

HP Service Health Reporter

Softwareversion: 9.20

Performance Insight - Handbuch zur Datenmigration



Inhaltsverzeichnis

Einführung	2
Warum Migration?	2
Ziel der Migration	2
Vorteile der Migration	2
Umfang der Migration	3
Qualitätsmerkmale eines guten Kandidaten für die Datenmigration	4
Definition der Begriffe.....	4
Datenmigration - Architektur.....	5
Voraussetzungen.....	5
Datenmigration aus Performance Insight – OM-Bereitstellungsszenario	6
Datenmigration aus Performance Insight – BSM-Bereitstellungsszenario.....	7
Konzept der Datenmigration.....	7
Bekannte Probleme und Einschränkungen.....	8
Schritte bei der Datenmigration	10
Vorbedingungen	10
PI-Migrationsinhalt – Paket	10
Schritte bei der Datenmigration aus PI	10
Typische Anwendungsfälle/Szenarios	12
Häufig gestellte Fragen [FAQs].....	13
Referenzen.....	15
Datenzuordnung zwischen System Resource Report Pack von PI und systemleistungsbezogenem Inhalt in SHR	15

Einführung

HP Performance Insight ist eine Applikation für das Leistungsmanagement und die Erstellung der entsprechenden Reports. Die Migration vorhandener Performance Insight-Kunden zu anderen Reporting-Lösungen im BSM-Portfolio wird als Mehrphasenkonzept geplant.

Warum Migration?

HP PI wird irgendwann einmal veraltet sein. Das Migrationsprogramm soll den aktuellen Performance Insight-Kunden (mit einem gültigen Support-Vertrag) den Übergang zu anderen BSM-Reporting-Tools erleichtern.

HP PI-Kunden, die eine Migration zu einem oder mehreren Produkten benötigen (je nach Anwendungsfall)

- Service Health Reporter
- NNMi + Performance iSPI

Ziel der Migration

- Den HP Performance Insight-Kunden soll eine nahezu äquivalente Funktionalität ermöglicht werden, da HP dazu übergeht, die BSM-Reporting-Tools zu konsolidieren.
- HP PI-Kunden könnten sich für die Migration zu Service Health Reporter entscheiden, wodurch sie erweiterte Funktionen wie domänenübergreifende Analysen und Reports nutzen können.

Dieses Dokument enthält Details zum Prozess der Migration von Daten aus dem HP PI System Resource Report Pack zu SHR zur Nutzung des systemleistungsbezogenen Inhalts von SHR.

Vorteile der Migration

Die Migration zu Service Health Reporter (Phase 1) hat unter anderem die folgenden Vorteile:

- Upgrade zur Reporting-Lösung der nächsten Generation im BSM-Portfolio
 - SHR erstellt modellabhängige Reports => bindet die Applikationsleistung an die zugrunde liegende Infrastruktur
 - Ermöglicht eine ganzheitliche Sicht auf Ihre Umgebung
 - Topologieaktualisierungen werden automatisch nachverfolgt
 - Zentraler Überblick über die Leistung von Applikationen, Infrastruktur, Datenbank und Netzwerk für die Wartung
- Bringt Business Intelligence in das Datacenter-Management
 - SAP Business Objects Enterprise als Reporting-Framework
 - Einfachere Anpassung
- Zusätzliche Funktionalität
 - Berücksichtigung der Ausfallzeiten / Benutzerdefinierte Gruppierung / Anpassbare Schichten usw. in Reports
 - Eine Plattform für zukünftige Aspekte der Analyse, wie zum Beispiel Planung der Virtualisierungskapazität usw.
- Skalierbarkeit
 - Eine einzelne SHR-Instanz kann Daten direkt von 5000 Systemknoten sammeln.

Der Zweck dieses Dokuments besteht in der Erläuterung der folgenden Themen:

- Konzept der Datenmigration aus Performance Insight zu SHR
- Schritte bei der Datenmigration aus Performance Insight für den systemleistungsbezogenen Inhalt
- Voraussetzungen
- Bekannte Einschränkungen
- Aspekte von Skalierbarkeit und Leistung

Umfang der Migration

Die Datenmigration aus Performance Insight (Phase 1) ist in ihrem Inhalt auf das System Resource Report Pack und die folgenden Packages beschränkt:

