

HP Service Health Reporter

소프트웨어 버전: 9.20

Performance Insight 데이터 마이그레이션 설명서



목차

소개.....	2
마이그레이션하는 이유.....	2
마이그레이션의 목표.....	2
마이그레이션의 장점.....	2
마이그레이션의 범위.....	3
좋은 데이터 마이그레이션 후보의 조건.....	4
용어 정의.....	4
데이터 마이그레이션 - 아키텍처.....	5
가정.....	5
Performance Insight 에서의 데이터 마이그레이션 - OM 배포 시나리오.....	6
Performance Insight 에서의 데이터 마이그레이션 - BSM 배포 시나리오.....	7
데이터 마이그레이션 방법.....	7
알려진 문제 및 제한.....	8
데이터 마이그레이션 단계.....	10
사전 조건.....	10
PI 마이그레이션 콘텐츠 - 패키징.....	10
PI로부터 데이터 마이그레이션을 수행하기 위한 단계.....	10
일반적인 사용 사례/시나리오.....	11
자주 묻는 질문.....	12
참조.....	14
PI의 System Resource 리포트 팩 및 SHR의 System Performance 콘텐츠 간 데이터 매핑.....	14

소개

HP Performance Insight 는 성능 관리 및 보고 응용 프로그램입니다. 기존 Performance Insight 고객을 BSM 포트폴리오에서 다른 보고 솔루션으로 마이그레이션하는 과정을 단계적 접근으로 계획합니다.

마이그레이션하는 이유

HP PI가 언젠가는 사용되지 않을 수 있습니다. 마이그레이션 프로그램은 현재 Performance Insight 고객이 유효한 지원 계약에 따라 다른 BSM 보고 도구로 전환하도록 도와 줍니다.

사용 사례를 바탕으로 하나 이상의 제품으로 마이그레이션하는 HP PI 고객

- Service Health Reporter
- NNMi + Performance iSPI

마이그레이션의 목표

- HP 가 BSM 보고 도구를 통합하면서 HP Performance Insight 고객에게 거의 동일한 기능 제공
- HP PI 고객은 Service Health Reporter 로 마이그레이션하여 x-domain 분석 및 보고와 같은 향상된 기능을 활용할 수 있습니다.

이 문서에서는 System Performance 용 SHR 콘텐츠를 활용하기 위해 HP PI System Resource 보고서 팩의 데이터를 SHR 로 마이그레이션하는 과정을 자세하게 설명합니다.

마이그레이션의 장점

다음은 Service Health Reporter(1 단계)로 마이그레이션하여 얻을 수 있는 장점 중 일부입니다.

- 차세대 교차 BSM 보고 솔루션으로 업그레이드
 - SHR 에서 모델 중심 보고 수행 => 응용 프로그램 성능을 기반 인프라와 연결
 - 환경에 대한 전체적인 시야 제공
 - 토폴로지 업데이트가 자동으로 추적됨
 - 서비스의 응용 프로그램, 인프라, 데이터베이스 및 네트워크의 성능을 한 화면에서 확인
- 비즈니스 인텔리전스를 데이터 센터 관리에 접목
 - 보고 프레임워크로 SAP Business Objects Enterprise 활용
 - 손쉬운 사용자 지정
- 추가 기능
 - 보고에서 가동 중지 시간 고려/사용자 지정 그룹/사용자 지정 가능 교대
 - 가상화 용량 계획과 같은 향후 분석을 고려한 플랫폼
- 확장성
 - 단일 SHR 인스턴스로 5000 개의 시스템 노드에서 직접 수집 가능

이 문서의 목적은 다음 내용을 설명하는 것입니다.

- Performance Insight 에서 SHR 로 데이터를 마이그레이션하는 데 채택된 방법
- System Performance 콘텐츠를 위한 Performance Insight 데이터 마이그레이션에서 수행할 단계

- 가정
- 알려진 제한
- 확장성 및 성능 측면

마이그레이션의 범위

Performance Insight(1 단계)에서의 데이터 마이그레이션을 위한 콘텐츠의 범위는 System Resource 보고서 팩 및 다음 하위 패키지로 제한됩니다.

- SystemResourceCPU
- SystemResource_Disk
- SystemResource_NetInterface

SHR 에서 제공하는 기능	SHR 에서 제공하지 않는 기능
<p>지속적인 서비스 향상 및 내역 보고 사용 사례 지원</p> <ul style="list-style-type: none"> - 통합 보고 - BSM 및 비 BSM 시나리오 <p>중간 데이터 저장소에서 수집</p> <ul style="list-style-type: none"> - 성능 에이전트 데이터 수집 제외 	<p>SHR 의 보고서는 지난 내역 기반이고 개략적이며 실시간에 가까운 보고서가 아니며 사용 사례의 문제를 해결하는 데 사용하지 않습니다.</p> <ul style="list-style-type: none"> - 진단 또는 문제 해결을 위한 기본 제공 보고서 없음
<p>PI 로부터 1 회용 데이터 마이그레이션</p> <ul style="list-style-type: none"> - 데이터를 마이그레이션한 후 SHR 에서 원본(System Performance 콘텐츠용 에이전트 및 토폴로지용 HP OM/RtSM)에서의 실시간 수집을 구성해야 함 	<p>1 회용 데이터 마이그레이션에 포함되지 않는 항목:</p> <ul style="list-style-type: none"> - HP OM SPI(가상 SPI 포함) - InterfaceReporting/DeviceResource, 트래픽, MPLS, QoS, IPTelephony, 서비스 보증 및 다른 네트워크 성능 SPI 메트릭
<p>집계된 데이터(시간별 및 일별 단위)가 빈도 데이터가 아닌 PI 로부터 마이그레이션됨 구성이 마이그레이션되지 않음</p>	<p>지속적 데이터 수집의 원본으로 PI 를 사용하지 않음</p>
<p>CDE(콘텐츠 개발 환경)를 통한 콘텐츠 사용자 지정 및 확장 지원</p> <ul style="list-style-type: none"> - PI 의 개별 보고서는 그 자체로는 기본 제공 SHR 콘텐츠로 마이그레이션되지 않지만 추가 사용 사례가 있는 경우 사용자 지정 보고서를 만들어 처리할 수 있습니다. - 기본 제공 콘텐츠도 사용자 지정할 수 있음 	<p>대기 시간이 낮은 데이터를 수집하지 않음</p> <ul style="list-style-type: none"> - 보고서에 실시간 데이터가 없음
<p>보고서용 사용자 지정 그룹 생성이 지원됨</p>	<p>Linux/Unix 버전에서 지원되지 않거나 Oracle/Sybase ASE- Sybase IQ 가 지원되는 유일한 데이터베이스임</p>

좋은 데이터 마이그레이션 후보의 조건

다음은 데이터 마이그레이션의 좋은 후보를 식별하는 방법입니다.

