

从世行暂停 2020 年报告发布探析营商环境指标优化的路径

Analysis on the optimization path of business environment indicators from the world bank's suspension of 2020 Report

【摘要】 本文以世界银行重新审计营商环境报告违规数据为案例，分析了事件原因以及世界银行在营商环境数据抽样采集过程中存在的问题，对事件后续发展进行了推演，并籍此提出了克服“沉默投票者”偏差，应用大数据建立我国营商环境评估指标的方式方法以及应对建议。

【abstract】 In this paper, taking the world bank's re audit of business environment report violation data as an example, this paper analyzes the causes of the event and the problems existing in the sampling and collection process of business environment data by the world bank, deduces the follow-up development of the event, and puts forward the methods and Countermeasures to overcome the "silent voter" deviation and apply big data to establish China's state-owned business environment assessment indicators Discussion.

【关键词】 营商环境 数据违规 指标体系 抽样调查 沉默投票者

【Key words】 Business environment; Data violation; Index system; sampling survey; Silent voters

2020 年 8 月 27 日，世界银行（以下简称“世行”）宣布，因数据变化导致营商环境计算方法与数据不符，将暂停发布 2020 年营商环境报告，并对过去 5 年的营商环境报告进行重新审核。上述事件引发广泛关注，8 月 29 日，中国商务部官网通过《因数据问题，世行暂停发布 2020 年营商环境报告》一文对上述事件进行了报道。准确评估上述事件

对我国营商环境排名的影响进而以此为契机，坚定理论自信，建立健全我国整体营商环境指标体系，切实发挥科学指标体系作为优化营商环境的指挥棒、红绿灯、指南针作用，需要详细分析造成世界银行数据不一致、数据违规的原因，从而为搭建我国营商环境指标体系提供经验借鉴和路径选择。

一、世行商业数据违规声明及营商环境报告有关情况

营商环境是指一个企业在筹备设立到清算终止的完整周期中，遵照一国或地区的法律法规，所需要付出的时间和费用成本的总和。良好的营商环境是建设现代化经济体系、推动高质量发展、实现产业转型升级的有力支撑。一个国家或地区营商环境的优劣，事关企业竞争力和盈利能力。因此，自 2003 年世界银行逐年发布《营商环境报告》以来，营商环境受到世界各国广泛关注，逐渐成为各国政府吸引外资、优化社会治理的重要参照。2018 年 11 月 1 日，外交部发言人陆慷就曾明确表示：“世界银行营商环境评价客观、公正，对中国营商环境评价反映了中国政府不断完善营商环境的坚定承诺和有效举措”。

（一）我国高度重视营商环境优化工作。

营商环境是一个国家或地区经济软实力的重要体现，关系着市场配置资源的效率和主要流向，党和政府高度重视营商环境优化工作。2018 年 4 月 8 日，习近平总书记在亚洲博鳌论坛发表主旨演讲时强调：“营商环境就像空气，空气清新才能吸引更多外资”。2017 年 6 月 13 日，李克强总理

在全国深化“放管服”改革电视电话会议上指出：“营商环境就是生产力，要进一步深化改革、扩大开放，加快打造国际化、法治化、市场化营商环境”。根据世界银行提供的《2019年营商环境报告》，我国营商环境便利度在全球 190 个经济体排名中大幅提升，从 2018 年全球排名第 78 位上升至全球第 46 位。2020 年 7 月 27 日，世行发布《中国优化营商环境的成功经验——改革驱动力与未来改革机遇》，专题介绍推广了中国近年来在改革优化营商环境领域取得的巨大成就，以及在开办企业、获得电力、办理建筑许可等多个领域推出的具体改革举措，为全球其他经济体优化营商环境改革提供参考借鉴。

（二）世行营商环境评估整体情况。

1.世行营商环境报告发布情况。世界银行 2001 年成立 Doing Business 小组 (简称 DB 小组) 负责全球经济体营商环境指标体系的构建，并于 2003 年发布了第一份营商环境报告——《2004 年营商环境报告》，至今已连续发布了 16 年。但是，由于世界各国及主要经济体政治经济体制不同，世行对营商环境的评价指标适用程度并不一致，客观上造成评价指标的局限性，使得国际上许多研究者对 DB 小组报告的评价体系提出了批评，这也促进了世行评价指标的不断调整完善，样本采集也越来越丰富。例如：2003 年，DB《营商环境报告》只有 5 个指标，涵盖世界上 133 个经济体；而到了 2019 年，则扩大至 11 个指标，涵盖了世界上 190 个经济体。总体而言，相比较经济学人智库 (EIU) 全球营商

环境（5 年发布 1 次）、科尼尔全球城市指数（已连续发布 11 年）等排名，世行《营商环境报告》因样本经济体覆盖面广，发布周期稳定、测评时间相对久远，成为较具国际影响力的营商环境评估体系，并对全球资本流动、行业聚集、货物贸易、服务贸易等带来实质性影响。

2.世行营商环境指标采集情况。世行将营商环境理解为：企业从筹备设立到清算终止的全生命周期过程中，依据所在地区的法律法规，从事生产经营活动所付出的时间和金钱成本的总和，并进一步将企业生命周期设定为开办企业、获得电力、登记财产、执行合同、办理破产等 11 个方面，从而分别计算营商环境便利度得分（Ease of Doing Business Score）和营商环境便利度排名（Ease of Doing Business Ranking）。在具体采集样本时，世行把大多数经济体的最大商业城市的问卷调查数据作为该经济体的实际定量数据，并将截至 2013 年，人口超过 1 亿的 11 个经济体（孟加拉国、巴西、中国、印度、印度尼西亚、日本、墨西哥、尼日利亚、巴基斯坦、俄罗斯联邦和美国）的第二大商业城市的数据也计算在内，采取按照城市人口比例的方式加权平均计算。

（三）世行商业数据违规声明有关情况。

2020 年 8 月 27 日，世行在其官网就《营商环境报告》商业数据违规发表声明如下：

《营商环境报告》发布 17 年来，一直是衡量国家营商环境的的宝贵工具。相关指标和方法的设计不考虑某一国家的特殊性，而是以有助于改善整体商业环境为目标。在 2017