- SystemResourceCPU
- SystemResource_Disk
- SystemResource_NetInterface

SHR bietet dann die folgenden Funktionen:	Die folgenden Funktionen bietet SHR nicht:
<p>Unterstützt Anwendungsfälle bei der kontinuierlichen Verbesserung von Services und bei der Erstellung historischer Reports</p> <ul style="list-style-type: none"> - konsolidierte Reports - Szenarios mit und ohne BSM <p>Sammelt Daten aus zwischengeschalteten Datenspeichern</p> <ul style="list-style-type: none"> - Ausnahme: Datensammlung mit Performance Agent 	<p>Reports in SHR enthalten historische Daten und Zusammenfassungen, die nicht in Echtzeit ermittelt wurden und nicht für Anwendungsfälle bei der Fehlerbehebung vorgesehen sind.</p> <ul style="list-style-type: none"> - Keine vorgefertigten Reports für Diagnose oder Fehlerbehebung
<p>Einmalige Datenmigration aus PI</p> <ul style="list-style-type: none"> - Nach der Migration der Daten wird vorausgesetzt, dass die Sammlung von Echtzeitdaten aus den Quellen (Agenten für den systemleistungsbezogenen Inhalt und HPOM/RTSM für die Topologie) in SHR konfiguriert ist. 	<p>Die einmalige Datenmigration von PI enthält nicht:</p> <ul style="list-style-type: none"> - HPOM-SPIs (inkl. SPI for Virtualization) - Schnittstellen-Reports/Geräteressourcen, Datenverkehr, MPLS, QoS, IP-Telefonie, Servicegewährleistung und sonstige Metriken des SPI for Network Performance
<p>Aus PI werden nur aggregierte Daten (stündliche und tägliche Körnung) und keine Satzdaten migriert.</p> <p>Konfigurationen werden nicht migriert.</p>	<p>Verwendet dann PI nicht als Quelle für die kontinuierliche Datensammlung</p>
<p>Unterstützt Anpassung und Erweiterung des Inhalts über Content Development Environment (CDE).</p> <ul style="list-style-type: none"> - Einzelne Reports in PI werden nicht automatisch in vordefinierte SHR-Inhalte migriert. Für zusätzliche Anwendungsfälle können aber benutzerdefinierte Reports erstellt werden. 	<p>Erfasst keine Daten mit niedriger Latenz</p> <ul style="list-style-type: none"> - Keine Echtzeitdaten in Reports

- Vordefinierte Inhalte können auch angepasst werden.	
Erstellung von benutzerdefinierten Gruppen für Reports wird unterstützt	Wird auf Linux/UNIX-Versionen ODER Oracle/Sybase ASE nicht unterstützt – Sybase IQ ist die einzige unterstützte Datenbank.

Qualitätsmerkmale eines guten Kandidaten für die Datenmigration

Die folgenden Merkmale könnten helfen, um einen guten Kandidaten für die Datenmigration zu finden:

- Verwendet PI vorrangig für:
 - Systeme (OM Nachrichten/SPI for Databases/Virtualisierung und SiteScope)
 - Netzwerkdaten (die durch Perf/iSPI gesammelt werden)
- Größe der Umgebung
 - ~ 5000 Knoten für SHR
- Fachwissen zur Anpassung von Reports auf Basis von SAP Business Objects
- Gutes Verständnis für die zugrunde liegende Domäne (zum Erstellen von benutzerdefinierten Reports)
- Aktueller OM- oder BSM-Benutzer
- Auf der Suche nach einer Reporting-Lösung mit neueren Möglichkeiten als in PI
- Schichten, Ausfallzeiten, Benutzerdefinierte Gruppierung usw.

Definition der Begriffe

Begriff	Definition
CP	Content Pack
RP	Report Pack
DWH	Data Warehouse
HP PI	HP Performance Insight
BSM	Business Service Management
RTSM	Run-Time Service Model
OM	Operations Manager
CDE	Content Development Environment (Umgebung für die Inhaltentwicklung)
SPI	Smart Plug-In
SHR	Service Health Reporter
ETL	Extract, Transform and Load (Extrahieren, Transformieren und Laden)
ABC	Audit, Balance and Control (Prüfen, Ausgleichen und Steuern)

Datenmigration - Architektur

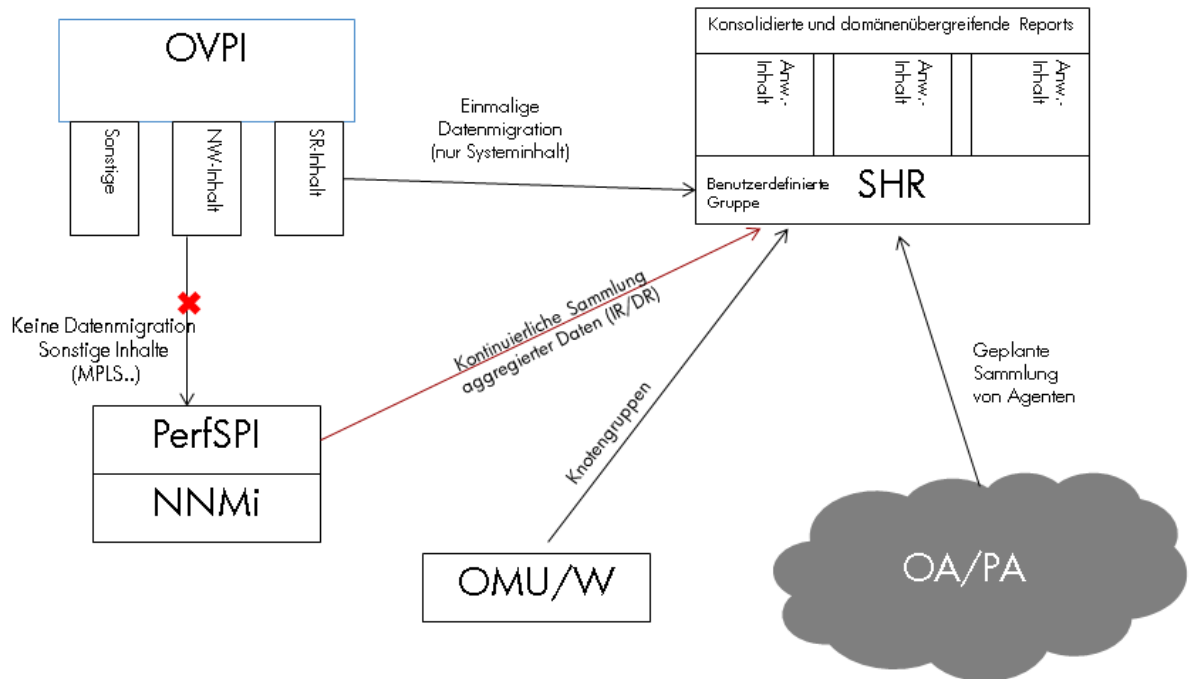
In Phase 1 erfolgt die Migration von PI-Kunden zu einem oder mehreren BSM-Produkten (je nach Anwendungsfall). In diesem Abschnitt werden das verwendete Architekturkonzept für die Datenmigration mit den entsprechenden Voraussetzungen, die damit erfassten Szenarios und die bekannten Einschränkungen beschrieben.