- PI의 기본 용도는 다음과 같습니다.
 - 시스템(OM 메시지/데이터베이스 SPI/가상화 및 사이트 범위)
 - 네트워크 데이터(Perf/iSPI에 의해 처리)
- 환경의 규모
 - SHR용 5000개 이하의 노드
- SAP Business Objects 기반 보고서를 사용자 지정하기 위한 전문 기술 보유
- 사용자 지정 보고서 작성을 위한 기본 분야를 올바르게 이해
- 현재 OM 또는 BSM 사용자
- 보고 솔루션에서 PI의 기능 이상을 제공하는 새로운 기능을 모색 중
- 교대, 가동 중지 시간, 사용자 지정 그룹 등.

용어 정의

용어	정의
CP	컨텐츠 팩
RP	보고서 팩
DWH	데이터 웨어하우스
HP PI	HP Performance Insight
BSM	Business Service Management
RISM	실시간 서비스 모델
OM	Operations Manager
CDE	컨텐츠 개발 환경
SPI	Smart Plug-In
SHR	Service Health Reporter
ETL	압축 해제, 변환 및 로드
ABC	감사, 균형 및 제어

데이터 마이그레이션 - 아키텍처

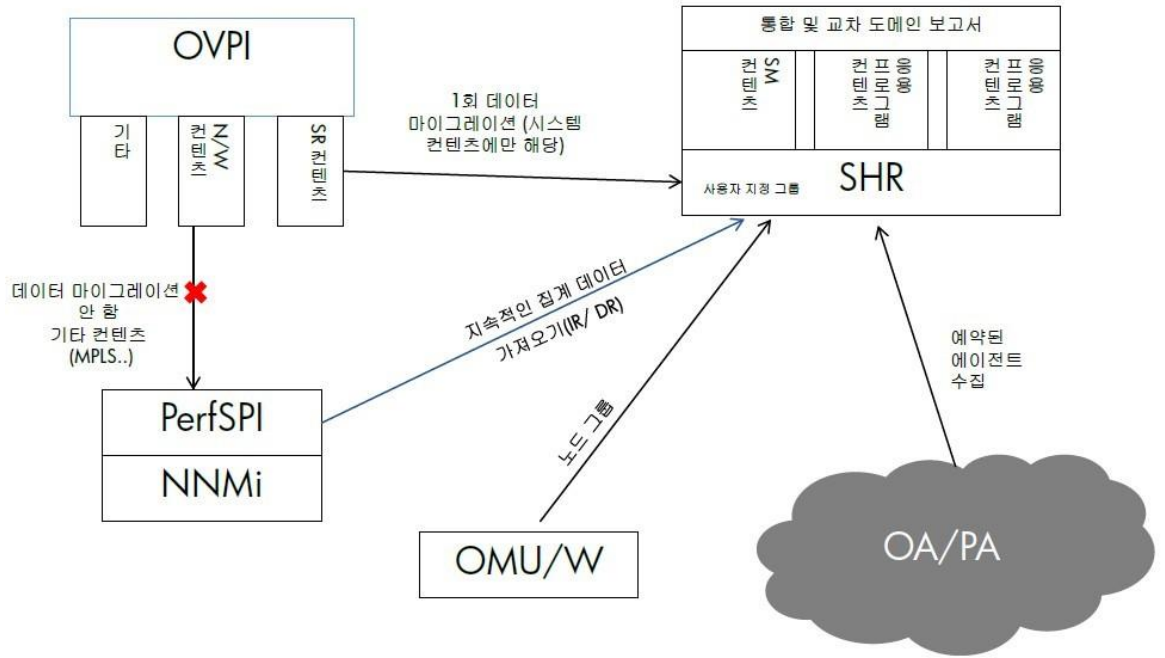
1 단계에서는 PI 고객이 하나 이상의 BSM 제품(사용 사례를 바탕으로)으로 마이그레이션됩니다. 이 섹션에서는 데이터 마이그레이션에 활용되는 아키텍처 접근, 가정, 시나리오, 그리고 알려진 제한에 대해 자세히 설명합니다.

가정

- PI는 지속적 데이터 수집을 위한 데이터 원본이 아님(1 회용 데이터 마이그레이션 방법)
- PI 고객은 현재 OM 또는 BSM 사용자임
- 보고 솔루션의 규모는 SHR v9.20에서 지원하는 규모보다 작아야 함
- SHR이 별도의 서버에 설치됨(PI 서버에 공존하지 않음)
- HP PI(5.3 및 5.4x) 버전만 지원됨 - 이전 버전의 PI(및 지원되지 않는 구성의 HP PI)는 마이그레이션을 위한 데이터 원본으로 지원되지 않음
- SHR의 토폴로지 원본 구성은 1 회용 구성이며 마이그레이션 전/후 동일하게 유지되어야 합니다.
- 데이터 마이그레이션 패키지는 PI로부터 디멘션 및 팩트 데이터를 가져옵니다. 사용자가 다음을 위해 SHR ETL 패키지(예: SysPerf_ETL_PerformanceAgent)를 설치해야 합니다.
 - 토폴로지 데이터 가져오기(해당하는 토폴로지 원본으로부터 - 예: OM의 경우 노드 그룹, RtSM의 경우 비즈니스 서비스 기반 토폴로지)
 - 에이전트로부터 지속적 실시간 수집(1 회용 데이터 마이그레이션이므로)

다음 섹션에서는 마이그레이션 방법(1 단계)을 설명합니다.

