法中对Item是否处理过的判断应该还要加上Success = 1.