Voraussetzungen

- PI ist keine Datenquelle für die kontinuierliche Datensammlung (Konzept der einmaligen Datenmigration).
- Der PI-Kunde ist ein aktueller OM- oder BSM-Benutzer.
- Die Größe der Reporting-Lösung muss unter der Größe liegen, die durch SHR 9.20 unterstützt wird.
- SHR wird auf einem separaten Server installiert (es wird keine Koexistenz auf dem PI-Server erwartet).
- Nur die Versionen von 5.3 und 5.4x von HP PI werden unterstützt – frühere Versionen von PI (und nicht unterstützte Konfigurationen von HP PI) werden als Datenquelle für die Migration nicht unterstützt.
- Die Konfiguration der Topologiequelle in SHR ist eine einmalige Konfiguration und muss vor/nach der Migration konsistent sein.
- Das Datenmigrationspaket übernimmt die Dimensions- und Faktdaten aus PI. Vom Benutzer wird erwartet, dass er in SHR ein ETL-Package (zum Beispiel SysPerf_ETL_PerformanceAgent) für die folgenden Zwecke installiert:
 - Übernahme der Topologiedaten (aus den entsprechenden Topologiequellen – zum Beispiel Knotengruppen in OM und Geschäftsservice-basierte Topologie in RTSM)
 - Ständige Sammlung von Echtzeitdaten aus Agenten (da dies eine einmalige Datenmigration ist)

In den folgenden Abschnitten wird das Migrationskonzept (Phase 1) beschrieben.

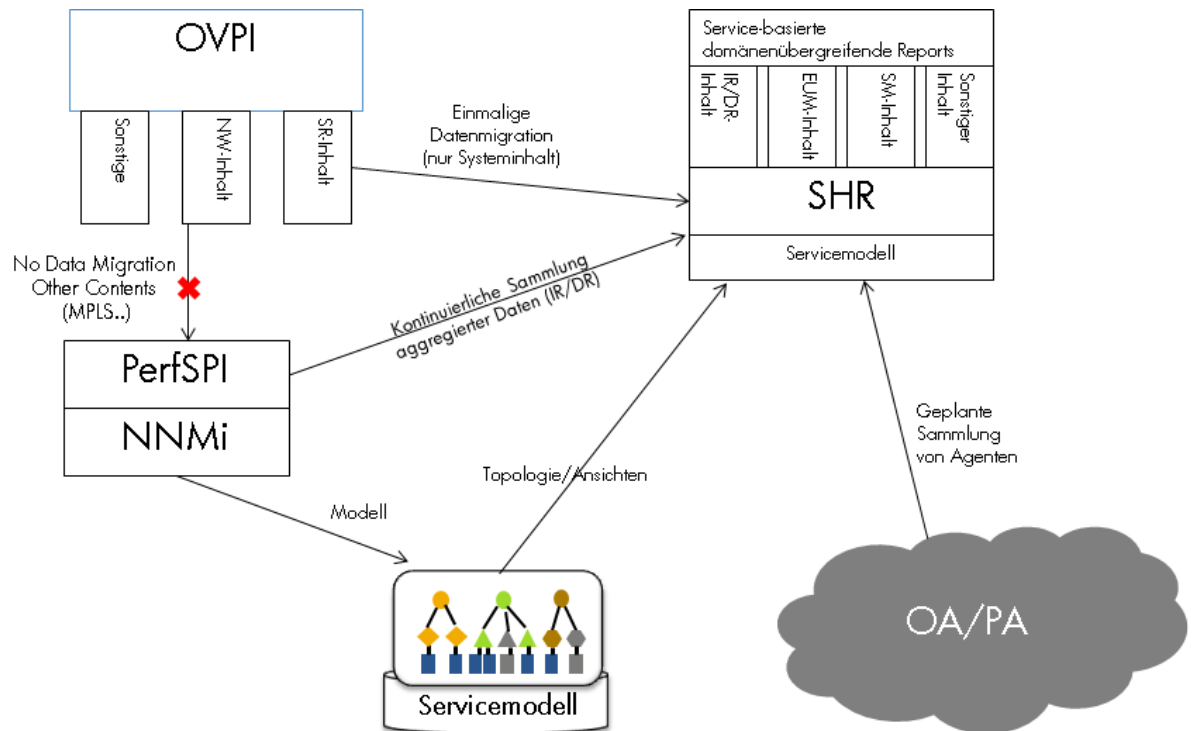
Datenmigration aus Performance Insight – OM-Bereitstellungsszenario



