TEAMCENTER

Data Indexing and Search Configuration

Teamcenter 2312

SIEMENS

게시되지 않은 작업. © 2024 Siemens

이 문서에는 Siemens Industry Software, Inc., 그 계열사(총칭 "Siemens") 또는 라이선스 허가자가 소유한 영업 비밀 또는 기밀 정보가 포함되어 있습니다. 이 문서의 액세스 및 사용은 Siemens와 고객 간의 적용 가능한 계약에 명시된 대로 엄격하게 제한됩니다. 이 문서는 Siemens의 명시적 서면 허가 없이 고객이 복사, 배포 또는 공개할 수 없으며, Siemens가 명시적으로 허가하지 않은 방식으로 일체 사용할 수 없습니다.

이 문서는 정보 및 지침 제공용입니다. Siemens는 사전 통지 없이 이 문서에 포함된 사양 및 기타 정보를 변경할 수 있는 권리를 보유하며, 독자는 모든 경우에 Siemens로 문의하여 변경 사항이 있는지 여부를 확인해야 합니다.

이 문서에 포함된 사실 정보에 대한 진술 또는 기타 확인은 보증으로 간주되지 않으며, Siemens에 어떠한 책임도 야기하지 않습니다.

이 문서가 사용될 제품에 대해 Siemens와 서명한 라이선스 계약이 있는 경우, 이 문서의 사용에는 해당 계약의 라이선스 범위와 소프트웨어 보호 및 보안 조항이 적용됩니다. 해당하는 서명된 라이선스 계약이 없는 경우, 귀하의 사용은 Siemens Universal Customer Agreement의 적용을 받으며, 이와 관련된 내용은 다음에서 확인할 수 있습니다. **https://www.sw.siemens.com/en-US/sw-terms/base/uca/**또한 다음에서 확인할 수 있는 제품별 규정에 의해 보완됩니다. **https://www.sw.siemens.com/en-US/sw-terms/supplements/**.

Siemens는 상업성, 특정 목적을 위한 적합성, 지식 재산권 비위반에 대한 묵시적 보증을 포함하나 이에 국한되지 않는, 본 문서와 관련된 일체의 보증 의무를 부담하지 않습니다. Siemens는 본 문서 또는 그 안에 포함된 정보로 인해, 또는 그와 관련해 발생하는 직접적, 간접적, 부수적, 결과적 또는 징벌적 손해, 데이터 손실 및 이익 손실에 대해, 그 가능성을 사전 고지를 받았더라도 일체의 책임 의무를 부담하지 않습니다.

상표: 본 문서에 사용된 상표, 로고 및 서비스 마크(총칭 "마크")는 Siemens 또는 기타 당사자의 자산입니다. Siemens 또는 마크 소유자 (해당하는 경우)의 사전 서면 동의 없는 마크 사용은 전면 제한됩니다. 본 문서에 사용된 제3자 마크는 Siemens를 제품 공급원으로 표시하기 위한 차원이 아니며, 특정 제3자의 제품이거나 관련 제품임을 나타내기 위한 것입니다. Siemens의 마크 목록은 다음 링크에서 확인할 수 있습니다. **www.plm.automation.siemens.com/global/en/legal/trademarks.html**. 등록 상표 Linux®는 마크 소유자인 Linus Torvalds의 독점 라이선스 사용자인 LMI의 2차 라이선스 계약에 의거해 전 세계에서 사용됩니다.

Siemens Digital Industries Software 정보

Siemens Digital Industries Software는 성장하고 있는 PLM(제품 라이프사이클 관리), MOM(제조 운영 관리), EDA(전자 설계 자동화) 소프트웨어, 하드웨어 및 서비스 분야의 글로벌 리더입니다. Siemens는 10만여 고객사와 협력하며 계획 및 제조 프로세스 디지털화를 주도하고 있습니다. Siemens Digital Industries Software는 가상/물리적, 하드웨어/소프트웨어, 설계 및 제조 환경을 통합하여 산업 부문 간의 단절을 해소합니다. 빠르게 혁신하는 업계 환경에서 디지털화는 더 이상 미래의 영역이 아닙니다. Siemens는 고객이 현재는 물론 미래에 대응할 수 있도록 지원합니다. Where today meets tomorrow. Siemens는 직원, 비즈니스 및 고객의 잠재력을 최대한 실현할 수 있도록 창의성을 북돋우고 새로운 발상에 귀 기울이며 성장에 주력하는 문화를 조성합니다.

Support Center: **support.sw.siemens.com**

문서에 대한 피드백 보내기: **support.sw.siemens.com/doc_feedback_form**

목차

데이터 색인화

© 2024 Siemens

© 2024 Siemens

© 2024 Siemens

© 2024 Siemens

1. 데이터 색인화

색인화 전략화

색인화 프로세스 이해 및 색인화 전략 결정

*색인화*는 사용자가 Active Workspace에서 해당 정보를 효율적으로 검색할 수 있도록 개체, 메타데이터 및 파일 내용을 카탈로그화하는 프로세스입니다.

> 주:
>
> 색인 사용자는 개체 데이터, 데이터 집합 및 텍스트 내용을 색인화할 관련 파일에 대한 읽기 액세스 권한이 있어야 합니다.

색인화와 관련된 **하드웨어 선택**을 검토하고 고려합니다. 정보 유형에 따라 적절한 색인화 방법을 선택하십시오.

파일 내용 색인화 요구사항에 따라 동기화 색인화 프로세스 또는 비동기화 파일 내용 **색인화 프로세스**를 선택합니다. 연관된 파일 내용이 없는 개체는 항상 동기화 프로세스를 사용하여 색인화됩니다. 연결된 모든 데이터 집합에 대해 **동기화 또는 비동기화 색인화 프로세스**를 선택할 수 있습니다.

색인화 데이터에 대한 필수 요건

- 모든 색인화 컴포넌트를 설치합니다.

  배포 센터 또는 Teamcenter Environment Manager(TEM)를 사용하여 Indexing Engine(Solr) 및 Indexer를 설치할 수 있습니다.