年 10 月和 2019 年 10 月发布的《2018 年营商环境报告》和《2020 年营商环境报告》中，发现了一些数据违规变更行为，致使数据与实际商业状况不一致。出于对数据和分析的完整性和公正性的考虑，我们立即采取以下行动：一是对最近发生的五项业务数据进行系统的评估。二是要求世行审计部门对开展业务所需的数据收集和审查流程以及保障数据完整性的控制措施进行审计。三是根据调查结果纠正受违规行为影响较大的国家数据。世行董事会以及受数据违规影响较大的国家已经听取了有关情况的汇报。因此，在相关评估完成前，《营商环境报告》将暂停发布。

二、相关原因分析

从世行的官方声明中，我们尚且不能判断违规商业数据的情况。但是，从世行营商环境数据抽样采集方法和样本中，我们可以探析世行营商环境采集数据存在的短板以及由此造成的结果偏差。

（一）“沉默投票者”偏差较难避免。

1.折戟 2016 年美国总统大选的抽样调查。在大数据时代，受随机误差和粗大误差的影响，没有全量数据支撑的监测评估很容易因为漏掉了“沉默的投票者”而出现重大误判。2016 年美国大选，美国主流媒体和主要民调机构的预判，几乎无一漏网地错了。以《纽约时报》《赫芬顿邮报》为首的主流纸媒、以准确性强著称的知名民调网站，都在大选投票日早上给出了希拉里 70%-90%的获胜概率，几乎众口一词的预测希拉里板上钉钉的胜出。后来经过调查，主流媒体

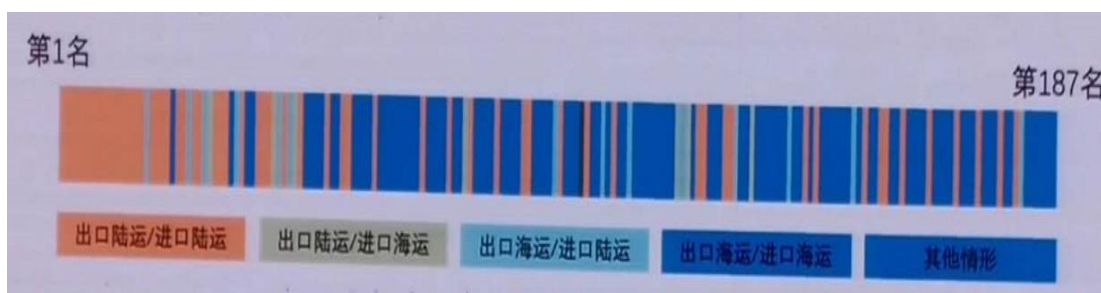
和主要民调机构之所以做出错误的预判，是因为大量支持特朗普的“沉默的投票者”并没有在民调中表达自己的真实想法，或者没有纳入统计调查的范围。这件事情也引发了美国上下对“数据新闻主义”的广泛质疑。这也说明，即使是信息技术、大数据技术最为发达的美国，以及接受能力、学习能力相对较强的新闻媒体和民调网站，以传统的抽样调查为主的数据采集方式，依然难以保证数据的客观准确。

2.世行抽样调查难免落入“窠臼”。与美国主流媒体和民调机构几乎无一例外预测错 2016 年美国总统大选结果相反，微软公司开发的大数据分析平台——微软小娜 Cortana 准确预测了美国 51 个选区中 50 个地区的选举结果。表面上看，微软小娜 Cortana 与美国主流媒体和民调机构是预测结果的差异，实际上是大数据时代全量数据分析方法与小样本时代抽样调查方法的效果差异。因此，世行《营商环境报告》虽然是比较重大的经济评估行为，但是与美国总统大选相比，抽样调查范围、对象类别、样本审核、数据筛查、异常剔除等要相对较弱，整体测算结果的代表性和客观性很难超过美国民调机构以及主流媒体的水平，较难避免抽样调查中“沉默投票者”偏差的影响。

(二) 世行营商环境报告主要的短板。

1.国家主体未进行类型区分。受世界各国政治经济体制及经济社会发展水平差异的影响，DB 报告中的各项指标在不同经济体中的适用程度并不一致，而世行不对国家主体进行类型区分的做法，也导致不少国家对此提出质疑。例如：

世行把诸如法国与比利时等同一关税区之间的贸易统计为跨境贸易，致使“通关时间”、“通关费用”等指标基本为0，这也造成营商环境便利度排名前30位的国家里，28个是“纯陆运”的经济体（具体见下表：《2020年营商环境报告》不同运输方式经济体跨境贸易排名情况）。



备注：数据来自2019年12月11日召开的“贸易便利化暨口岸营商环境”发布活动

2.样本采集覆盖面较窄。长期以来，世行营商环境数据采集以有限责任公司为主，对发展中国家存在的大量的其他经济实体覆盖不足。同时，调查问卷的采样数量较窄，显著性和代表性受到较大影响，一定程度上降低了《营商环境报告》中营商环境便利度得分的客观性和真实性。

经济体	年度							
	2013	2014	2015	2016	2017	2018	2019	2020
中国	8	9	10	10	9	18	69	55
美国	6	8	16	6	6	10	7	8
德国	3	2	2	2	2	3	3	3
日本	4	3	12	6	17	14	18	17
英国	1	3	5	6	4	3	4	3

备注：数据来自2019年12月11日召开的“贸易便利化暨口岸营商环境”发布活动

从《营商环境报告》跨境贸易答卷人数可以看出，虽然2020年相比2013年，答卷数量出现了较大幅度的增长，但是相较于中国、美国、德国、日本等贸易大国的进出口体量

和经营主体的数量，基数仍然较小，覆盖面值得商榷，整体结果的客观性、公正性、真实性有所降低。

3.答卷人的参与度有待提高。世行《营商环境报告》采取调查问卷的方式采集数据，答卷人对所调查业务的参与度和熟练程度决定了整体指标结果的质量。尤其是国际贸易体量已经十分巨大的今天，通关过程中所需的业务能力、专业知识、法律法规、经验储备已经十分巨大，非长期浸染或长期从事本项业务的专业人士很难给出客观合理的评价。这也一定程度上造成了指标给分随意、长期数值不变等削弱营商环境便利度排名公信力的情况。例如：我国 2020 年跨境贸易 55 名答卷人中，具有通关专业背景的人员约为 30 人，其余则为律师、会计师等，对通关时间、通关费用、单证管理等具体业务了解有限。