Im OM-Bereitstellungsszenario beschafft das Datenmigrationspaket die Topologieinformationen von OM-Quellen (Unix/Linux/Windows), die für die Topologiesammlung in SHR konfiguriert wurden. Die Daten werden für die Knoten, die durch die als Topologiequelle konfigurierte OM-Instanz überwacht werden, aus den stündlichen und täglichen Tabellen zum PI System Resource-Inhalt zu SHR migriert.

Sobald die Datenmigration abgeschlossen ist, muss der SysPerf_ETL_PerformanceAgent installiert werden, um die Informationen zu Topologie- und Knotengruppen und die Echtzeit-Satzdaten aus Agenten in SHR zu übernehmen.

Datenmigration aus Performance Insight – BSM-Bereitstellungsszenario



Im BSM-Bereitstellungsszenario beschafft das Datenmigrationspaket die Topologieinformationen von der RTSM-Instanz, die für die Topologiesammlung in SHR konfiguriert wurde. Die Daten werden für die Knoten, die in der SM_PA-Ansicht in RTSM vorhanden sind, aus den stündlichen und täglichen Tabellen zum PI System Resource-Inhalt zu SHR migriert.

Sobald die Datenmigration abgeschlossen ist, muss der SysPerf_ETL_PerformanceAgent installiert werden, um die service-basierten Topologieinformationen (aus SHR-Ansichten wie SM_PA in RTSM) und die Echtzeit-Satzdaten aus Agenten in SHR zu übernehmen.

Konzept der Datenmigration

Das Konzept der Datenmigration nutzt das ETL-Framework und das Koordinierungs-Framework (Audit, Balance and Control (ABC)) von SHR. Die folgende Übersicht zeigt den Datenfluss durch die einzelnen ABC-Streams:

- Topologiedaten sammeln
- Abstimmungs-Registry erstellen
- Daten (Dimensionsdaten, stündliche und tägliche Faktdaten) aus PI (Oracle und Sybase ASE) mit Datenbankcollector extrahieren (Die betroffenen PI-Tabellen werden im Abschnitt "Referenzen" in diesem Dokument beschrieben.)
- Daten aus PI mit Topologiequelle abstimmen
- Daten mit Loader in SHR-Aggregationstabellen laden

Topologiedaten sammeln

Die Daten werden aus der in SHR konfigurierten Topologiequelle (RTSM und OM) gesammelt. Die Topologiesammlung ist notwendig, um die Abstimmungs-Registry zu erstellen, die für die Abstimmung der Faktdaten aus PI genutzt werden soll.

Hinweis: Das Migrationspaket ist nicht für die Sammlung detaillierter Dimensions- und Topologiedaten zuständig. Dies erfolgt nach der Migration, sobald der Echtzeit-Datenfluss gestartet wurde.

Extrahieren der Daten aus PI

Unterstützt wird die Extraktion von Daten aus den Oracle-/Sybase ASE-Datenbanken, die durch HP PI verwendet werden. Für diesen Zweck nutzt das Migrationspaket den SHR-Mechanismus zur Datenbanksammlung.

Beim Extrahieren der Daten muss das Content Pack für die Migration mit der stündlichen Tabelle in der PI-Datenbank beginnen. Da SHR nur stündliche und tägliche Tabellen für die Systemleistung enthält, erfolgt die Datenmigration von ihren Gegenstücken im System Resource Report Pack von PI (und den zugehörigen Unterpaketen). Die jährlich und monatlich aggregierten Daten und Satzdaten aus PI werden nicht in SHR migriert.

Direkt nach der Datenmigration könnte die Satzdatensammlung in SHR mit einer passenden ersten Historie gestartet werden, um Rohmetriken aus HP Performance Agents zu übernehmen.

Verarbeiten der Daten aus HP PI

Der SHR-Datenbankcollector speichert die gesammelten Daten im Ordner '%PMDB_HOME%\collect' und muss mit dem Schritt 'Sammeln' für die weitere Verarbeitung in den Ordner 'stage' kopiert werden. Die aus PI gesammelten Daten enthalten die Topologieinformationen nicht (und sind auch nicht mit der CI_UID angereichert). Deshalb müssen die gesammelten Daten mit den aus einer Topologiequelle gesammelten Daten abgestimmt werden.