  배포 센터를 사용하여 비동기화 파일 내용 색인화 및 을 설치할 수 있습니다.

  색인화 컴포넌트 설치에 대한 자세한 내용은 해당 Teamcenter 설치 안내서를 참조하십시오.

  - Teamcenter Installation Using Deployment Center
  - Teamcenter Installation on Windows Using TEM
  - Teamcenter Installation on Linux Using TEM

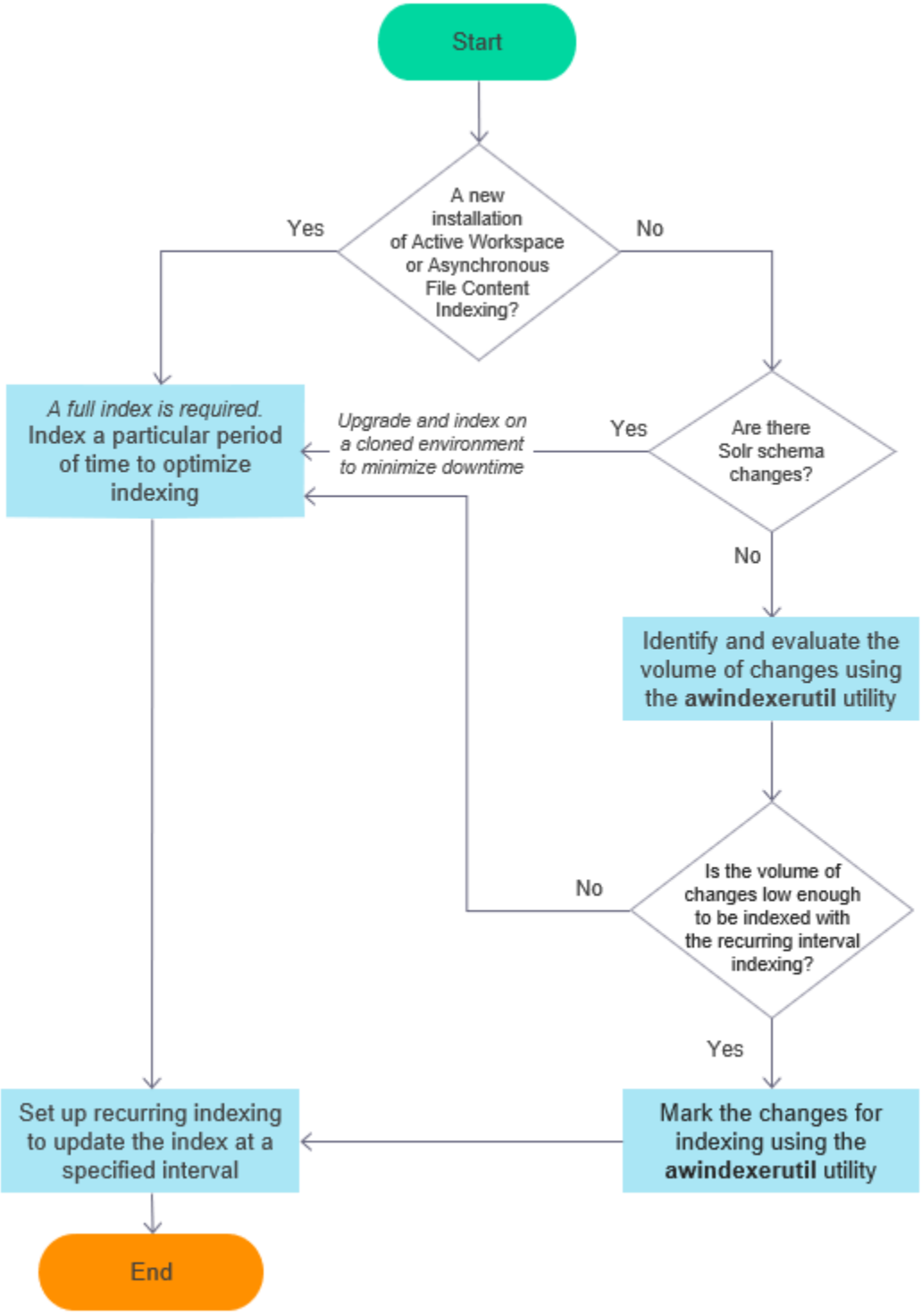
- **색인화 구성.**

  - 비동기화 파일 내용 색인화의 경우 Teamcenter 외부에서 발생한 CAD 파일 색인화를 위한 추가 **구성** 단계가 필요합니다.

© 2024 Siemens

색인화 전략 결정

다음 차트는 색인화 전략을 결정하는 프로세스를 보여줍니다.



© 2024 Siemens

색인화 방법 평가

색인화 요구사항은 하드웨어, 사용 가능한 시간 및 데이터 고려 사항에 따라 다를 수 있습니다.