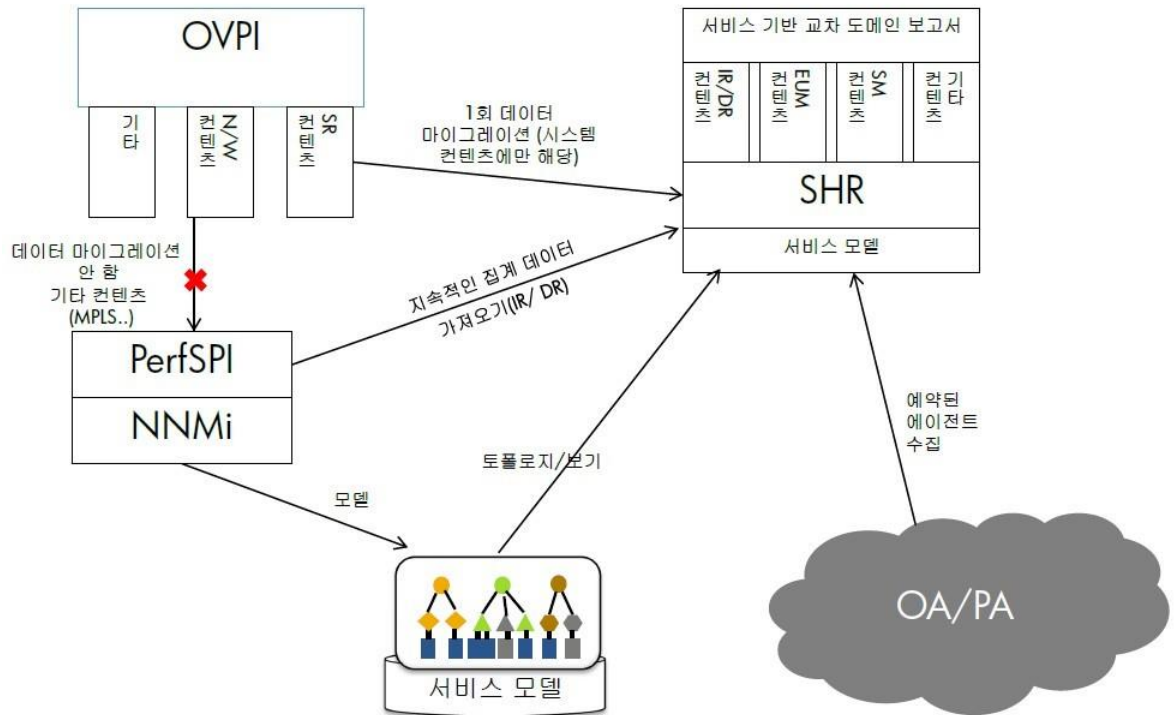
Performance Insight에서의 데이터 마이그레이션 – OM 배포 시나리오



OM 배포 시나리오에서 데이터 마이그레이션 패키지는 SHR에서 토폴로지 수집을 위해 구성된 OM(Unix/Linux/Windows) 원본으로부터 토폴로지 정보를 얻습니다. 데이터는 PI System Resource 콘텐츠의 시간별 및 일별 테이블로부터 토폴로지 원본으로 구성된 OM이 모니터링하는 노드를 기준으로 SHR로 마이그레이션됩니다.

데이터 마이그레이션이 완료되면 에이전트에서 SHR로 토폴로지/노드 그룹 정보와 실시간 빈도 데이터를 가져오기 위해 SysPerf_ETL_PerformanceAgent를 설치해야 합니다.

Performance Insight에서의 데이터 마이그레이션 – BSM 배포 시나리오



BSM 배포 시나리오에서 데이터 마이그레이션 패키지는 SHR에서 토폴로지 수집을 위해 구성된 RrSM 인스턴스로부터 토폴로지 정보를 얻습니다. 데이터는 PI System Resource 컨텐츠의 시간별 및 일별 테이블로부터 RrSM의 SM_PA 뷰에 있는 노드를 기준으로 SHR로 마이그레이션됩니다.

데이터 마이그레이션이 완료되면 서비스 기반 토폴로지 정보(RrSM에 배포된 SM_PA와 같은 SHR 뷰로부터)와 실시간 빈도 데이터를 에이전트에서 SHR로 가져오기 위해 SysPerf_ETL_PerformanceAgent를 설치해야 합니다.

데이터 마이그레이션 방법

데이터 마이그레이션을 수행할 때 ETL 및 SHR의 오케스트레이션(ABC(감사, 균형 및 제어)) 프레임워크를 사용합니다. 다음은 각 ABC 스트림에서 발생하는 데이터 흐름의 요약입니다.

- 토폴로지 데이터 수집
- 조정 레지스트리 구축
- db 수집자를 사용하여 PI(Oracle 및 Sybase ASE)로부터 데이터(디멘션/시간별 및 일별 팩트 데이터) 추출(연관된 PI 테이블에 대해서는 이 문서의 참조 섹션에서 자세히 설명됩니다)
- PI의 데이터를 토폴로지 원본과 비교하여 조정
- 로더를 사용하여 SHR 집계 테이블로 데이터 로드

토폴로지 데이터 수집

데이터는 SHR 에 구성된 토폴로지 원본(RISM 및 OM)으로부터 수집됩니다. 토폴로지 수집은 PI 의 팩트 데이터를 조정하는 데 활용되는 조정 레지스트리를 구축하기 위한 것입니다.

참고: 세부적인 디멘션 및 토폴로지 수집은 마이그레이션 패키지에서 처리되지 않습니다. 따라서 실시간 데이터 흐름이 시작되면 마이그레이션 후 작업이 필요합니다.

PI 로부터 데이터 추출

데이터 추출은 HP PI 에서 사용하는 Oracle/Sybase ASE 데이터베이스에서 지원됩니다. 마이그레이션 패키지는 이를 위해서 SHR 데이터베이스 수집 매커니즘을 활용합니다.

마이그레이션 콘텐츠 팩은 PI 데이터베이스에서만 시간별 테이블에서 시작하는 데이터를 추출할 필요가 있습니다. SHR 에는 System Performance 용 시간별 및 일별 테이블만 있으므로 데이터 마이그레이션은 PI(및 해당 하위 패키지)의 시스템 리소스 보고서 팩의 해당 부분에서 수행됩니다. PI 의 연별 및 월별 집계 데이터 및 빈도 데이터는 SHR 로 마이그레이션되지 않습니다.

데이터 마이그레이션에 이어 SHR 에서 적절한 초기 내역으로 빈도 데이터 수집을 시작하여 HP Performance Agent 로부터 원시 메트릭을 가져올 수 있습니다.