Laden der HP PI-Daten in SHR

Das Lademodul von SHR wird verwendet, um die stündlich und täglich aggregierten Faktdaten und auch die Dimensions- und Standort/Kunden-Bridge-Daten, die aus PI gesammelt wurden, zu laden.

Bekannte Probleme und Einschränkungen

- Elemente, die nicht migriert werden:
 - Aus PI werden nur die Daten (Metriken und Dimensionen) übernommen, die dem Datenmodell in SHR zugeordnet werden können (zum Beispiel werden Dateisystemdaten und Prognosemetriken nicht migriert). Eine detaillierte Liste der Zuordnung zwischen PI und SHR finden Sie im Abschnitt "Referenzen" in diesem Dokument.
 - Konfigurationsdetails (Sammlungskonfiguration, fest codierte Standardschichten in PI, Reportkonfigurationen wie Zeitpläne usw.) werden nicht zu SHR migriert.
 - Da die Funktion für geplante Ausfallzeiten in PI nicht unterstützt wurde, enthält der migrierte Datenbestand keine entsprechenden Daten.
 - SHR 9.20 enthält keine vorgefertigten Reports nach Kunde. Die Kundendimensionsdaten und die zugehörigen Zuordnungen zur Knotendimension

werden aber zu SHR migriert (ein benutzerdefiniertes Universum mit Reports könnte daraus erstellt werden).

- Da PI keine konsistente Zeitzone unterstützt, ist der migrierte Datenbestand mit der Zeitzone der Quelle als Basis verfügbar. In SHR 9.20 hat der Benutzer die Möglichkeit, die Zeitzone auf GMT oder auf eine lokale Zeitzone festzulegen. Bei der Datenmigration erfolgt jedoch keine Konvertierung der Zeitzone, wenn sich PI und SHR hinsichtlich der Zeitzone unterscheiden. Die migrierten Daten wären dann mit der Zeitzone der Quelle als Basis verfügbar und alle neu gesammelten Daten (nach der Installation von SysPerf_ETL_PerformanceAgent) mit der in SHR konfigurierten Zeitzone als Basis.
- Die aus PI übernommenen Dimensionsattribute werden wahrscheinlich mit denen aktualisiert, die mit der Sammlung von Echtzeitdaten in SHR erfasst werden.
- Alle in PI verfügbaren stündlichen und täglichen Daten werden gesammelt. Die Anpassung eines Intervalls oder Filterung solcher Daten ist im Migrationsinhalt derzeit nicht verfügbar.
- Historische Reports in SHR enthalten migrierte Daten aus PI nur in dem Umfang, wie es die in PI konfigurierte Beibehaltung vorsah (stündlich/täglich). Wenn in PI stündliche Daten nur 7 Tage lang beibehalten werden sollten, erbringt ein Drilldown von täglichen Daten vor dem 7-Tages-Zeitraum keine stündlich aggregierten Daten mehr.
- Das Datenmigrationspaket übernimmt aus PI die Faktdaten der Knoten, die als Teil der konfigurierten OM/RTSM-Topologiequelle verfügbar sind.
- Die aus PI in SHR übernommenen Metriken "Ausnahme" und "Grade of Service" werden anhand der dann in PI konfigurierten Schwellenwerte berechnet. Die Sammlung von Echtzeitdaten in SHR übernimmt Daten, die für die Berechnung dieser Metriken in SHR konfigurierte Schwellenwerte verwendet – die in PI und SHR konfigurierten Schwellenwerte sind möglicherweise nicht konsistent.
- Auf die migrierten Daten werden nur Standardschichten angewendet – es wird nicht erwartet, dass vor der Datenmigration Schichten in SHR konfiguriert sind (die Anreicherung mit Schichten könnte den ordnungsgemäßen Zustand der zu migrierenden Daten beeinträchtigen). Nach der Migration können Schichten konfiguriert werden, sodass auf die ab dann gesammelten Daten relevante Schichten angewendet werden.
- Derzeit gibt es ein bekanntes Problem bei der Konfiguration einer generischen Datenbank in der Verwaltungsoberfläche. Wenn die Performance iSPI-Datenbank (oder eine andere externe Datenbank) bereits als Datenquelle für die generische Datenbank konfiguriert ist und so die Migration aus PI erfolgt, besteht bei den implementierten Sammlungsrichtlinien die Gefahr, dass Daten aus beiden Quellen gesammelt werden. In einem solchen Fall erfolgt die Sammlung für die richtige Quelle ordnungsgemäß. Für die andere Quelle werden jedoch in den Protokolldateien erfasst und für den ABC-Stream wird der Status WARNUNG angezeigt.

Schritte bei der Datenmigration

Vorbedingungen

Vor dem Starten der Datenmigration des systemleistungsbezogenen Inhalts aus PI zu SHR müssen die folgenden Bedingungen erfüllt sein:

- Der Timer-Service in PI ist gestoppt (und bleibt es während der gesamten Migration).
- Der Collection- und der Timer-Service in SHR wurde deaktiviert.
- In der Verwaltungsoberfläche in SHR sind keine Schichten konfiguriert (nur die Standardschichten sollten verfügbar sein). Dies ist sehr wichtig – wenn konfigurierte Schichten in SHR vorhanden sind, könnte dies den ordnungsgemäßen Zustand der aus PI geladenen aggregierten Daten beeinträchtigen.