전체 색인 수행	다음 경우에 전체 색인을 완료합니다. • 새 설치를 완료합니다. 새 설치를 완료할 때와 비동기화 파일 내용 색인화가 설치된 후 첫 번째 색인화의 경우 전체 색인이 필요합니다. • 예를 들어, **필드 유형** 정의가 Solr 스키마 파일에서 변경된 경우 Solr 스키마가 변경됩니다. 최신 버전의 Active Workspace에서 스키마가 변경되어 전체 색인화가 필요할 수 있습니다. • 불륨이 높거나 다양한 변경사항이 있습니다. 업그레이드, 패치 또는 데이터 모델 변경으로 인해 대량 또는 다양한 업데이트가 발생하는 경우 모든 변경사항을 포함하는 전체 색인을 완료하는 것이 좋습니다. 주: 새 설치 이외의 상황에서 전체 색인이 필요한 경우 **복제된 환경에서 업그레이드 및 색인화**를 선택하여 업데이트 기간 동안 다운타임을 최소화할 수 있습니다. 생산 환경의 복제된 병렬 환경에서 색인을 업데이트한 후 업데이트된 색인을 다시 업그레이드된 생산 환경으로 병합합니다. 동기화 플로는 복제 후 수행된 모든 중간 변경사항을 색인화합니다.
최적의 성능을 위해 시간 슬라이스 색인화	특정 기간 또는 시간 슬라이스만 한 번에 색인화하여 프로세스를 최적화합니다. 예를 들어, 이전 년도에 대해서만 데이터를 색인화할 수 있습니다. 나머지 데이터는 시스템이 생산 단계로 전환된 후 색인에 더 많은 시간 단위를 점진적으로 추가하여 색인화할 수 있습니다. 초기 색인 후 색인을 동기화할 경우에는 성능이 사용자에게 미치는 영향을 확인하십시오. Active Workspace에서는 검색 결과에 아이템이 아닌 아이템 리비전을 반환합니다. 반환되는 아이템의 리비전을 식별하기 위해 리비전 규칙을 적용하는 경우 모든 이력 데이터가 색인화되지 않으면 리비전 규칙이 일부 검색에서 정확하지 않을 수 있습니다.
색인을 업데이트하기 위해 반복적 플로 설정	지정된 간격으로 색인화를 동기화하도록 **TcFTSIndexer** 일정을 설정합니다. 이 방법은 이전 색인 이후에 변경된 색인화된 모든 정보를 업데이트합니다.
색인화하기 위해 수정된 데이터 식별	전체 색인화가 필요한지 또는 동기화 플로가 변경사항을 색인화할 수 있는지 확인하기 위해 데이터 모델 업데이트의 변경사항 불륨을 식별하고 평가합니다.

© 2024 Siemens

동기화 또는 비동기화 색인화 프로세스 선택

파일 내용 색인화란?

파일 내용을 색인화하면 사용자가 Active Workspace를 사용하여 포함된 정보를 빠르게 검색할 수 있습니다;. 연관된 데이터 집합이 없는 개체는 항상 동기화 프로세스를 사용하여 색인화됩니다. 연결된 데이터 집합의 경우 설치 중에 동기화 또는 비동기화 색인화 프로세스를 사용할 수 있습니다.

동기화 및 비동기화 색인화의 차이점

동기화 색인화 프로세스에서 **TcFTSIndexer**는 Teamcenter 개체 메타데이터와 파일 내용을 하나의 순차적 프로세스로 함께 색인화합니다. 색인화가 완료된 후 사용자가 모든 내용을 검색할 수 있습니다.

비동기화 색인화 프로세스에서 디스패처[1]는 개체 색인화 타스크와 함께 파일 내용 색인화 타스크를 관리합니다. 파일 내용 색인화가 디스패처 기반 변환기[2]로 오프로드되며, 이는 별도의 프로세스에서 내용을 추출하고 색인화합니다. 비 데이터 집합 개체의 메타데이터는 항상 동기화되어 색인화되며, 사용자는 백그라운드에서 파일 내용 색인화가이 완료되는 동안 메타데이터를 검색할 수 있습니다.

비동기화 파일 내용 색인화를 처음 설치하는 경우 이후에 **기존 데이터의 전체 색인화**를 수행해야 합니다.

사용해야 하는 색인화 프로세스

다음과 같은 경우 *동기화* 색인화 프로세스를 사용합니다.

- **파일을 색인화할 필요가 없는 경우**

  비동기화 파일 내용 색인화 기능은 파일 내용 추출 및 색인화에만 사용됩니다. 파일 내용을 색인화하고 검색할 필요가 없는 경우 이 기능이 필요하지 않습니다.

- **적은 파일만 색인화하려는 경우.**

  생산 환경의 기존 파일 내용 색인화 로드를 관리할 수 있는 경우 비동기화 플로가 필요하지 않습니다.

- **하드웨어 리소스가 제한되어 있는 경우.**

  비동기화 파일 내용 색인화는 Teamcenter 디스패처 프레임워크를 사용하여 파일 내용 색인화 요청을 예약하고 처리합니다. 디스패처 설치가 필요합니다. 최상의 성능을 위해 색인기와 별도의 서버에 디스패처를 설치하십시오.

다음과 같은 경우 *비동기화* 색인화 프로세스를 사용하도록 선택합니다.

- **Teamcenter 개체와 연결된 CAD 파일에서 내용을 색인화하려는 경우.**

1 는 Teamcenter 문서의 *디스패처 소개*를 참조하십시오.
2 Teamcenter 문서의 *디스패처 설치 및 구성*에서 사용자 정의 변환기 추가에 대한 섹션을 참조하십시오.

© 2024 Siemens

비동기화 파일 내용 색인화 프레임워크를 사용하면 사용자의 CAD 파일에서 내용을 추출하고 Teamcenter 개체의 일부로 해당 내용을 색인화할 수 있습니다. 이를 통해 사용자가 이 추출된 정보를 검색할 수 있습니다. 외부 CAD 파일 추출기 구성 및 설치가 필요합니다.

- **메타데이터 정보를 사용하여 먼저 연관된 데이터 집합 없이 개체를 검색하려는 경우.**

  모든 색인화가 완료되면 동기식 색인화 프로세스가 종료됩니다. 파일 내용 색인화가 필요한 개체가 여러 개 있는 경우 사용자는 개체 메타데이터를 검색하기 전에 완료될 때까지 기다려야 합니다. 비동기화 색인화가 설치되어 있는 경우 파일 내용 색인화가 메타데이터 색인화와 별도로 수행됩니다. 메타데이터 색인화가 완료되면 사용자는 파일 내용 색인화 대기 없이 색인화 가능한 모든 개체를 검색할 수 있습니다.

  그러나 비동기화 파일 내용 색인화를 사용하면 데이터 집합의 파일이 메타데이터 색인화와 함께 색인화되며 메타데이터 색인화가 완료된 후에 완료될 수 있습니다. 추출된 파일 내용은 원래 색인화된 Teamcenter 개체에 추가됩니다.