经济体	年度				
	2016	2017	2018	2019	2020
美国	92.01	92.01	92.01	92.01	92.01
德国	91.77	91.77	91.77	91.77	91.77
日本	86.51	86.51	86.51	86.51	85.94
英国	93.76	93.76	93.76	93.76	93.76
韩国	92.52	92.52	92.52	92.52	92.52
新加坡	89.3	89.3	89.57	89.57	89.57

备注：数据来自 2019 年 12 月 11 日召开的“贸易便利化暨口岸营商环境”发布活动

从部分经济体跨境贸易得分变化情况可以看出，美国、德国、英国、韩国 5 年评分完全一致，新加坡、英国虽然有所变化，但也存在多年保持不变的情况。上述情况与美国特朗普政府近年来采取的贸易保护主义、单边主义、加征关税措施等国际整体营商环境波动不尽相符。

(三) 后续世行处置方式及发展展望。

面对 190 余个经济的营商环境评估，世行很难短时间内完成指标统计口径和计算方法的调整。因此，对商业数据违规的审计和处置，世行大概率有两种处置方式。一种是，若存在大面积的问卷调查数据与实际商业情况不符，世行扩大样本选取空间，并重新进行问卷测评的方式来弥补。但考虑到世行声明是 2018 年和 2020 年两个营商环境报告中发现数据违规行为，时间跨度较大，重新进行问卷抽样调查的成本相对较高，且客观性和准确性也较难保证。另一种是，根据审计结果，对个别存在数据违规嫌疑的调查问卷进行修正或者剔除，然后按照合格样本空间对各国营商环境便利度得分进行重新计算，然后重新计算各国营商环境便利度排名。考虑到问卷调查时间限制、审计成本，世行采取第二种方式的可能性较大。

三、下一步应对建议

营商环境是激发市场活力，加快法制化、市场化、国际化建设的重要抓手。建立健全高水平的与当前高质量发展相适应的营商环境指标体系，有利于发挥指标考核指挥棒作用，切实聚焦营商环境优化的难点、堵点，推动我国整体市场运行效率和国际竞争能力的大幅提升。但是，我国政府迄今尚未建立起统一的营商环境评价指标体系，一定程度上导致了我国对世界《营商环境报告》的依赖。正如习近平总书记在 2013 年访问莫斯科时所指出的“鞋子合不合脚，自己穿着才知道。一个国家的发展道路合不合适，只有这个国家

的人民才最有发言权。”因此，切实掌握自身发展的主动权、话语权，建立与我国国情相适应的营商环境指标体系已经是迫在眉睫。

（一）密切跟踪世行对违规数据审计情况。

世行《营商环境报告》虽然是众多国际营商环境评估报告中的一种，且存在诸多不足，但因其出台时间长、指标设计全面科学、发布周期稳定，其采集数据被世界经济论坛《全球促进贸易报告》和《全球竞争力报告》、美国国务院《投资环境报告》、英国政府《营商环境国别报告》、印度政府《印度营商便利度报告》等众多国际组织、政府部门广泛引用，也成为跨国公司全球产业布局、国际资本跨境流动的重要参考依据，是国家软实力的重要体现。因此，建议国家口岸管理办公室、商务部等部门高度重视、密切跟踪、及时响应世行审计及后续处置情况，为我国营商环境整体优化提供经验借鉴。

（二）探索应用大数据建立新型考核指标。

进入大数据时代，随着以大智云物移（大数据、人工智能、云计算、物联网、移动互联网）为代表的新一代信息技术的大量普及应用，不仅让很多信息实现了实时数据化和网络化采集，同时又可以快速高效地进行分析运算处理。这让人们可以花费更低的成本、更小的代价就能做到全量分析，而不用像传统抽样调查一样，随机选取部分样本进行分析运算。这种全量分析已经显著不同于以点带面、以偏概全的样本分析模式，而是由全部数据反映客观事实，以更好地帮助

分析人员准确的提炼数据背后隐藏的规律。因此，和抽样调查等传统社会调查方式相比，用大数据建立的考核指标最突出的特点在于不需要抽样，从而大幅减少了抽样过程中出现的粗大误差和随机误差。例如：笔者利用 SQL 大数据建模，建立的口岸通关时间和海关通关时间，将运算逻辑、数据提取、深度挖掘等全部集成进入程序，由机器自动完成，较好的规避的“沉默投票者”偏差（具体见附录）。因此，建议我国政府按照市场准入、商贸环境、基础设施、政府效率、科技创新、政策供给、经营成本、便利水平、法治环境、融资信贷等国内外营商环境评估实践，利用大数据技术，打通部门信息壁垒和数据鸿沟，积极构建跨部门、跨领域的营商环境指标，实现指标的人工智能采集、加工和应用。

（三）建立科学完善的营商环境指标体系。

单一的考核指标很容出现数据“注水”的情况。例如：2005 年起，教育部将毕业生就业状况与高校招生挂钩，规定连续两年就业率低于 50% 的高校，严格控制招生规模，连续三年低于 30%，必须减少招生，甚至停止招生。2011 年再次加码，规定连续两年低于 60% 的专业，调减招生计划甚至停招。为了满足就业率，很多没有找到工作的同学不得不“假就业”，通过皮包公司盖章、签署虚假三方协议、填报自媒体、淘宝店主、电子竞技工作者等，完成就业考核。因此，建议发改委、商务部等政府部门，借鉴国内外营商环境测评的理论和实践探索，建立科学合理的符合我国国情的营

商环境指标体系，避免一刀切和机械评估，为我国整体营商环境持续优化提供量化指引。

参考资料：

【1】李杰.营商环境评价指标体系研究述评与展望[J].当代经济.2019 (07)

【2】王美舒.营商环境评估：国际实践及其中国启示[J]. 师大学.2018 (01)

【3】曲宁.世界银行《营商环境报告》梳理及中国营商环境述评[J].商业研究.2019 (13)

【4】崔建高.从通关常数探析口岸营商环境持续优化的新着力[J].中国口岸科学技术.2020 (01)

【5】崔建高.中美贸易战企业增加成本监控评估模型研究[J].中国口岸科学技术.2019 (05)

【6】钱佳慧 韩滨阳 罗晶钰 沈超. 我国区域营商环境评价指标体系研究[J].区域经济.2019 (09)

【7】因数据问题，世行暂停发布 2020 年营商环境报告. 中华人民共和国商务部官网.