PI-Migrationsinhalt – Paket

Das PI-Datenmigrationspaket (`HPSHRSmPIMgr.msi`) wird auf Datenträger und im Live Network verfügbar sein. Dasselbe Paket kann für das RTSM- und für das OM-Bereitstellungsszenario verwendet werden und besitzt eine Abhängigkeit vom Domäneninhalt von System Management (`SysPerf_Domain`).

Schritte bei der Datenmigration aus PI

Installieren des Migrationspakets

- Installieren Sie das PI-Migrationspaket von der SHR-DVD (`HPSHRSmPIMgr.msi`). Dabei wird das Datenmigrationspaket (`Migration_SM_PI\Migration_SM_PI.ap`) in den Ordner `%PMDB_HOME%/packages` extrahiert.
- Implementieren Sie das Paket mit dem Bereitstellungs-Manager. (Stellen Sie sicher, dass der Collection- und der Timer-Service NICHT aktiv sind, bevor Sie mit der Datenmigration fortfahren.)

Konfigurieren der Datenquellen

Im nächsten Schritt werden die Topologie- und PI-Datenbankquellen für die Datensammlung konfiguriert.

- Die Details der Topologiequellen, die für die Topologiesammlung erforderlich sind, müssen auf der Seite zum Konfigurieren der OM/RTSM-Topologiequelle in der Verwaltungsoberfläche in SHR konfiguriert werden.
- Die Details der HP PI-Datenbank müssen als Konfiguration einer generischen Datenbank in der Verwaltungsoberfläche in SHR konfiguriert werden.

(Hinweis: Informationen zur zusätzlichen Konfiguration bei einer dezentralen Implementierung von PI finden Sie im Abschnitt "Anwendungsfälle" in diesem Dokument.)

Ausführung des Datenmigrationsskripts

Um die Datenmigration zu starten, führen Sie auf der Befehlszeile das Datenmigrationsskript wie folgt aus:

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

Da die Abfolge der Datenmigrationen durch das Framework zur Datenflusskoordination (ABC-Framework) in SHR unterstützt wird, ist der Status der Schritte bei der Migration in der Verwaltungsoberfläche verfügbar.

Überprüfung der migrierten Daten

Für den Vergleich der Anzahl der in PI verfügbaren Zeilen mit den in SHR importierten Daten ist ein Hilfsprogramm verfügbar.

Zur Überprüfung der Daten müssen die folgenden Schritte ausgeführt werden.

Kopieren Sie die Datei Systems.xml aus dem PI-System (%DPIPE_HOME%/data) in den SHR-Pfad %PMDB_HOME%/data.

```
ovpiupgrade -ovpi <PI-Pfad zu *.csv> -shr92 <SHR-Pfad zu *.csv> -xmlPath <absoluter Pfad zur XML mit den Zuordnungen> -report <absoluter Pfad zum Ordner für den zusammenfassenden Report>
```

Wenn Unterschiede in der Anzahl der migrierten Zeilen festgestellt werden, sind dafür die folgenden Ursachen möglich:

- i) Abstimmungsfehler, da der Knoten in PI in der Topologiequelle nicht verfügbar ist
- ii) Abweichungen zwischen dem Wert eines natürlichen Schlüssels in PI (zum Beispiel Knotenname) und dem entsprechenden Wert in der Topologiequelle
- iii) Das Tool führt einen Vergleich der Gesamtzahl der Knoten in PI durch und berücksichtigt dabei nicht die benutzerdefinierte Filterung der Knoten – dies ist eine Einschränkung.

Relevante Protokolldateien

Die folgenden Protokolldateien im Ordner %PMDB_HOME%/log enthalten dann die Details, die bei der Installation/Ausführung des Datenmigrationspakets protokolliert wurden:

- Packagemanager.log – Details zur Implementierung des Migrationspakets (und der zugehörigen Abhängigkeit SysPerf_Domain) mit dem Bereitstellungs-Manager.
- Topologycollector.log – Details zur OM/RTSM-Topologiesammlung
- ReconcileStep.log – Details zur erfolgten Abstimmung mit den Dimensionsdaten aus der konfigurierten Topologiequelle
- Dbcollector.log – Details zur Ausführung von Abfragen in der HP PI-Datenbank und der Verknüpfungsabfragen in der Extraktionsphase vor dem Generieren der CSV-Ausgangsdatei
- Dwablauncher.log – Details zur Ausführung des ABC-Streams
- Loader.log – Details zum letzten Schritt, bei dem die stündlich und täglich aggregierten Daten in SHR-Tabellen geladen werden

Deinstallieren des Migrationspakets

Nach der erfolgreichen Migration der Daten muss das Migrationspaket deinstalliert werden.

Das Datenmigrationspaket kann mit dem Bereitstellungs-Manager deinstalliert werden.