HP PI 에서 데이터 처리

SHR DB Collector 는 수집한 데이터를 '%PMDB_HOME%\collect' 폴더로 덤프하며, 이 데이터는 추가 처리를 위해 '수집' 단계를 사용하여 스테이지 폴더로 복사해야 합니다. PI 로부터 수집한 데이터에는 토폴로지 정보가 없으며 CI_UID 가 지정되지도 않습니다. 따라서 수집한 데이터를 토폴로지 원본에서 수집한 데이터를 바탕으로 조정해야 합니다.

HP PI 데이터를 SHR 로 로드

SHR 의 로더 모듈은 시간별 및 일별 집계 팩트 데이터와 PI 에서 수집된 디멘션 및 위치/고객 브릿지 정보를 로드하는 데 사용됩니다.

알려진 문제 및 제한

- 다음 항목은 마이그레이션되지 않습니다.
 - PI 에서 가져오는 데이터(메트릭 및 디멘션)의 범위는 SHR 의 데이터 모델로 매핑할 수 있는 항목만 포함됩니다(예를 들어 파일 시스템 데이터/예측 메트릭은 마이그레이션되지 않음). PI 및 SHR 스키마 간의 자세한 매핑 목록은 이 문서의 참조 섹션에 정리되어 있습니다.
 - 구성 세부 정보(수집 구성, PI 의 기본 하드 코드 교대, 예약과 같은 보고서 구성 등)는 SHR 로 마이그레이션되지 않습니다.
 - 기획된 가동 중지 시간 기능은 PI 에서 지원되지 않으므로 마이그레이션된 데이터 세트에서 사용할 수 없습니다.
 - SHR 9.20 에는 고객별 기본 제공 보고서가 없습니다. 그러나 고객 디멘션 데이터 및 노드 디멘션에 대한 해당하는 연결은 SHR 로 마이그레이션됩니다(사용자 지정 범용/보고서도 마찬가지로 생성 가능).
- PI 에서는 일관적인 시간대를 지원하지 않으므로 마이그레이션된 데이터 세트는 원본의 시간대로 제공됩니다. SHR 9.20 에서 사용자는 시간대를 GMT 또는 현지 시간대로 설정하는 옵션을 선택할 수 있습니다. 그러나 PI 및 SHR 간에 시간대가 다른 경우 데이터

마이그레이션 중에 시간대가 변환되지 않습니다. 마이그레이션된 데이터는 원본의 시간대를 따르며 새로 수집된 데이터(SysPerf_ETL_PerformanceAgent 콘텐츠 설치 후)는 SHR 에서 구성된 시간대를 따릅니다.

- PI로부터 가져온 디멘션 특성은 SHR 실시간 수집으로 가져온 특성으로 업데이트됩니다.
- PI에서 사용 가능한 모든 시간별 및 일별 데이터가 수집됩니다. 현재 마이그레이션 콘텐츠에서는 이러한 데이터의 간격 또는 필터링을 사용자 지정할 수 없습니다.
- SHR의 내역 보고서에 PI에서 구성된 보존 기간(시간별/일별) 동안만 PI 마이그레이션 데이터가 표시됩니다. 예를 들어 PI의 시간별 데이터 보존 기간이 7일인 경우 7일 기간 이전의 일별 데이터에서 드릴 다운을 수행하면 자세한 시간별 데이터를 얻을 수 없습니다.
- 데이터 마이그레이션 패키지는 구성된 OM/RISM 토폴로지 원본의 일부로 사용 가능한 PI에서만 노드의 팩트 데이터를 가져옵니다.
- PI에서 SHR로 가져온 예외 및 서비스 등급 메트릭은 PI에 구성된 임계값을 기준으로 다시 계산됩니다. SHR의 실시간 수집은 이러한 메트릭을 계산하기 위해 SHR에 구성된 임계값을 활용합니다. PI 및 SHR에 구성된 임계값은 서로 다를 수 있습니다.
- 기본 교대만 마이그레이션되는 데이터에 적용됩니다. 데이터 마이그레이션 이전에 SHR에서 교대를 구성하지 않습니다. 교대 보장은 마이그레이션되는 데이터의 정확성에 영향을 줄 수 있습니다. 마이그레이션 후 교대를 구성할 수 있으며 이 경우 이후 수집되는 데이터에 해당 교대가 적용됩니다.
- 현재 관리자 UI의 일반 데이터베이스 구성에는 알려진 문제가 있습니다. 이미 일반 db 데이터 원본으로 Perf iSPI 데이터베이스(또는 다른 외부 데이터베이스)를 구성하였고 마이그레이션을 위해 PI도 구성한 경우 배포된 수집 정책 때문에 두 원본에서 모두 데이터를 수집하려고 할 수 있습니다. 이 경우 올바른 원본에 대한 수집에는 문제가 없지만 다른 원본에 대해서는 로그 파일에 실패 오류가 기록되고 ABC 스트림 상태가 경고로 표시됩니다.

데이터 마이그레이션 단계

사전 조건

PI 에서 SHR 로 System Performance 컨텐츠의 데이터 마이그레이션을 시작하기 전에 다음 사항을 확인하십시오.

- PI Timer Service 가 중지되어야 합니다(마이그레이션 기간 동안 계속 중지 상태 유지).
- SHR Collection 및 Timer 서비스가 비활성화되어야 합니다.
- SHR 관리자 GUI 에 구성된 교대가 없어야 합니다(기본 교대만 사용 가능). SHR 에 구성된 교대가 있으면 PI 로부터 로드한 집계 데이터의 정확성에 문제가 생기므로 매우 중요합니다.

PI 마이그레이션 컨텐츠 – 패키징

PI 데이터 마이그레이션 패키지(HPSHRSmPIMgr.msi)는 미디어 및 실시간 네트워크로 모두 사용 가능합니다. 동일한 패키지로 RiSM 및 OM 배포 시나리오를 모두 지원할 수 있으며 System Management 의 도메인 컨텐츠(SysPerf_Domain) 에 종속됩니다.

PI로부터 데이터 마이그레이션을 수행하기 위한 단계

마이그레이션 패키지 설치

- SHR DVD 에 있는 PI 마이그레이션 패키지(HPSHRSmPIMgr.msi)를 설치합니다. 그러면 %PMDB_HOME%/packages 폴더에 데이터 마이그레이션 패키지(Migration_SM_PI\ Migration_SM_PI.ap)가 생성됩니다.
- 배포 관리자를 사용하여 패키지를 배포합니다. (데이터 마이그레이션을 진행하기 전에 Collection 및 Timer 서비스가 실행되고 있지 않아야 합니다.)