<http://www.mofcom.gov.cn/article/i/jyjl/k/202009/20200902999040.shtml>

附录：口岸通关时间和海关通关时间脚本：

```
SELECT NC.YUE AS YUE,NC.I_E_FLAG AS I_E_FLAG,NC.KOUAN1 AS KOUAN1,NC.HAIGUAN1 AS HAIGUAN1,OC.KOUAN AS  
KOUAN,OC.HAIGUAN AS HAIGUAN  
from (SELECT GC.YUE AS YUE,GC.I_E_FLAG AS I_E_FLAG,avg(GC.KOUAN) AS KOUAN1,avg((case when (GC.HAIGUAN)  
<'0'  
then '0'
```

```

else (GC. HAIGUAN)
end)) AS HAIGUAN1
from (SELECT (case when datediff((DC. JIESHUSHIJIAN), (DC. STARTE), 'ss') < '0'
then '0'
else datediff((DC. JIESHUSHIJIAN), (DC. STARTE), 'ss') / 3600
end) AS KOUAN, (case when datediff((DC. STARTE), (DC. D_DATE), 'ss') > '0'
then datediff((DC. JIESHUSHIJIAN), (DC. STARTE), 'ss') / 3600
else datediff((DC. JIESHUSHIJIAN), (DC. D_DATE), 'ss') / 3600
end) AS HAIGUAN, DC. DECL_PORT_CNAME AS DECL_PORT_CNAME, DC. NIAN AS NIAN, DC. YUE AS YUE, DC. I_E_FLAG AS
I_E_FLAG, DC. TRAF_MODE_CNAME AS TRAF_MODE_CNAME, DC. DECL_PORT AS DECL_PORT
from (SELECT CC. DECL_PORT_CNAME AS DECL_PORT_CNAME, CC. NIAN AS NIAN, CC. YUE AS YUE, CC. I_E_FLAG AS
I_E_FLAG, CC. ENTRY_ID AS ENTRY_ID, (case when substr((CC. STARTE), 12, 8) like '%00:00:00%'
then dateadd((CC. STARTE), 12, 'hh')
else (CC. STARTE)
end) AS STARTE, CC. JIESHUSHIJIAN AS JIESHUSHIJIAN, CC. TRAF_MODE_CNAME AS TRAF_MODE_CNAME, CC. DECL_PORT AS
DECL_PORT, CC. D_DATE AS D_DATE
from (SELECT (Case when (BC. I_E_FLAG) like '%I%' and (BC. CREATE_DATE81) is null
Then (BC. CREATE_DATE80)
when (BC. I_E_FLAG) like '%I%' and (BC. CREATE_DATE81) is not null
Then (BC. CREATE_DATE81)
when (BC. I_E_FLAG) like '%E%' and (BC. CREATE_DATE81) is null
Then (BC. CREATE_DATE70)
when (BC. I_E_FLAG) like '%E%' and (BC. CREATE_DATE81) is not null
Then (BC. CREATE_DATE81)
else '0'
end) AS JIESHUSHIJIAN, (case when (BC. I_E_FLAG) like '%I%' and (BC. CREATE_DATE81) is null and
(BC. CREATE_DATE80) is null
then '1'
when (BC. I_E_FLAG) like '%E%' and (BC. CREATE_DATE70) is null and (BC. CREATE_DATE81) is null
then '2'
else '0'
end) AS PANDUAN, BC. DECL_PORT_CNAME AS DECL_PORT_CNAME, BC. NIAN AS NIAN, BC. YUE AS YUE, BC. I_E_FLAG AS
I_E_FLAG, BC. ENTRY_ID AS ENTRY_ID, (Case when (BC. I_E_FLAG) like 'E'
and (BC. ARRIVE_TIME) is not null
then (BC. ARRIVE_TIME)
when (BC. I_E_FLAG) like 'E'
and (BC. ARRIVE_TIME) is null
then (BC. D_DATE)
when (BC. REAL_ARRIVE_DATE) is not null
and (BC. I_E_FLAG) like 'I'
then (BC. REAL_ARRIVE_DATE)
when (BC. I_E_FLAG) like 'I'
and (BC. GEN_DATE) is not null
then (BC. GEN_DATE)
when (BC. REAL_ARRIVE_DATE) is null
and (BC. GEN_DATE) is null
and (BC. I_E_FLAG) like 'I'
then (BC. I_E_DATE)
else '0'
end) AS STARTE, BC. D_DATE AS D_DATE, BC. DECL_PORT AS DECL_PORT, BC. TRAF_MODE_CNAME AS TRAF_MODE_CNAME
from (SELECT distinct XB. ENTRY_ID AS ENTRY_ID, XB. I_E_FLAG AS I_E_FLAG, XB. DECL_PORT_CNAME AS
DECL_PORT_CNAME, XB. NIAN AS NIAN, XB. YUE AS YUE, XB. CREATE_DATE70 AS CREATE_DATE70, XB. CREATE_DATE80 AS
CREATE_DATE80, XB. D_DATE AS D_DATE, XB. I_E_DATE AS I_E_DATE, XB. CREATE_DATE81 AS CREATE_DATE81, XB. TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, XB. DECL_PORT AS DECL_PORT, XB. REAL_ARRIVE_DATE AS REAL_ARRIVE_DATE, XB. GEN_DATE AS
GEN_DATE, ZB. ARRIVE_TIME AS ARRIVE_TIME
from (SELECT PB. ENTRY_ID AS ENTRY_ID, PB. I_E_FLAG AS I_E_FLAG, PB. DECL_PORT_CNAME AS
DECL_PORT_CNAME, PB. NIAN AS NIAN, PB. YUE AS YUE, PB. CREATE_DATE70 AS CREATE_DATE70, PB. CREATE_DATE80 AS
CREATE_DATE80, PB. D_DATE AS D_DATE, PB. I_E_DATE AS I_E_DATE, PB. CREATE_DATE81 AS CREATE_DATE81, PB. TRAF_MODE_CNAME AS