Zusätzlich sollte das PI-Migrationspaket (HPSHRSmPIMgr.msi) deinstalliert werden. Dadurch wird auch sichergestellt, dass das Migrationspaket nach der Migration nicht mehr im Bereitstellungs-Manager ausgewählt werden kann.

Neustart der SHR-Services und Installation des SystemPerformance-ETL-Package

SHR-Services, wie zum Beispiel der Collection- und der Timer-Service können jetzt aktiviert und gestartet werden.

Für den systemleistungsbezogenen Inhalt installieren Sie das ETL-Package (SysPerf_ETL_PerformanceAgent), damit neben den Echtzeitdaten für die Reports auch die benötigten Topologiedaten aus Performance Agents gesammelt werden.

Hinweis: Das Content Pack für die Migration aus PI migriert nur die Fakt- und Dimensionsdaten aus PI – die Sammlung von Topologie-/Dimensionsdaten aus den konfigurierten Topologiequellen erfolgt erst, nachdem das relevante ETL-Package installiert wurde und die Topologiesammlungen einmal komplett ausgeführt wurden. Deshalb müssen unbedingt der Collection- und der Timer-Service gestartet werden, bevor Daten in Reports angezeigt werden können.

Typische Anwendungsfälle/Szenarios

Verteilte Umgebung – Satelliten- und Zentralserver

In diesem Fall muss der Benutzer nach dem Konfigurieren der Details zu den PI-Satellitenservern und -Zentralservern in der Verwaltungsoberfläche in SHR den Migrationsstream mehrmals ausführen. Wenn 2 Satellitenserver und 1 Zentralserver vorhanden sind, muss der Benutzer also zuerst die Details zum ersten Satellitenserver in der Verwaltungsoberfläche konfigurieren und die Datei Migrate_pi_shr.pro ausführen. Nach der Ausführung des Streams muss er die Details zum zweiten Satellitenserver konfigurieren und den Stream ausführen. Für alle Satelliten- und Zentralserver müssen dieselben Schritte ausgeführt werden. (Bei einer kleineren Umgebung können auch zuerst alle Quellen konfiguriert werden – Satelliten- und Zentralserver in der Verwaltungsoberfläche mit anschließender Ausführung der Datei Migrate_pi_shr.pro.)

Hinweis: Wenn stündliche Daten im Zentralserver selbst verfügbar sind (je nach der Kopierrichtlinie, die der Kunde in HP PI definiert hat), ist es nicht notwendig, Daten aus den einzelnen Satellitenservern zu sammeln.

Häufig gestellte Fragen [FAQs]

In diesem Abschnitt wird versucht, einige häufig gestellte Fragen zum Datenmigrationsprozess zu beantworten.

F1. Ist es möglich, Daten aus PI zu migrieren, wenn SHR nach der Installation bereits einige Monate ausgeführt wurde? Wenn weitere Report Packs implementiert wurden, muss dann die Migration aufgeschoben werden, bis alle erforderlichen Migrationspakete verfügbar sind?

A1. Die Migration von Daten aus PI kann auch gestartet werden, wenn SHR nach der Installation bereits eine gewisse Zeit ausgeführt wurde. Beachten werden muss aber, dass aus PI die Daten für den Zeitraum migriert werden, für den in SHR noch keine Daten gesammelt wurden. Wenn zum Beispiel SHR bereits Daten für die letzte Woche gesammelt und aggregiert hat, dann werden aus PI nur die Daten für die Zeit vor diesem Zeitraum in SHR geladen.

Report Packs können sehr gut in einem Mehrphasenkonzept migriert werden. Daher ist es nicht notwendig, zu warten, bis alle Datenmigrationspakete verfügbar sind.

F2. Können benutzerdefinierte Knotengruppen erstellt und dann für die Reports in SHR verwendet werden?

A2. Benutzerdefinierte Gruppen können in SHR wie folgt in einem XML-Format definiert werden:

```
<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=" node2.ind.hp.com " operator="EQUALS"
          relation="OR" />
        <attribute name="Node_Name" value=" node3.ind.hp.com " />
      </instance>
    </instances>
  </group>
</groups>
```

- *group name* – Name der benutzerdefinierten Gruppe
 - *type* - vorzugsweise "CUSTOMGROUP" oder ein anderer einzigartiger Datentyp (zur Unterscheidung von anderen Gruppentypen wie VIEWS für Ansichten und NODEGROUPS für Knotengruppen in OM)
 - *instances type* – Name der Dimensionstabelle mit den Dimensionsinstanzen, die bei der Gruppierung verwendet werden können.
 - *attribute name* - Name der Spalte in der Dimensionstabelle, für die Instanzwerte angegeben werden können.
 - *value* - die eigentlichen Datenwerte der Spalten, für die eine Gruppe erstellt werden soll.
- Die XML-Datei sollte einen Namen im Format *customgroup.xml erhalten und im Ordner %PMDB_HOME%/config abgelegt werden.
 - Führen Sie die folgenden Befehle aus, um die XML-Datei zu verarbeiten (CSV-Dateien generieren und bereitstellen):
 - `abcloadNrun -loadBatch -streamId CustomGroup@Platform`
 - `abcloadNrun -runStream -streamId CustomGroup@Platform`

- Verwenden Sie die folgende Abfrage, um zu überprüfen, ob die Daten in die Staging-Tabellen geladen wurden:
 - `select * from K_CI_Group_Bridge_`
- Der Dimensionsstream in Core CP führt dann den Loader-Schritt aus (nach Aufruf über den Timer) und lädt dasselbe in die Data Warehouse-Tabelle K_CI_Group_Bridge.
- Stellen Sie sicher, dass die Werte für die benutzerdefinierten Gruppen in den Eingabeaufforderungen zur Gruppenauswahl in SHR-Reports angezeigt werden.