데이터 원본 구성

다음 단계는 데이터 수집을 위한 토폴로지 및 PI 데이터베이스 원본을 구성하는 것입니다.

- 토폴로지 수집에 필요한 토폴로지 원본 세부 정보는 SHR 관리자 UI 의 RiSM/OM 토폴로지 원본 구성 페이지를 사용하여 구성해야 합니다.
- HP PI 데이터베이스 세부 정보는 SHR 관리자 UI 에서 ‘일반 데이터베이스 구성’으로 구성해야 합니다.

(참고: 이러한 사례의 추가 구성에 대해서는 이 문서의 ‘PI 의 분산 배포’에 나오는 사용 사례 섹션을 참조하십시오.)

데이터 마이그레이션 스크립트 실행

데이터 마이그레이션을 시작하려면 명령줄에서 다음과 같이 데이터 마이그레이션 스크립트를 실행합니다.

```
trend_proc -f %PMDB_HOME%/packages/Migration_SM_PI/Migration_SM_PI.ap/migrate_pi_shr.pro
```

데이터 마이그레이션 시퀀스는 SHR 의 데이터 흐름 오케스트레이션(ABC) 프레임워크의 지원을 받으므로 마이그레이션과 관련된 단계의 상태를 관리자 UI 에서 볼 수 있습니다.

마이그레이션된 데이터 확인

PI 에서 사용 가능한 행 수와 SHR 로 가져온 행 수를 비교하는 유틸리티가 있습니다.

데이터를 확인하려면 다음 단계를 수행합니다.

PI 시스템(%DPIPE_HOME%/data)에서 systems.xml 파일을 SHR 경로 %PMDB_HOME%/data 로 복사합니다.

```
ovpiupgrade -ovpi <PI *.csv path> -shr92 <SHR *.csv path> -xmlPath <absolute path to the mapping xml> -report <absolute path to the folder for the summary report>
```

마이그레이션된되는 행 수에 차이가 있을 수 있으며, 다음이 그 원인일 수 있습니다.

- i) PI의 노드를 토폴로지 원본에서 사용할 수 없어 조정이 실패함
- ii) PI의 원래 키 값(예: 노드 이름)이 토폴로지 원본과 일치하지 않음
- iii) 도구가 사용자 지정 노드 필터링을 감안하지 않고 PI의 전체 노드를 비교 - 제한적 기능임

관련 로그 파일

%PMDB_HOME%/log 폴더의 다음 로그 파일에는 데이터 마이그레이션 패키지 설치/실행 중 기록된 세부 정보가 포함되어 있습니다.

- **packagemanager.log** - 배포 관리자를 사용하는 마이그레이션 패키지(및 해당 종속성인 SysPerf_Domain 패키지)의 배포 세부 정보
- **Topologycollector.log** - OM/RtSM 토폴로지 수집 세부 정보
- **reconcileStep.log** - 구성된 토폴로지 원본의 디멘션 데이터를 기준으로 수행된 조정 세부 정보
- **dbcollector.log** - HP PI 데이터베이스 대상의 쿼리 실행 및 초기 .csv 파일을 생성하기 전 추출 단계와 관련된 조인 쿼리 세부 정보
- **dwabclauncher.log** - ABC 스트림 실행 세부 정보
- **loader.log** - 시간별/일별 집계 데이터를 SHR 테이블로 로드하는 마지막 단계 세부 정보

마이그레이션 패키지 제거

데이터를 마이그레이션한 후에는 마이그레이션 패키지를 제거해야 합니다.

데이터 마이그레이션 패키지는 배포 관리자를 사용하여 제거할 수 있습니다.

이 밖에도 PI 마이그레이션 패키지(HPSHRSmPIMgr.msi)를 제거해야 합니다. 이것은 데이터 마이그레이션 프로세스가 완료된 후 배포 관리자에 선택 항목으로 마이그레이션 팩을 표시하지 않기 위한 것입니다.

SHR 서비스 재시작 및 System Performance ETL 패키지 설치

이제 Collection 및 Timer 서비스를 포함한 SHR 서비스를 활성화 및 시작할 수 있습니다.

System Performance 컨텐츠의 경우 필요한 토폴로지 데이터를 수집하고 Performance Agent 에서 보고용 실시간 데이터 수집을 위해 ETL 패키지(SysPerf_ETL_PerformanceAgent)를 설치합니다.

참고: PI 마이그레이션 컨텐츠 팩은 PI로부터 팩트 및 디멘션 데이터만 마이그레이션합니다. 구성된 토폴로지 원본으로부터의 토폴로지/디멘션 수집은 해당 ETL 패키지가 설치되고 토폴로지 수집 라운드가 실행되어야 수행됩니다. 따라서 먼저 Timer 및 Collection 서비스를 시작해야 보고서에서 데이터를 볼 수 있습니다.

일반적인 사용 사례/시나리오

분산 환경 - 위성 및 중앙 서버

이 경우에는 SHR 관리자 UI에서 PI 위성/중앙 서버 세부 정보를 구성한 후에 마이그레이션 스트림을 여러 번 실행해야 합니다. 즉, 위성 서버 2 개와 중앙 서버 1 개가 있는 경우 사용자는 먼저 관리자 UI에서 첫 번째 위성 서버를 구성하고 migrate_pi_shr.pro 파일을 실행해야 합니다. 스트림 실행이 완료되면 두 번째 위성 서버 세부 정보를 구성하고 스트림을 실행해야 합니다. 그리고 각 위성 및 중앙 서버에 동일한 단계를 반복합니다. (이보다 규모가 작은 배포의 경우 관리자 UI에서 먼저 중앙 및 위성 원본을 모두 구성하고 migrate_pi_shr.pro 파일을 실행해도 됩니다.)

참고: 중앙 서버 자체에서 시간별 데이터를 사용할 수 있는 경우(고객이 HP PI에서 정의한 복사 정책에 따라) 개별 위성 서버에서 데이터를 끌어올 필요가 없습니다.

자주 묻는 질문

이 섹션에서는 데이터 마이그레이션 프로세스와 관련된 몇 가지 공통적인 질문과 대답을 제공합니다.

Q1. SHR 을 설치하고 실행한 지 몇 달이 지난 후에 PI 에서 데이터를 마이그레이션할 수 있습니까? 다른 보고서 팩이 배포된 경우 필요한 마이그레이션 패키지를 모두 사용할 수 있을 때까지 마이그레이션 작업을 연기해야 합니까?