- **파일 내용 색인화로 인해 발생하는 색인화 로드를 분산시키려는 경우.**

  데이터 집합이 있는 Teamcenter 개체가 많은 경우 파일 내용 색인화는 추가 CPU 및 메모리 리소스가 필요한 리소스 집약적인 프로세스가 될 수 있습니다. 이로 인해 전체 색인화 프로세스에 대한 로드가 증가할 수 있습니다. 반대로, 비동기화 파일 내용 색인화는 요청을 추적하기 위해 별도의 독립적인 디스패처 프로세스를 사용합니다. 디스패처 변환기는 파일 내용을 추출하고 수직 또는 수평으로 크기를 조정하여 기본 **TcFTSIndexer** 환경을 오프로드할 수 있습니다. **TcFTSIndexer**는 Teamcenter 데이터를 처리하고 Solr 색인화 엔진으로 가져오는 4계층 SOA 클라이언트입니다.

- **큰 파일이 색인화를 완료할 때까지 기다리지 않으려는 경우.**

  큰 파일에서 내용을 추출해야 하는 경우 추출 결과를 기다리는 동안 동기화 플로의 전체 색인화 파이프라인이 차단될 수 있습니다. 비동기화 파일 내용 색인화는 기본 색인기 외부의 독립적으로 스케일링된 별도의 서버에서 큰 파일을 처리하기 때문에 도움이 될 수 있습니다. 이렇게 하면 사용자는 더 큰 파일의 색인화가 완료되기 전에 먼저 색인화된 개체 메타데이터를 검색할 수 있습니다.

동기화 및 비동기화 색인화 프로세스 비교

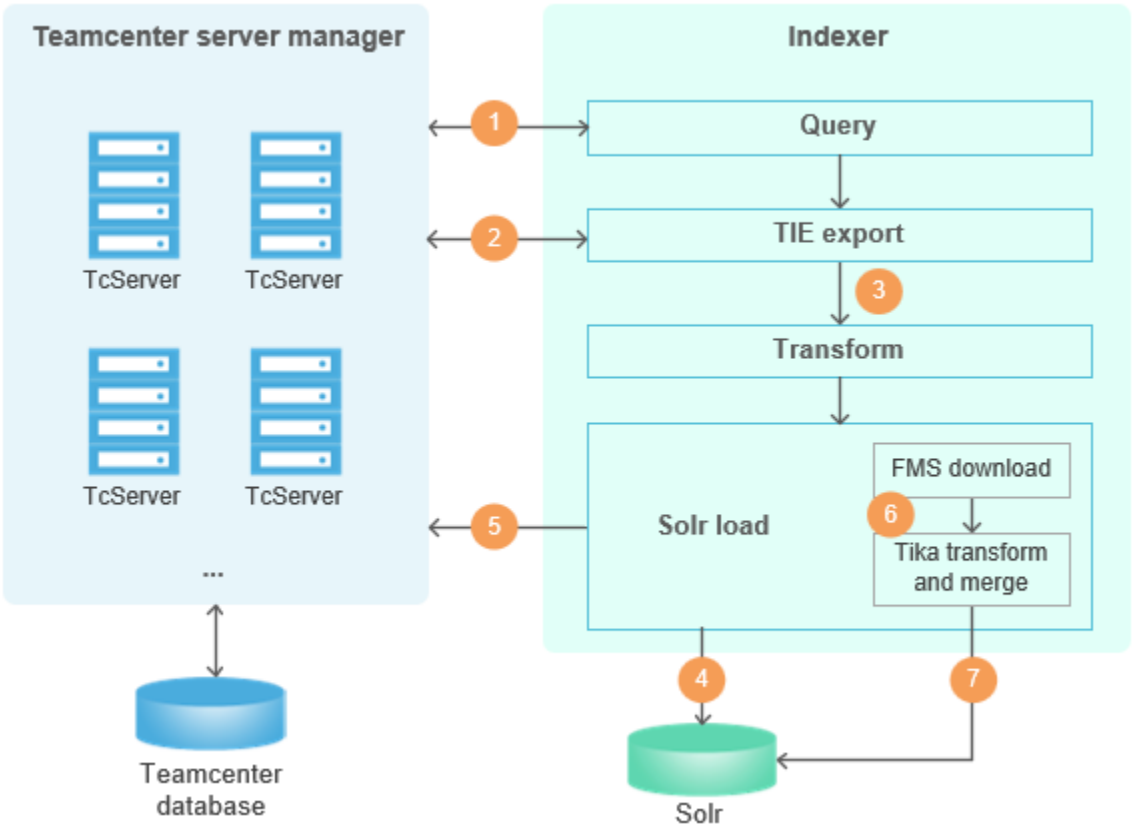
시스템의 요구사항에 따라 파일 내용에 대해 **동기화** 프로세스 또는 **비동기화** 프로세스를 사용할지 여부를 확인합니다.

두 색인화 프로세스 모두 Teamcenter 데이터를 처리하고 Solr 색인화 엔진으로 가져오는 4계층 SOA(서비스 중심 아키텍처) 클라이언트인 **TcFTSIndexer**를 사용합니다. TcFTSIndexer는 순차적으로 실행되는 *조회*, *내보내기*, *변환* 및 *로드* 단계로 구성됩니다.

파일 내용 및 개체에 대한 동기화 색인화 프로세스

다음 그래픽은 Microsoft® 365 파일, PDF 파일 및 텍스트 파일과 같은 Teamcenter 개체 및 관련 표준 파일에 대한 동기화 색인화 프로세스의 절차를 다이어그램을 보여줍니다. 이 예제에서 **TcFTSIndexer**는 PDF 데이터 집합이 있는 아이템 리비전을 색인화합니다.