```

```

TRAF_MODE_CNAME, PB.DECL_PORT AS DECL_PORT, PB.MANIFEST_ID AS MANIFEST_ID, PB.BILL_NO AS BILL_NO, PB.GEN_DATE AS
GEN_DATE, RB.REAL_ARRIVE_DATE AS REAL_ARRIVE_DATE
    from (SELECT KB.ENTRY_ID AS ENTRY_ID, KB.I_E_FLAG AS I_E_FLAG, KB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, KB.NIAN AS NIAN, KB.YUE AS YUE, KB.CREATE_DATE70 AS CREATE_DATE70, KB.CREATE_DATE80 AS
CREATE_DATE80, KB.D_DATE AS D_DATE, KB.I_E_DATE AS I_E_DATE, KB.CREATE_DATE81 AS CREATE_DATE81, KB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, KB.DECL_PORT AS DECL_PORT, MB.MANIFEST_ID AS MANIFEST_ID, MB.BILL_NO AS BILL_NO, MB.GEN_DATE AS
GEN_DATE
    from (SELECT FB.ENTRY_ID AS ENTRY_ID, FB.I_E_FLAG AS I_E_FLAG, FB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, FB.NIAN AS NIAN, FB.YUE AS YUE, FB.CREATE_DATE10 AS CREATE_DATE10, FB.CREATE_DATE70 AS
CREATE_DATE70, FB.CREATE_DATE80 AS CREATE_DATE80, FB.D_DATE AS D_DATE, FB.I_E_DATE AS I_E_DATE, FB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, FB.DECL_PORT AS DECL_PORT, HB.CREATE_DATE81 AS CREATE_DATE81
    from (SELECT AB.ENTRY_ID AS ENTRY_ID, AB.I_E_FLAG AS I_E_FLAG, AB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, AB.NIAN AS NIAN, AB.YUE AS YUE, AB.CREATE_DATE10 AS CREATE_DATE10, AB.CREATE_DATE70 AS
CREATE_DATE70, AB.D_DATE AS D_DATE, AB.I_E_DATE AS I_E_DATE, AB.TRAF_MODE_CNAME AS TRAF_MODE_CNAME, AB.DECL_PORT AS
DECL_PORT, CB.CREATE_DATE80 AS CREATE_DATE80
    from (SELECT V.ENTRY_ID AS ENTRY_ID, V.I_E_FLAG AS I_E_FLAG, V.DECL_PORT_CNAME AS DECL_PORT_CNAME, V.NIAN
AS NIAN, V.YUE AS YUE, V.CREATE_DATE10 AS CREATE_DATE10, V.D_DATE AS D_DATE, V.I_E_DATE AS I_E_DATE, V.TRAF_MODE_CNAME
AS TRAF_MODE_CNAME, V.DECL_PORT AS DECL_PORT, X.CREATE_DATE70 AS CREATE_DATE70
    from (SELECT O.ENTRY_ID AS ENTRY_ID, O.I_E_FLAG AS I_E_FLAG, O.DECL_PORT_CNAME AS DECL_PORT_CNAME, O.NIAN
AS NIAN, O.YUE AS YUE, O.D_DATE AS D_DATE, O.I_E_DATE AS I_E_DATE, O.TRAF_MODE_CNAME AS TRAF_MODE_CNAME, O.DECL_PORT
AS DECL_PORT, S.CREATE_DATE10 AS CREATE_DATE10
    from (SELECT D.ENTRY_ID AS ENTRY_ID, D.I_E_FLAG AS I_E_FLAG, D.DECL_PORT_CNAME AS
DECL_PORT_CNAME, (substr((D.END_DATE), 1, 4)) AS NIAN, (substr((D.END_DATE), 1, 7)) AS YUE, D.DECL_PORT AS
DECL_PORT, D.TRAF_MODE_CNAME AS TRAF_MODE_CNAME, D.TRAF_MODE AS TRAF_MODE, max(D.D_DATE) AS D_DATE, max(D.I_E_DATE)
AS I_E_DATE
    from (SELECT A.ENTRY_ID AS ENTRY_ID, A.I_E_FLAG AS I_E_FLAG, A.I_E_DATE AS I_E_DATE, A.DECL_PORT_CNAME AS
DECL_PORT_CNAME, A.DECL_PORT AS DECL_PORT, A.I_E_PORT AS I_E_PORT, A.I_E_PORT_CNAME AS I_E_PORT_CNAME, A.END_DATE AS
END_DATE, A.ORIGIN_COUNTRY AS ORIGIN_COUNTRY, A.CODE_TS AS CODE_TS, A.EDI_REMARK AS EDI_REMARK, A.D_DATE AS
D_DATE, A.TRAF_MODE AS TRAF_MODE, A.TRAF_MODE_CNAME AS TRAF_MODE_CNAME
    from MDB_ENTRY_LIST A
    ) D
    where
    ( ( D.ORIGIN_COUNTRY <> '142' ) and ( (case when substr((D.EDI_REMARK), 7, 1) = 'C' and
substr((D.EDI_REMARK), 15, 1) = '1'
    then '1'
    else '0'
    end) = '0' ) and ( (substr((D.CODE_TS), 1, 4)) not in ('2701', '2702') ) )
    group by
D.ENTRY_ID, D.I_E_FLAG, D.DECL_PORT_CNAME, (substr((D.END_DATE), 1, 4)), (substr((D.END_DATE), 1, 7)), D.DECL_PORT, D. TR
AF_MODE_CNAME, D.TRAF_MODE ) O
    left outer join MDB_SPECIAL_ENTRY Q on
    O.ENTRY_ID = Q.ENTRY_ID left outer join (SELECT N.ENTRY_ID AS ENTRY_ID, max(N.CREATE_DATE) AS
CREATE_DATE10
    from MDB_ENTRY_WORKFLOW N
    where
    (N.STEP_ID like '10%')
    group by N.ENTRY_ID ) S on
    O.ENTRY_ID = S.ENTRY_ID
    where
    (Q.ENTRY_ID is null) V
    left outer join (SELECT U.ENTRY_ID AS ENTRY_ID, max(U.CREATE_DATE) AS CREATE_DATE70
    from MDB_ENTRY_WORKFLOW U
    where
    (U.STEP_ID like '70%')
    group by U.ENTRY_ID ) X on
    V.ENTRY_ID = X.ENTRY_ID) AB
    left outer join (SELECT Z.ENTRY_ID AS ENTRY_ID, max(Z.CREATE_DATE) AS CREATE_DATE80
    from MDB_ENTRY_WORKFLOW Z