A1. SHR 을 설치하고 실행한 지 오래되었더라도 PI 로부터의 데이터 마이그레이션을 시작할 수 있습니다. 요점은 SHR 에서 데이터가 수집되지 않은 기간에 해당하는 데이터가 PI 에서 마이그레이션된다는 것입니다. 예를 들어 SHR 에서 마지막 한 주 동안의 데이터를 수집하고 집계한 경우 그 이전의 데이터만 PI 에서 SHR 로 로드됩니다. 보고서 팩 역시 단계별 방식으로 마이그레이션할 수 있습니다. 따라서 모든 데이터 마이그레이션 패키지를 사용할 수 있을 때까지 기다릴 필요가 없습니다.

Q2. SHR 에서 보고에 사용할 사용자 지정 노드 그룹을 만들 수 있습니까?

A2. SHR 에서 다음과 같이 xml 형식으로 사용자 지정 그룹을 정의할 수 있습니다.

```
<groups>
<group name="System_customgroup" type="CUSTOMGROUP">
<instances type="K_CI_System">
<instance>
<attribute name="Node_Name" value="node1.ind.hp.com" operator="EQUALS"
relation="OR" />
<attribute name="Node_Name" value="node1.ind.hp.com" operator="EQUALS"
relation="OR" />
<attribute name="Node_Name" value="node1.ind.hp.com" operator="EQUALS"
relation="OR" />
</instance>
</instances>
</groups>
</groups>
```

- *group name* - 사용자 지정 그룹 이름
 - *type* - 가급적 "CUSTOMGROUP" 또는 다른 고유한 유형(OM 에서 VIEWS 및 NODEGROUPS 와 같은 다른 그룹 유형과의 차별화를 위해)
 - *instances type* - 디멘션 인스턴스를 그룹으로 지정할 디멘션 테이블 이름.
 - *attribute_name* - 인스턴스 값을 제공할 디멘션 테이블의 열 이름.
 - *value* - 그룹을 생성하려는 열의 실제 데이터 값.
- xml 파일의 이름은 *customgroup.xml 형식이어야 하며 파일을 %PMDb_HOME%/config 폴더에 넣어야 합니다.
 - 다음 명령을 실행하여 .xml 을 처리합니다(.csv 파일을 생성하고 파일 준비).
 - `abcloadNrun -loadBatch -streamId CustomGroup@Platform`
 - `abcloadNrun -runStream -streamId CustomGroup@Platform`
 - 다음 쿼리를 사용하여 데이터가 스테이지 테이블에 로드되었는지 확인합니다.
 - `select * from K_CI_Group_Bridge_`
 - Core CP 의 디멘션 스트림이 로더 단계(타이머를 통한 호출 시)를 실행하며 동일한 항목을 데이터 웨어하우스 테이블 K_CI_Group_Bridge 로 로드합니다.

- SHR 보고서의 그룹 선택 프롬프트에 사용자 지정 그룹 값이 나오는지 확인합니다.

Q3. 노드/노드 그룹의 선택 세트에 대해 데이터 마이그레이션을 구성할 수 있습니까?

A3. 현재는 SHR의 OM 토폴로지 원본 시나리오에서만 가능합니다. 기본적으로 노드 필터링은 비활성화됩니다.

- 노드 필터링을 활성화하려면 `%PMDB_HOME%\config\collection.properties` 파일에서 다음과 같이 속성을 설정해야 합니다.
`om.filtering.enabled=true`
- 선택적으로 SHR로 가져와야 하는 노드 목록을 포함하는 `%PMDB_HOME%\config\filterednodes.conf` 파일을 만듭니다. 이 파일은 OM 콘솔에 표시되는 것과 동일한 노드 이름/FQDN을 포함해야 합니다.
- 선택적으로 OM에서 SHR로 가져와야 하는 그룹 목록을 포함하는 `%PMDB_HOME%\config\filteredgroups.conf` 파일을 만듭니다.
- 두 파일이 존재하고 올바른 값을 포함하는 경우 `%PMDB_HOME%\config\filterednodes.conf` 파일에 지정된 노드와 `%PMDB_HOME%\config\filteredgroups.conf` 파일에 지정된 각 그룹에 속한 모든 노드를 SHR로 가져옵니다.

참조

PI의 System Resource 리포트 팩 및 SHR의 System Performance 컨텐츠 간 데이터 매핑

이 섹션에서는 PI와 SHR의 데이터베이스 테이블/열 간의 매핑을 정리합니다. SHR 보고서에서 사용되지 않지만 PI에서 마이그레이션되는 데이터는 진한 빨간색으로 표시됩니다.

PI: K_Node	SHR: K_CI_System
NODE_NAME	DNS_Name Host_Name Name Display_Name
OPERATING_SYS	OS
MODEL	Model
MAKE	Vendor
MAKE	Manufacture
NODE_TYPE	HyperVisor_Type
SERIAL_NUM	Serial_Number
NODE_ID	
IP_ADDRESS	
DEPARTMENT	
SYSOBJECTID	
IP_STATE	
SYSOBJECTID	
	Host_Key
	OS_Version
	Internal_Name
	Server_Type
	isVirtual("FALSE"로 설정)
	Processor_Architecture
	Phys_Mem_GB
	CPU_Num
	CPU_Num_Core
	CPU_Speed_MHz
	Disk_Num
	Network_Num
	Node_Type
	CPUUtil_Threshold
	SwapUtil_Threshold
	MemUtil_Threshold
	RunQ_Threshold
	PageOut_Threshold
	StaticThresholdFlag
	CPUFamily
	CI_UID(토폴로지 원본에서 검색 - PI의

	NODE_NAME 이 조정에 사용됨)
	Creation_time
	Created_by
	Update_time
	Updated_by
	Description
	User_Key
	CPUUtil_STH1
	CPUUtil_STH2
	CPUUtil_STH3
	SwapUtil_STH1
	SwapUtil_STH2
	SwapUtil_STH3
	MemoryUtil_STH1
	MemoryUtil_STH2
	MemoryUtil_STH3
	RunQ_STH1
	RunQ_STH2
	RunQ_STH3
	CIT_Key
	Managed_BY
	Disk_Capacity_GB
	Total_Network_Speed
	State
	CPU_Unreserved
	Role
	HyperVisor_Type
	UUID
	ProcessorModel
	CPUCapacityGHZ
	Datacenter
	Standalone_View("1"로 설정)
	OSPatch
	Memory_Unreserved
	Cluster_Name
	Hypervisor_View