© 2024 Siemens

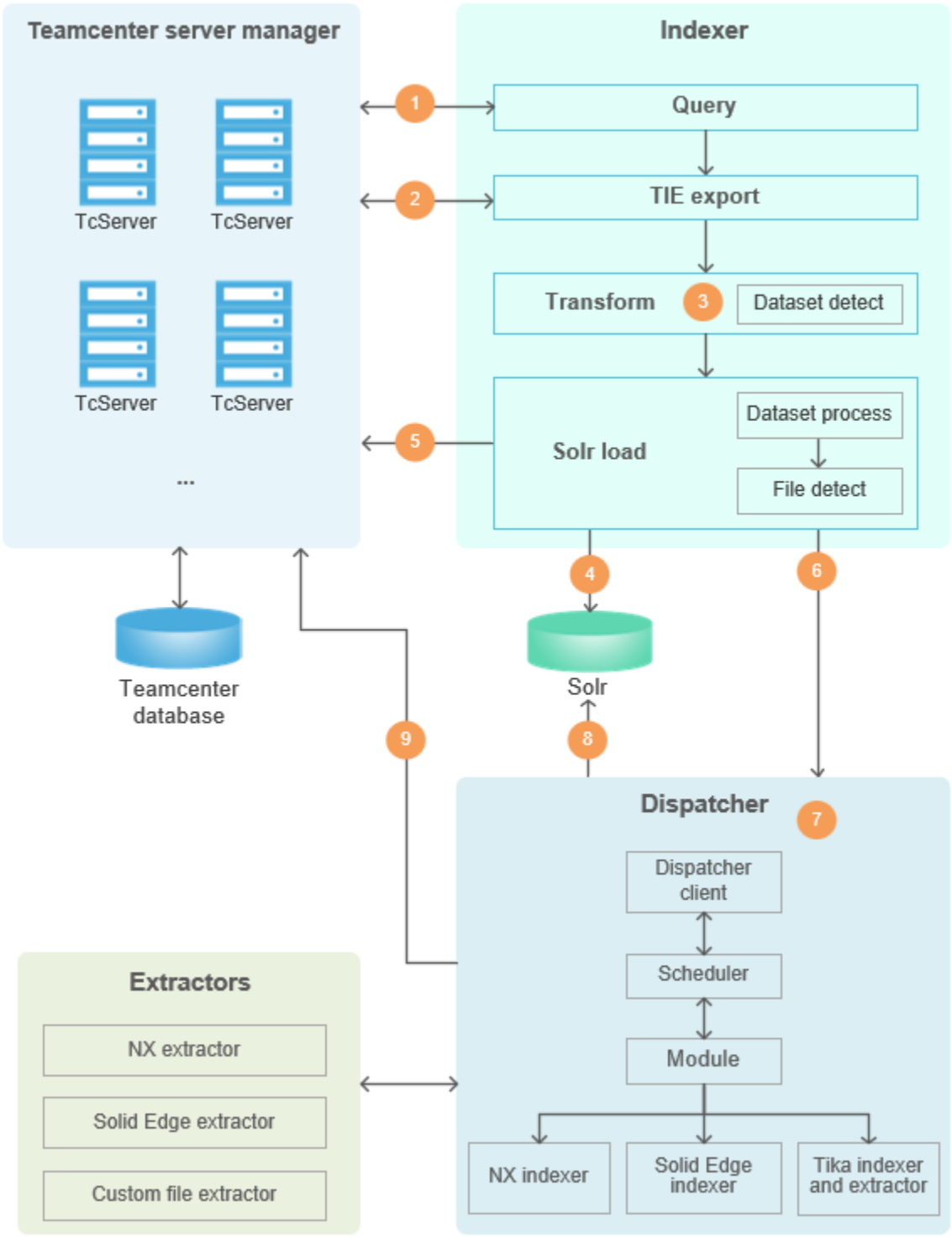


단계	작업	설명
1	조회	색인화 가능한 모든 개체에 대한 UID 리스트를 요청하고 수신합니다. 이 경우 TcServer는 아이템 리비전 및 데이터 집합에 대한 UID를 반환합니다.
2	내보내기	아이템 리비전 및 데이터 집합과 연결된 모든 색인화 가능한 메타데이터를 TC XML 파일로 요청하고 수신합니다.
3	변환	내보낸 TC XML 파일을 Solr XML 파일로 변환합니다.
4	로드	Solr XML 파일을 Solr로 로드합니다. 여기에는 아이템 리비전과 PDF 데이터 집합 모두에 대해 색인화된 메타데이터가 포함됩니다.
5	로드	개체 색인화 상태를 제품 센터에 알립니다.
6	로드	색인화 가능한 데이터 집합에 대한 추가 단계를 수행합니다. 1. FMS(표시되지 않음)를 사용하여 데이터 집합에 첨부된 파일을 다운로드합니다. 2. 다운로드한 파일에서 색인화 가능한 내용을 추출합니다.
7	로드	*변환* 단계의 모든 메타데이터 내용을 추출된 PDF 내용과 병합하여 Solr에 로드하는 단일 Solr XML 문서를 생성합니다.

© 2024 Siemens

파일 내용에 대한 비동기화 색인화 프로세스 및 개체에 대한 동기화 색인화 프로세스

다음 프로세스 다이어그램은 파일 내용에 대한 비동기화 색인화 프로세스의 단계를 보여줍니다. Teamcenter 개체 메타데이터는 동기화 색인화 플로를 사용하여 색인화됩니다. 파일 내용을 비동기화로 색인화하면 NX 또는 Solid Edge CAD 파일과 같은 비표준 파일 유형도 색인화할 수 있습니다.