F3. Ist es möglich, die Datenmigration für ausgewählte Knoten/Knotengruppen zu konfigurieren?

A3. Derzeit ist dies nur im Szenario mit OM-Topologiequellen von SHR möglich. Die Knotenfilterung ist standardmäßig deaktiviert.

- Um die Knotenfilterung zu aktivieren, müssen Sie in der Datei `%PMDB_HOME%\config\collection.properties` die folgende Eigenschaft festlegen:


```
om.filtering.enabled=true
```
- Erstellen Sie eine Datei `%PMDB_HOME%\config\filterednodes.conf`, die die Liste der Knoten enthält, die selektiv in SHR importiert werden müssen. Dabei müssen die Knotennamen/FQDNs so angegeben werden, wie sie in der OM-Konsole angezeigt werden.
- Erstellen Sie eine Datei `%PMDB_HOME%\config\filteredgroups.conf`, die die Liste der Gruppen aus OM enthält, die selektiv in SHR importiert werden müssen.
- Wenn diese beiden Dateien vorhanden sind und gültige Werte enthalten, dann werden neben den Knoten, die in der Datei `%PMDB_HOME%\config\filterednodes.conf` angegeben sind, alle Knoten, die zu den einzelnen Gruppen unter `%PMDB_HOME%\config\filteredgroups.conf` gehören, in SHR importiert.

Referenzen

Datenzuordnung zwischen System Resource Report Pack von PI und systemleistungsbezogenem Inhalt in SHR

Dieser Abschnitt enthält eine Zuordnung zwischen den Datenbanktabellen/Spalten in PI und in SHR. Spalten, deren Daten aus PI migriert werden, die aber in SHR-Reports nicht verwendet werden, sind dunkelrot dargestellt.

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 (Wert "FALSE")
	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 (Abgerufen von Topologiequelle – NODE_NAME von PI zur Abstimmung verwendet)
	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 (Wert "1")
	OSPatch
	Memory_Unreserved
	Cluster_Name
	Hypervisor_View

PI: SH_SR_SysXcep	SHR: SH_SM_Node_Res
	dsi_key_id(automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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(automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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 (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
TOTUPTIME	totuptime
TOTDOWNTIME	totdowntime
	Totplandtime
	Totexcdtime
	Totunknowntime
	Totavailability

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

PI: K_System_CPU	SHR: K_SM_CPU
	dsi_key_id (automatisch generiert)
CPUID	cpu_id
(K_Node.node_name(K_System_CPU.node_fk)	node_name
	Systemref (automatisch generiert (ciid – abgestimmt mit node_name))
	cpu_vendor
	cpu_speed

PI: SH_SR_CPU	SHR: SH_SM_CPU
	dsi_key_id (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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 (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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 (automatisch generiert)
prop_disk_name	disk_name
(K_Node.node_name(K_System_CPU.node_fk)	node_name
	Systemref (automatisch generiert (ciid – abgestimmt mit node_name))
	Dir_name

PI: SH_SR_Disk	SHR: SH_SM_Disk
	dsi_key_id (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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 (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
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 (automatisch generiert)
prop_netif_name	Interface_name
(K_Node.node_name(K_System_CPU.node_fk)	Node_Name
	Systemref (automatisch generiert (ciid – abgestimmt mit node_name))
	Network_speed

PI: SH_SR_NetInterface	SHR: SH_SM_NetInterface
	dsi_key_id (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
	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 (automatisch generiert)
ta_period	ta_period
K_Node.K_Location.dsi_key_id	LocationRef
	ShiftRef (automatisch generiert)
	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 (automatisch generiert)
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 (automatisch generiert)
K_SR_System.node_fk(abgestimmt zum Abrufen von CI_UID)	CI_Key
K_Customer.cust_name	Cust_Key
	Valid_Period_Start
	Valid_Period_End

PI: K_Location	SHR: K_Location
	dsi_key_id (automatisch generiert)
	Country ('Standard')
	State ('Standard')
	City ('Standard')
location_name	Region
	Address ('Standard')
	Building ('Standard')
	Floor ('Standard')
location_name	Name

PI: K_SR_System, K_Node, K_Location	SHR: K_CI_Loc_Bridge
	dsi_key_id (automatisch generiert)
K_SR_System.node_fk(abgestimmt zum Abrufen von CI_UID)	CI_Key
K_Location.location_name	Loc_Key
	Valid_Period_Start
	Valid_Period_End