PI: SH_SR_SysXcep	SHR: SH_SM_Node_Res
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGrunq	avgRunQ
P95runq	P95RunQ
TOTrunq_grade	totRunQ_Grade
AVGrunq_grade	avgRunQ_Grade
AVGcpuutil	avgCPUUtil
TCTcpuutil	tctCPUUtil
P95cpuutil	P95CPUUtil

TOTcpuutil_grade	totCPUUtil_Grade
AVGcpuutil_grade	avgCPUUtil_Grade
AVGmemutil	avgMemUtil
TCTmemutil	tctMemUtil
P95memutil	P95MemUtil
TOTmemutil_grade	totMemUtil_Grade
AVGmemutil_grade	avgMemUtil_Grade
AVGswapUtil	avgSwapUtil
TCTswapUtil	tctSwapUtil
P95swapUtil	P95SwapUtil
TOTswapUtil_grade	totSwapUtil_Grade
AVGswapUtil_grade	avgSwapUtil_Grade
AVGmemPageOutRate	avgMemPageOutRate
TCTmemPageOutRate	tctMemPageOutRate
P95memPageOutRate	P95MemPageOutRate
TOTavgNumProcs	totNumProcs
AvgNumProcs	avgNumProcs
PI: SH_SR_SysVolXcep	SHR: SH_SM_Node_Res
TOTInPackets	totInPackets
TOTOutPackets	totOutPackets
TOTvolume	totVolume
	maxVolume
WAVvolume	avgVolume
	avgPacketRate
	maxPacketRate
AVGCollisionRate	avgCollisionRate
	maxCollisionRate
AVGErrorRate	avgErrorRate
	maxMemUtil
	P90MemUtil
	avgFreeMemGB
	maxRunQ
	P90RunQ
	maxCPUUtil
	P90CPUUtil
	maxSwapUtil
	P90SwapUtil
	P90MemPageOutRate
	avgDiskPhysIORate
	maxDiskPhysIORate
	P90DiskPhysIORate
	avgNetIORate
	maxNetIORate
	P90NetIORate
	totNumStartedProcs
	totNumActiveProcs
	maxErrorRate
	avgSysModeUtil
	maxSysModeUtil

	avgUsrModeUtil
	maxUsrModeUtil
	avgCSwitchRate
	maxCSwitchRate
	avgInterruptRate
	maxInterruptRate
	avgByteRate
	ubsavgMemoryUtil
	lbsavgMemoryUtil
	ubsavgRunQ
	lbsavgRunQ
	ubsavgCPUUtil
	lbsavgCPUUtil
	ubsavgSwapUtil
	lbsavgSwapUtil
	ubsavgDiskPhysIORate
	lbsavgDiskPhysIORate
	ubsByteRate
	lbsByteRate
	avgNetworkUtilMBps
	NetInByteRate
	NetOutByteRate
	avgDiskPhysreadbyteRate
	avgDiskPhyswritebyteRate
	avgReadlatency
	avgWritelatency
	VMNum
	DiskSpaceUtil

PI:SD_SR_SysXcep	SHR:SD_SM_Node_Res
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGrunq	avgRunQ
P95runq	P95RunQ
TOTrunq_grade	totRunQ_Grade
AVGrunq_grade	avgRunQ_Grade
AVGcpuutil	avgCPUUtil
TCTcpuutil	tctCPUUtil
P95cpuutil	P95CPUUtil
TOTcpuutil_grade	totCPUUtil_Grade
AVGcpuutil_grade	avgCPUUtil_Grade
AVGmemutil	avgMemUtil
TCTmemutil	tctMemUtil
P95memutil	P95MemUtil
TOTmemutil_grade	totMemUtil_Grade
AVGmemutil_grade	avgMemUtil_Grade

AVGswapUtil	avgSwapUtil
TCTswapUtil	tctSwapUtil
P95swapUtil	P95SwapUtil
TOTswapUtil_grade	totSwapUtil_Grade
AVGswapUtil_grade	avgSwapUtil_Grade
AVGmemPageOutRate	avgMemPageOutRate
TCTmemPageOutRate	tctMemPageOutRate
P95memPageOutRate	P95MemPageOutRate
TOTavgNumProcs	totNumProcs
AvgNumProcs	avgNumProcs
PI: SD_SR_SysVolXcep	SHR: SD_SM_Node_Res
TOTInPackets	totInPackets
TOTOutPackets	totOutPackets
TOTvolume	totVolume
	maxVolume
WAVvolume	avgVolume
	avgPacketRate
	maxPacketRate
AVGCollisionRate	avgCollisionRate
	maxCollisionRate
AVGErrorRate	avgErrorRate
	maxMemUtil
	P90MemUtil
	avgFreeMemGB
	maxRunQ
	P90RunQ
	maxCPUUtil
	P90CPUUtil
	maxSwapUtil
	P90SwapUtil
	P90MemPageOutRate
	avgDiskPhysIORate
	maxDiskPhysIORate
	P90DiskPhysIORate
	avgNetIORate
	maxNetIORate
	P90NetIORate
	totNumStartedProcs
	totNumActiveProcs
	maxErrorRate
	avgSysModeUtil
	maxSysModeUtil
	avgUsrModeUtil
	maxUsrModeUtil
	avgCSwitchRate
	maxCSwitchRate
	avgInterruptRate
	maxInterruptRate
	avgByteRate

	ubsavgMemoryUtil
	lbsavgMemoryUtil
	ubsavgRunQ
	lbsavgRunQ
	ubsavgCPUUtil
	lbsavgCPUUtil
	ubsavgSwapUtil
	lbsavgSwapUtil
	ubsavgDiskPhysIORate
	lbsavgDiskPhysIORate
	ubsByteRate
	lbsByteRate
	avgNetworkUtilMbps
	NetInByteRate
	NetOutByteRate
	avgDiskPhysreadbyteRate
	avgDiskPhyswritebyteRate
	avgReadlatency
	avgWritelatency
	VMNum

PI: SH_SR_SysUp	SHR: SH_SM_Node_Avail
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
TOTUPTIME	totuptime
TOTDOWNTIME	totdowntime
	Totplandtime
	Totexcdtime
	Totunknowntime
	Totavailability