© 2024 Siemens

단계	작업	설명
1	조회	색인화 가능한 모든 개체에 대한 UID 리스트를 요청하고 수신합니다. 이 경우 TcServer는 아이템 리비전 및 데이터 집합에 대한 UID를 반환합니다.
2	내보내기	아이템 리비전 및 데이터 집합과 연결된 모든 색인화 가능한 메타데이터를 TC XML 파일로 요청하고 수신합니다.
3	변환	내보낸 TC XML 파일을 Solr XML 파일로 변환합니다. 또한 이 작업은 데이터 집합 개체를 식별합니다.
4	로드	Solr XML 파일을 Solr로 로드합니다. 여기에는 아이템 리비전과 PDF 데이터 집합 모두에 대해 색인화된 메타데이터가 포함됩니다.
5	로드	메타데이터 색인화 프로세스의 성공 또는 실패를 Teamcenter에 알립니다.
6	로드	메타데이터 색인화 프로세스가 성공한 경우 *변환* 단계에서 식별된 데이터 집합을 데이터 집합 유형 및 메타데이터 속성별로 그룹화합니다. 또한 이 작업은 DispatcherRequest 개체를 사용하여 디스패처에 데이터 집합 그룹을 제출합니다.
7	디스패처[3]	데이터 집합과 연결된 파일 내용을 추출하고 색인화합니다. 1. 디스패처 클라이언트는 **DispatcherRequest**[4] 개체를 읽고 스케줄러에 요청을 제출합니다. 2. 스케줄러는 디스패처 모듈을 사용하여 파일 내용 색인기에 요청을 라우팅합니다. 3. 각 파일 내용 색인기에서 데이터 집합과 연결된 파일을 다운로드하고 추출기를 사용하여 내용을 추출합니다. 데이터 집합의 유형에 따라 다른 색인기가 호출됩니다. 예를 들어, Tika 색인화는 Microsoft Office 파일을 처리하는 반면 NX 색인화는 NX 파트 파일을 처리합니다.
8	디스패처	Solr에서 이미 색인화된 메타데이터에 추출된 내용을 추가합니다.
9	디스패처	데이터 집합 색인화 상태를 제품 센터에 알립니다.

TcFTSIndexer에 대한 하드웨어 고려

최적의 색인화 및 검색 성능을 달성하기 위해 별도의 서버에 **TcFTSIndexer** 컴포넌트를 배포할 수 있습니다. Siemens Digital Industries Software은 특정 용도로 설계된 서버에 컴포넌트를 설치할 것을 강력히 권장합니다. TcServers는 단일 스레드이기 때문에 서버 머신에 필요한 CPU 코어 수를 결정하는 것이 중요합니다. 하드웨어 세부 사항 및 권장 사항은 *배포 참조 아키텍처*를 참조하십시오.

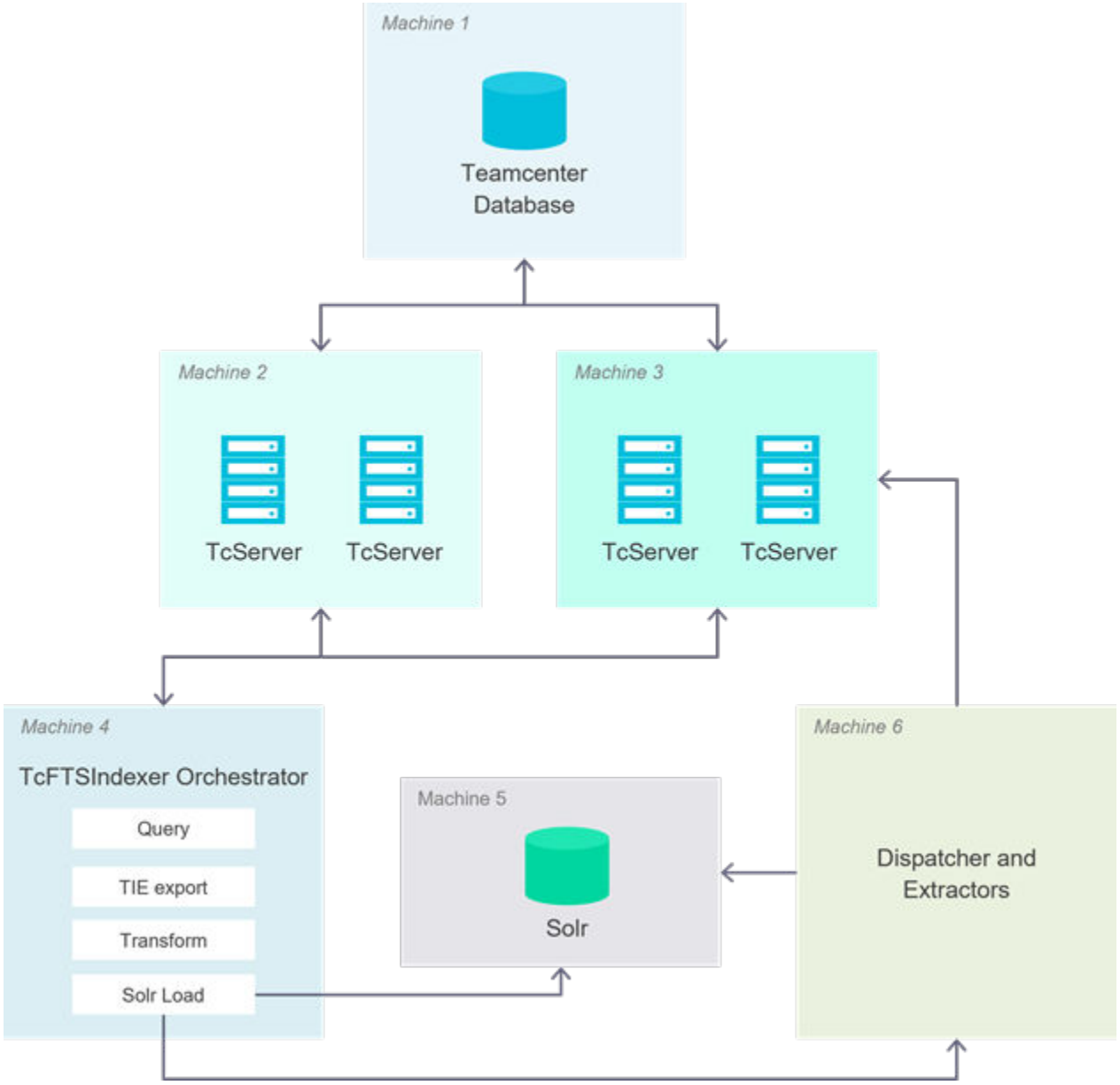
3 Teamcenter 문서에서 디스패처 소개를 참조하십시오.
4 Teamcenter 설명서의 *디스패처 설치 및 구성*에서 디스패처 요청에 대한 섹션을 참조하십시오.