```

```

where
    (Z.STEP_ID like '800%')
group by Z.ENTRY_ID ) CB on
    AB.ENTRY_ID = CB.ENTRY_ID) FB
    left outer join (SELECT EB.ENTRY_ID AS ENTRY_ID,max(EB.CREATE_DATE) AS CREATE_DATE81
from MDB_ENTRY_WORKFLOW EB
where
    (EB.STEP_ID like '81%')
group by EB.ENTRY_ID ) HB on
    FB.ENTRY_ID = HB.ENTRY_ID) KB
    left outer join (SELECT JB.MANIFEST_ID AS MANIFEST_ID,JB.BILL_NO AS BILL_NO,JB.FORM_ID AS
FORM_ID,max(JB.GEN_DATE) AS GEN_DATE
from mdb_manifest_chk_new JB
group by JB.MANIFEST_ID,JB.BILL_NO,JB.FORM_ID ) MB on
    KB.ENTRY_ID = MB.FORM_ID) PB
    left outer join (SELECT OB.MANIFEST_ID AS MANIFEST_ID,OB.BILL_NO AS BILL_NO,max(OB.REAL_ARRIVE_DATE)
AS REAL_ARRIVE_DATE
from MDB_MANIFEST_LIST_NEW OB
group by OB.MANIFEST_ID,OB.BILL_NO ) RB on
    PB.MANIFEST_ID = RB.MANIFEST_ID and PB.BILL_NO = RB.BILL_NO) XB
    left outer join (SELECT TB.MANIFEST_ID AS MANIFEST_ID,TB.BILL_NO AS BILL_NO,max(TB.ARRIVAL_DATE) AS
ARRIVE_TIME
from mdb_manifest_monitor_new TB
group by TB.MANIFEST_ID,TB.BILL_NO ) ZB on
    XB.MANIFEST_ID = ZB.MANIFEST_ID and XB.BILL_NO = ZB.BILL_NO) BC
where
    ( ( (case when BC.I_E_FLAG like '%I%' and BC.CREATE_DATE81 is null and BC.CREATE_DATE80 is null
then '1'
when BC.I_E_FLAG like '%E%' and BC.CREATE_DATE70 is null and BC.CREATE_DATE81 is null
then '2'
else '0'
end) = '0' ) ) CC
    ) DC
    ) GC
where
    ( ( GC.DECL_PORT in
('0720','0719','0718','0717','0716','0715','0714','0713','0712','0711','0710','0709','0708','0707','0706','070
5','0704','0703','0702','0701','0700')) )
group by GC.YUE,GC.I_E_FLAG ) NC
    left outer join (SELECT HC.YUE AS YUE,HC.I_E_FLAG AS I_E_FLAG,avg(HC.KOUAN) AS KOUAN,avg((case when
(HC.HAIGUAN) <'0'
then '0'
else (HC.HAIGUAN)
end)) AS HAIGUAN
from (SELECT (case when datediff((DC.JIESHUSHIJIAN),(DC.STARTE),'ss')<'0'
then '0'
else datediff((DC.JIESHUSHIJIAN),(DC.STARTE),'ss')/3600
end) AS KOUAN,(case when datediff((DC.STARTE),(DC.D_DATE),'ss')>'0'
then datediff((DC.JIESHUSHIJIAN),(DC.STARTE),'ss')/3600
else datediff((DC.JIESHUSHIJIAN),(DC.D_DATE),'ss')/3600
end) AS HAIGUAN,DC.DECL_PORT_CNAME AS DECL_PORT_CNAME,DC.NIAN AS NIAN,DC.YUE AS YUE,DC.I_E_FLAG AS
I_E_FLAG,DC.TRAF_MODE_CNAME AS TRAF_MODE_CNAME,DC.DECL_PORT AS DECL_PORT
from (SELECT CC.DECL_PORT_CNAME AS DECL_PORT_CNAME,CC.NIAN AS NIAN,CC.YUE AS YUE,CC.I_E_FLAG AS
I_E_FLAG,CC.ENTRY_ID AS ENTRY_ID,(case when substr((CC.STARTE),12,8) like '%00:00:00%'
then dateadd((CC.STARTE),12,'hh')
else (CC.STARTE)
end) AS STARTE,CC.JIESHUSHIJIAN AS JIESHUSHIJIAN,CC.TRAF_MODE_CNAME AS TRAF_MODE_CNAME,CC.DECL_PORT AS
DECL_PORT,CC.D_DATE AS D_DATE