PI: SD_SR_SysUp	SHR: SD_SM_Node_Avail
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
TOTUPTIME	totuptime
TOTDOWNTIME	totdowntime
	Totplandtime
	Totexcdtime
	Totunknowntime
	Totavailability

PI: K_System_CPU	SHR: K_SM_CPU
	dsi_key_id(자동 생성)
CPUID	cpu_id
(K_Node.node_name(K_System_CPU.node_fk))	node_name
	Systemref(자동 생성(ciid - 조정된 node_name 사용))
	cpu_vendor
	cpu_speed

PI: SH_SR_CPU	SHR: SH_SM_CPU
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGCPUUTIL	avgTotUtil
AVGCPUSYSTEMMODE	avgSysModeUtil
AVGCPUUSERMODE	avgUsrModeUtil
AVGINTRATE	avgInterruptRate
AVGCSRATE	avgCntxtSwitchRate
	maxTotUtil
	maxSysModeUtil
	maxUsrModeUtil
	maxInterruptRate
	maxCntxtSwitchRate

PI: SD_SR_CPU	SHR: SD_SM_CPU
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGCPUUTIL	avgTotUtil
AVGCPUSYSTEMMODE	avgSysModeUtil
AVGCPUUSERMODE	avgUsrModeUtil
AVGINTRATE	avgInterruptRate
AVGCSRATE	avgCntxtSwitchRate
	maxTotUtil
	maxSysModeUtil
	maxUsrModeUtil
	maxInterruptRate
	maxCntxtSwitchRate

PI: K_Disk_Dsk	SHR: K_SM_PhysicalDISK
	dsi_key_id(자동 생성)
prop_disk_name	disk_name
(K_Node.node_name(K_System_CPU.node_fk))	node_name
	Systemref(자동 생성(ciid - 조정된 node_name 사용))

	사용))
	Dir_name

PI: SH_SR_Disk	SHR: SH_SM_Disk
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGDISKUTIL	avgPctUtil
AVGPHYSICALIORATE	avgPhyIORate
AVGPHYSREADRATE	avgPhyReadRate
AVGPHYSWRITERATE	avgPhyWriteRate
AVGSYSTEMIORATE	avgSysIORate
AVGVMIORATE	avgVMIORate
	avgPhyByteRate
	avgPhyReadByteRate
	avgPhyWriteByteRate
	avgRawReadRate
	avgRawWriteRate
	maxPctUtil
	maxPhyIORate
	maxPhyReadRate
	maxPhyWriteRate
	maxSysIORate
	maxVMIORate
	maxPhyByteRate
	maxPhyReadByteRate
	maxPhyWriteByteRate
	maxRawReadRate
	maxRawWriteRate

PI: SD_SR_Disk	SHR: SD_SM_Disk
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
AVGDISKUTIL	avgPctUtil
AVGPHYSICALIORATE	avgPhyIORate
AVGPHYSREADRATE	avgPhyReadRate
AVGPHYSWRITERATE	avgPhyWriteRate
AVGSYSTEMIORATE	avgSysIORate
AVGVMIORATE	avgVMIORate
	avgPhyByteRate
	avgPhyReadByteRate
	avgPhyWriteByteRate
	avgRawReadRate
	avgRawWriteRate

	maxPctUtil
	maxPhyIORate
	maxPhyReadRate
	maxPhyWriteRate
	maxSysIORate
	maxVMIORate
	maxPhyByteRate
	maxPhyReadByteRate
	maxPhyWriteByteRate
	maxRawReadRate
	maxRawWriteRate

PI: K_NetInterface_NetIf	SHR: K_SM_NetInterface
	dsi_key_id(자동 생성)
prop_netif_name	Interface_name
(K_Node.node_name(K_System_CPU.node_fk)	Node_Name
	Systemref(자동 생성(ciid - 조정된 node_name 사용))
	Network_speed

PI: SH_SR_NetInterface	SHR: SH_SM_NetInterface
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
	avgPktRate
AVGINPACKETS	avgInPktRate
AVGOUTPACKETS	avgOutPktRate
	avgByteRate
AVGINBYTES	avgInByteRate
AVGOUTBYTES	avgOutByteRate
AVGCOLLISIONRATE	avgCollisionRate
AVGERRORRATE	avgErrorRate
	NetworkUtil
	maxPktRate
	maxInPktRate
	maxOutPktRate
	maxByteRate
	maxInByteRate
	maxOutByteRate
	maxCollisionRate
	maxErrorRate

PI: SD_SR_NetInterface	SHR: SD_SM_NetInterface
	dsi_key_id(자동 생성)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef(자동 생성)
	avgPktRate
AVGINPACKETS	avgInPktRate
AVGOUTPACKETS	avgOutPktRate
	avgByteRate
AVGINBYTES	avgInByteRate
AVGOUTBYTES	avgOutByteRate
AVGCOLLISIONRATE	avgCollisionRate
AVGERRORRATE	avgErrorRate
	NetworkUtil
	maxPktRate
	maxInPktRate
	maxOutPktRate
	maxByteRate
	maxInByteRate
	maxOutByteRate
	maxCollisionRate
	maxErrorRate

PI: K_Customer	SHR: K_Customer
	dsi_key_id(자동 생성)
cust_id	Customer_ID
cust_name	Name Display_Name
	User_Key
	Address1
	Address2
	City
	State
	ZIP_Code
	Phone_Number
	Description

PI: K_SR_System, K_Node, K_Customer	SHR: K_CI_Cust_Bridge
	dsi_key_id(자동 생성)
K_SR_System.node_fk(CI_UID 를 얻기 위해 조정)	CI_Key
K_Customer.cust_name	Cust_Key
	Valid_Period_Start
	Valid_Period_End

PI: K_Location	SHR: K_Location
	dsi_key_id(자동 생성)
	Country ('Default')
	State ('Default')
	City ('Default')
location_name	Region
	Address ('Default')
	Building ('Default')
	Floor ('Default')
location_name	Name

PI: K_SR_System, K_Node, K_Location	SHR: K_CI_Loc_Bridge
	dsi_key_id(자동 생성)
K_SR_System.node_fk(CI_UID 를 얻기 위해 조정)	CI_Key
K_Location.location_name	Loc_Key
	Valid_Period_Start
	Valid_Period_End