© 2024 Siemens

이 컴포넌트	설명	색인화 고려 사항도 있습니다.
Teamcenter 데이터베이스	짧은 시간에서 많은 수의 요청을 처리합니다.	데이터베이스는 하드웨어의 모든 영역에서 집약적입니다. 이러한 요청은 CPU 사용률이 높을 수 있으며, SQL 호출은 테이블에 읽기 및 쓰기 작업이 필요하며 이는 I/O 및 메모리 사용률이 높습니다. 데이터베이스 구성 및 하드웨어 요구 사항을 우선적으로 신중하게 고려해야 합니다.
Teamcenter 서버 관리자	색인기 조회, TIE 내보내기 및 Solr 로드 작업을 실행합니다.	이러한 오퍼레이션은 다중 **tcserver** 인스턴스에서 실행되며 데이터베이스에 연결됩니다. 색인화 성능은 동시에 완료할 수 있는 병렬 스레드 수에 좌우되기도 합니다. 스레드 수는 **tc.maxConnections** 속성에 의해 제어되지만, 이는 서버 관리자의 윔업 서버 수와 Teamcenter 데이터베이스가 지원할 수 있는 스레드 수를 통해 조정되어야 합니다. **Tc.maxConnections**를 최적화하면 스레스 수가 너무 많거나 적어 성능에 영향을 미치는 문제를 방지할 수 있습니다.
TcFTSIndexer Orchestrator	독립 실행형 모드에서 쿼리, TIE 내보내기, 변환 및 Solr 로드 오퍼레이션을 실행하는 단일 Java 프로세스를 실행합니다.	고급 병렬화, 변환은 CPU 및 메모리 사용률이 더 높습니다. **TcFTSIndexer** Orchestrator는 개체 데이터 색인화에 대해 Teamcenter 서버 관리자에 연결합니다. **TcFTSIndexer** 컴포넌트는 다른 시스템 또는 단순한 환경의 단일 시스템에서 독립적으로 설정할 수 있습니다.
Solr	Active Workspace에 전역 검색을 위해 색인화된 Teamcenter 데이터를 저장합니다.	쿼리 호출 동안 CPU 사용률이 높고 대용량 데이터 집합 파일을 색인화하는 동안 메모리 사용률이 높을 수 있으며, 많은 문서를 읽고 쓰므로 I/O 사용률도 높습니다. **비동기화 파일 내용 색인기** 응용 프로그램에는 색인화 프로세스 중에 추가 저장소가 필요합니다. 전체 색인 크기는 현재 크기의 두 배로 증가할 수 있습니다.
디스패처	**비동기화 파일 내용 색인기** 응용 프로그램을 사용할 때 색인기를 오프로드하는 데 도움이 됩니다.	디스패처 변환기는 색인기 리소스를 확보하기 위해 기본 색인기 외부에서 큰 파일을 처리하는 데 도움이 되도록 수직 또는 수평으로 확장될 수 있습니다.

© 2024 Siemens

색인화 시스템 레이아웃 예제



데이터 모델 변경의 영향 식별

데이터 모델 업데이트 후 수정된 데이터를 색인화해야 합니다. 변경사항을 처리하기 위해 델타 색인을 완료할 수 있습니다. 그러나 전체 색인과 델타 색인은 데이터 모델 업데이트로 인한 영향 받는 유형 및 속성 변경 비율이 높기 때문에 처리 시간이 유사할 수 있습니다. 영향을 계산하려면 **Teamcenter 및 Solr 스키마를 병합**한 다음 색인화 인프라에서 지원할 수 있는 개체 수와 변경사항 수를 비교하여 데이터 모델 변경 불룸을 평가합니다.

개체 데이터 유형 및 속성 색인화 변경 내용만 있으면 이러한 변경 내용을 평가하여 범위를 결정합니다.

© 2024 Siemens

유형 변경 새 유형 또는 이전에 색인화를 위해 구성되지 않은 기존 유형.

속성 변경 이전에 색인화된 유형의 경우, 새 속성 또는 이전에 색인화를 위해 구성되지 않은 기존 속성(참조 및 복합 속성 포함).

Teamcenter 스키마와 Solr 스키마를 병합한 후 **awindexerutil -delta -dryrun**을 사용하여 마지막 버전과 현재 버전 간의 변경사항을 식별합니다. 이 유틸리티는 Business Modeler IDE에서 수행한 개체에 대한 사용자 정의 스키마 변경을 포함하여 데이터 모델 개체에 대한 유형 및 속성 변경을 식별합니다. 결과 보고서를 사용하여 변경 내용의 볼륨을 평가하고 반복 간격 색인화를 사용하여 **변경 내용을 색인화할 수 있는지** 또는 **전체 색인**이 필요한지 여부를 확인할 수 있습니다.

복제된 환경을 사용하여 색인화 가속화

Active Workspace 업그레이드 또는 패치의 경우 복제 및 병합 방식을 사용하여 다시 색인화 프로세스를 가속화하고 생산 환경의 다운타임을 줄일 수 있습니다. 현재 생산 환경의 복사본을 만들고, 복제를 업그레이드하고, 다시 색인화를 수행한 다음, 다시 색인화된 파일을 업데이트된 생산 환경으로 병합합니다. 그런 다음 복제 시점과 병합 시점 사이에 변경된 내용을 색인화합니다.

복제된 환경에서 색인화하기 전에 다음 아이템을 고려하십시오.

- 생산 환경에는 기존 색인화된 개체 데이터가 있어야 합니다.
- **runTcFTSIndexer** 유틸리티를 사용하는 것을 잘 알고 있어야 합니다. 복제 및 색인화 작업을 수행하는 데 사용하는 **runTcFTSIndexer**에 대한 여러 가지 **admin** 흐름 작업이 있습니다.
- 복제된 환경을 구성하는 파일 집합을 복사하려면 적절한 디스크 공간이 필요합니다.
- 복제된 환경에 대한 사용자 정의를 수행하지 마십시오. 복제된 환경이 업그레이드된 생산 환경에 병합될 때까지 기다리십시오.