```

```

from (SELECT (Case when (BC.I_E_FLAG) like '%I%' and (BC.CREATE_DATE81) is null
Then (BC.CREATE_DATE80)
when (BC.I_E_FLAG) like '%I%' and (BC.CREATE_DATE81) is not null
Then (BC.CREATE_DATE81)
when (BC.I_E_FLAG) like '%E%' and (BC.CREATE_DATE81) is null
Then (BC.CREATE_DATE70)
when (BC.I_E_FLAG) like '%E%' and (BC.CREATE_DATE81) is not null
Then (BC.CREATE_DATE81)
else '0'
end) AS JIESHUSHIJIAN, (case when (BC.I_E_FLAG) like '%I%' and (BC.CREATE_DATE81) is null and
(BC.CREATE_DATE80) is null
then '1'
when (BC.I_E_FLAG) like '%E%' and (BC.CREATE_DATE70) is null and (BC.CREATE_DATE81) is null
then '2'
else '0'
end) AS PANDUAN, BC.DECL_PORT_CNAME AS DECL_PORT_CNAME, BC.NIAN AS NIAN, BC.YUE AS YUE, BC.I_E_FLAG AS
I_E_FLAG, BC.ENTRY_ID AS ENTRY_ID, (Case when (BC.I_E_FLAG) like 'E'
and (BC.ARRIVE_TIME) is not null
then (BC.ARRIVE_TIME)
when (BC.I_E_FLAG) like 'E'
and (BC.ARRIVE_TIME) is null
then (BC.D_DATE)
when (BC.REAL_ARRIVE_DATE) is not null
and (BC.I_E_FLAG) like 'I'
then (BC.REAL_ARRIVE_DATE)
when (BC.I_E_FLAG) like 'I'
and (BC.GEN_DATE) is not null
then (BC.GEN_DATE)
when (BC.REAL_ARRIVE_DATE) is null
and (BC.GEN_DATE) is null
and (BC.I_E_FLAG) like 'I'
then (BC.I_E_DATE)
else '0'
end) AS STARTE, BC.D_DATE AS D_DATE, BC.DECL_PORT AS DECL_PORT, BC.TRAF_MODE_CNAME AS TRAF_MODE_CNAME
from (SELECT distinct XB.ENTRY_ID AS ENTRY_ID, XB.I_E_FLAG AS I_E_FLAG, XB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, XB.NIAN AS NIAN, XB.YUE AS YUE, XB.CREATE_DATE70 AS CREATE_DATE70, XB.CREATE_DATE80 AS
CREATE_DATE80, XB.D_DATE AS D_DATE, XB.I_E_DATE AS I_E_DATE, XB.CREATE_DATE81 AS CREATE_DATE81, XB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, XB.DECL_PORT AS DECL_PORT, XB.REAL_ARRIVE_DATE AS REAL_ARRIVE_DATE, XB.GEN_DATE AS
GEN_DATE, ZB.ARRIVE_TIME AS ARRIVE_TIME
from (SELECT PB.ENTRY_ID AS ENTRY_ID, PB.I_E_FLAG AS I_E_FLAG, PB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, PB.NIAN AS NIAN, PB.YUE AS YUE, PB.CREATE_DATE70 AS CREATE_DATE70, PB.CREATE_DATE80 AS
CREATE_DATE80, PB.D_DATE AS D_DATE, PB.I_E_DATE AS I_E_DATE, PB.CREATE_DATE81 AS CREATE_DATE81, PB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, PB.DECL_PORT AS DECL_PORT, PB.MANIFEST_ID AS MANIFEST_ID, PB.BILL_NO AS BILL_NO, PB.GEN_DATE AS
GEN_DATE, RB.REAL_ARRIVE_DATE AS REAL_ARRIVE_DATE
from (SELECT KB.ENTRY_ID AS ENTRY_ID, KB.I_E_FLAG AS I_E_FLAG, KB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, KB.NIAN AS NIAN, KB.YUE AS YUE, KB.CREATE_DATE70 AS CREATE_DATE70, KB.CREATE_DATE80 AS
CREATE_DATE80, KB.D_DATE AS D_DATE, KB.I_E_DATE AS I_E_DATE, KB.CREATE_DATE81 AS CREATE_DATE81, KB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, KB.DECL_PORT AS DECL_PORT, MB.MANIFEST_ID AS MANIFEST_ID, MB.BILL_NO AS BILL_NO, MB.GEN_DATE AS
GEN_DATE
from (SELECT FB.ENTRY_ID AS ENTRY_ID, FB.I_E_FLAG AS I_E_FLAG, FB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, FB.NIAN AS NIAN, FB.YUE AS YUE, FB.CREATE_DATE10 AS CREATE_DATE10, FB.CREATE_DATE70 AS
CREATE_DATE70, FB.CREATE_DATE80 AS CREATE_DATE80, FB.D_DATE AS D_DATE, FB.I_E_DATE AS I_E_DATE, FB.TRAF_MODE_CNAME AS
TRAF_MODE_CNAME, FB.DECL_PORT AS DECL_PORT, HB.CREATE_DATE81 AS CREATE_DATE81
from (SELECT AB.ENTRY_ID AS ENTRY_ID, AB.I_E_FLAG AS I_E_FLAG, AB.DECL_PORT_CNAME AS
DECL_PORT_CNAME, AB.NIAN AS NIAN, AB.YUE AS YUE, AB.CREATE_DATE10 AS CREATE_DATE10, AB.CREATE_DATE70 AS
CREATE_DATE70, AB.D_DATE AS D_DATE, AB.I_E_DATE AS I_E_DATE, AB.TRAF_MODE_CNAME AS TRAF_MODE_CNAME, AB.DECL_PORT AS
DECL_PORT, CB.CREATE_DATE80 AS CREATE_DATE80
from (SELECT V.ENTRY_ID AS ENTRY_ID, V.I_E_FLAG AS I_E_FLAG, V.DECL_PORT_CNAME AS DECL_PORT_CNAME, V.NIAN

```



```

AS NIAN, V. YUE AS YUE, V. CREATE_DATE10 AS CREATE_DATE10, V. D_DATE AS D_DATE, V. I_E_DATE AS I_E_DATE, V. TRAF_MODE_CNAME
AS TRAF_MODE_CNAME, V. DECL_PORT AS DECL_PORT, X. CREATE_DATE70 AS CREATE_DATE70
    from (SELECT O. ENTRY_ID AS ENTRY_ID, O. I_E_FLAG AS I_E_FLAG, O. DECL_PORT_CNAME AS DECL_PORT_CNAME, O. NIAN
AS NIAN, O. YUE AS YUE, O. D_DATE AS D_DATE, O. I_E_DATE AS I_E_DATE, O. TRAF_MODE_CNAME AS TRAF_MODE_CNAME, O. DECL_PORT
AS DECL_PORT, S. CREATE_DATE10 AS CREATE_DATE10
    from (SELECT D. ENTRY_ID AS ENTRY_ID, D. I_E_FLAG AS I_E_FLAG, D. DECL_PORT_CNAME AS
DECL_PORT_CNAME, (substr((D. END_DATE), 1, 4)) AS NIAN, (substr((D. END_DATE), 1, 7)) AS YUE, D. DECL_PORT AS
DECL_PORT, D. TRAF_MODE_CNAME AS TRAF_MODE_CNAME, D. TRAF_MODE AS TRAF_MODE, max(D. D_DATE) AS D_DATE, max(D. I_E_DATE)
AS I_E_DATE
    from (SELECT A. ENTRY_ID AS ENTRY_ID, A. I_E_FLAG AS I_E_FLAG, A. I_E_DATE AS I_E_DATE, A. DECL_PORT_CNAME AS
DECL_PORT_CNAME, A. DECL_PORT AS DECL_PORT, A. I_E_PORT AS I_E_PORT, A. I_E_PORT_CNAME AS I_E_PORT_CNAME, A. END_DATE AS
END_DATE, A. ORIGIN_COUNTRY AS ORIGIN_COUNTRY, A. CODE_TS AS CODE_TS, A. EDI_REMARK AS EDI_REMARK, A. D_DATE AS
D_DATE, A. TRAF_MODE AS TRAF_MODE, A. TRAF_MODE_CNAME AS TRAF_MODE_CNAME
    from MDB_ENTRY_LIST A
    ) D
    where
        ( ( D. ORIGIN_COUNTRY <> '142') and ( (case when substr((D. EDI_REMARK), 7, 1) = 'C' and
substr((D. EDI_REMARK), 15, 1) = '1'
        then '1'
        else '0'
        end) = '0') and ( (substr((D. CODE_TS), 1, 4)) not in ('2701', '2702')) )
    group by
D. ENTRY_ID, D. I_E_FLAG, D. DECL_PORT_CNAME, (substr((D. END_DATE), 1, 4)), (substr((D. END_DATE), 1, 7)), D. DECL_PORT, D. TR
AF_MODE_CNAME, D. TRAF_MODE ) O
    left outer join MDB_SPECIAL_ENTRY Q on
    O. ENTRY_ID = Q. ENTRY_ID left outer join (SELECT N. ENTRY_ID AS ENTRY_ID, max(N. CREATE_DATE) AS
CREATE_DATE10
    from MDB_ENTRY_WORKFLOW N
    where
        (N. STEP_ID like '10%')
    group by N. ENTRY_ID ) S on
    O. ENTRY_ID = S. ENTRY_ID
    where
        (Q. ENTRY_ID is null) V
    left outer join (SELECT U. ENTRY_ID AS ENTRY_ID, max(U. CREATE_DATE) AS CREATE_DATE70
    from MDB_ENTRY_WORKFLOW U
    where
        (U. STEP_ID like '70%')
    group by U. ENTRY_ID ) X on
    V. ENTRY_ID = X. ENTRY_ID) AB
    left outer join (SELECT Z. ENTRY_ID AS ENTRY_ID, max(Z. CREATE_DATE) AS CREATE_DATE80
    from MDB_ENTRY_WORKFLOW Z
    where
        (Z. STEP_ID like '800%')
    group by Z. ENTRY_ID ) CB on
    AB. ENTRY_ID = CB. ENTRY_ID) FB
    left outer join (SELECT EB. ENTRY_ID AS ENTRY_ID, max(EB. CREATE_DATE) AS CREATE_DATE81
    from MDB_ENTRY_WORKFLOW EB
    where
        (EB. STEP_ID like '81%')
    group by EB. ENTRY_ID ) HB on
    FB. ENTRY_ID = HB. ENTRY_ID) KB
    left outer join (SELECT JB. MANIFEST_ID AS MANIFEST_ID, JB. BILL_NO AS BILL_NO, JB. FORM_ID AS
FORM_ID, max(JB. GEN_DATE) AS GEN_DATE
    from mdb_manifest_chk_new JB
    group by JB. MANIFEST_ID, JB. BILL_NO, JB. FORM_ID ) MB on
    KB. ENTRY_ID = MB. FORM_ID) PB
    left outer join (SELECT OB. MANIFEST_ID AS MANIFEST_ID, OB. BILL_NO AS BILL_NO, max(OB. REAL_ARRIVE_DATE)

```

```

AS REAL_ARRIVE_DATE
    from MDB_MANIFEST_LIST_NEW OB
    group by OB.MANIFEST_ID,OB.BILL_NO ) RB on
    PB.MANIFEST_ID = RB.MANIFEST_ID and PB.BILL_NO = RB.BILL_NO) XB
    left outer join (SELECT TB.MANIFEST_ID AS MANIFEST_ID, TB.BILL_NO AS BILL_NO, max(TB.ARRIVAL_DATE) AS
ARRIVE_TIME
    from mdb_manifest_monitor_new TB

    group by TB.MANIFEST_ID, TB.BILL_NO ) ZB on
    XB.MANIFEST_ID = ZB.MANIFEST_ID and XB.BILL_NO = ZB.BILL_NO) BC
    where
    ( ( (case when BC.I_E_FLAG like '%I%' and BC.CREATE_DATE81 is null and BC.CREATE_DATE80 is null
then '1'
when BC.I_E_FLAG like '%E%' and BC.CREATE_DATE70 is null and BC.CREATE_DATE81 is null
then '2'
else '0'
end) = '0') )) CC
    ) DC
    ) HC
    group by HC.YUE, HC.I_E_FLAG ) OC on
    NC.YUE = OC.YUE and NC.I_E_FLAG = OC.I_E_FLAG
order by YUE asc, I_E_FLAG asc

```

(两种打开方式，二选一即可)

★ 电脑打开：评价此文章，请打开该网站：

<https://www.wjx.cn/jq/92217593.aspx>

★ 手机打开：评价此文章，请扫描该二维码：