다중 사이트 공동 작업을 사용하여 다른 사이트의 데이터 색인화

사용자는 다중 사이트 공동 작업을 사용하여 다른 사이트에서 제공되는 데이터를 검색할 수 있습니다. 개체는 원본 사이트에서 게시된 후 여러 사이트로 복제되어 사용자가 공유 게시 개체를 검색할 수 있도록 해줍니다. 사용자는 **사이트 검색**을 사용하여 **로컬** 또는 **원격** 결과를 표시하여 결과를 필터링할 수 있습니다. **로컬** 결과에는 공유 사이트의 데이터가 포함됩니다. **원격** 필터는 사용자가 선택할 수 있는 특정 **원격 사이트**를 나열하는 관련 하위 필터를 갖습니다.

개체 디렉터리 서비스(ODS) 게시 개체는 색인화용으로 구성되어야 합니다. 게시된 개체(또는 게시 레코드)는 마스터의 업데이트 여부를 추적하고 공유 데이터를 다른 사이트에서 업데이트해야 하는지의 여부를 결정합니다.

다중 사이트 공동 작업 정보는 Teamcenter 도움말에 있습니다.

© 2024 Siemens

다중 사이트 색인화 활성화

- 다중 사이트 공동 작업을 수행하려면 **TcFTSIndexer**를 구성합니다.

  플랫폼용 Teamcenter 소프트웨어 키트에서 *soa_client.zip*를 열고 *\soa_client\java\libs*를 찾습니다. **TcSoaMultisiteStrong_***version***.jar** 및 **TcSoaMultisiteTypes_***version***.jar** 파일을 색인기 서버의 다음 디렉터리로 복사:

  TC_ROOT

  *TC_ROOT***TcFtsIndexer\lib**

- **ODS_site** 환경설정을 기본 ODS 사이트에 지정된 Teamcenter 사이트 이름으로 설정합니다. **ODS_searchable_sites** 환경설정을 게시된 개체를 검색할 수 있는 ODS 사이트로 설정합니다.

- *TC_ROOT \TcFTSIndexer\conf*에 있는 *TcFtsIndexer_multisite.properties* 파일을 사용하여 다중 사이트 색인화를 위한 **색인 속성**을 구성합니다. 예를 들어, 파일에서 개체 색인화의 시작 시간 및 종료 시간이 설정됩니다.

- **POM_owning_site** 속성을 설정하여 게시된 레코드의 색인화를 활성화합니다.

- **runTcFTSIndexer** 유틸리티 다중 사이트 플로 타스크를 사용하는 **게시 레코드를 색인화합니다**.

색인화 구성

색인화 타스크 구성

데이터의 전체 색인화 또는 델타 다시 색인화가 필요한지 결정하고 **개체 데이터 색인화에 대한 표준 프로세스**를 이해해야 합니다.

사이트에 대해 수행해야 할 수 있는 **검색 구성 타스크**를 평가합니다.

또한 구조적 내용 색인화를 구성할 수도 있습니다.

모든 색인화 컴포넌트를 설치한 후 데이터를 색인화하기 전에 색인화를 구성해야 합니다. 색인화 구성은 사용자가 데이터, 개체 및 파일 내용을 효율적으로 검색할 수 있도록 합니다.

색인화 컴포넌트가 설치되어 실행 중인지 확인합니다.

- 검색 색인 기능이 설치되어 있는지 확인합니다:

  - Teamcenter Installation Using Deployment Center

  - Teamcenter Installation on Windows Using TEM

© 2024 Siemens

  - Teamcenter Installation on Linux Using TEM

- **Solr 색인화 엔진이 실행 중인지 확인합니다.**
- **TcFTSIndexer 연결을 테스트합니다.**
- **Teamcenter 및 Solr 스키마를 병합합니다.**

색인화 상세정보 구성:

- **TcFTSIndexer 인스턴스 최적화.**
- **색인 데이터 및 필터 정의.**
- **검색 색인 속성 구성.**
- **TcFTSIndexer를 실행하는 사용자 변경.**
- **파일 내용에서 조각 표시.**
- **지역화된 디스플레이 값 구성.**
- **오른쪽에서 왼쪽으로 언어 색인화.**
- **보고 마이크로서비스 구성.**
- **비동기화 파일 내용 색인화 구성.**

색인화 컴포넌트 설치 개요

색인화 엔진 및 색인기는 Active Workspace 클라이언트에 대한 전역 검색 기능을 제공합니다.

배포 센터 또는 Teamcenter Environment Manager(TEM)를 사용하여 Indexing Engine(Solr) 및 Indexer를 설치할 수 있습니다.

배포 센터를 사용하여 비동기화 파일 내용 색인화 및 을 설치할 수 있습니다.

색인화 컴포넌트 설치에 대한 자세한 내용은 해당 Teamcenter 설치 안내서를 참조하십시오.

- Teamcenter Installation Using Deployment Center
- Teamcenter Installation on Windows Using TEM
- Teamcenter Installation on Linux Using TEM

© 2024 Siemens

색인화 엔진 (Solr)	Solr 엔터프라이즈 검색 플랫폼을 설치합니다. 색인화는 Active Workspace에 전역 검색을 위해 색인화된 Teamcenter 데이터를 저장합니다. 선택된 제품 데이터는 Apache의 오픈 소스 검색 플랫폼인 Solr에서 색인화됩니다. 마스터 제품 데이터는 Solr에 저장되지 않습니다. 항상 Teamcenter에서 로드됩니다.
색인기	Solr에 병합하기 위한 Teamcenter 데이터를 내보내는 4계층 SOA 클라이언트를 설치합니다. 색인기는 전체 색인화 프로세스를 관리합니다. **TcFTSIndexer**는 개체 데이터의 초기 색인화를 관리합니다. 그런 다음 개체 데이터 또는 구조 데이터 색인에 대한 후속 업데이트를 위해 동기화가 주기적으로 실행되도록 예약할 수 있습니다. TcFTSIndexer는 외부 및 Teamcenter 개체를 Solr로 색인화합니다. 서버 관리자에 연결하여 Solr로 색인화할 Teamcenter 데이터를 조회 및 추출합니다.
비동기화 파일 내용 색인화	개체 메타데이터에서 비동기적으로 파일 내용을 색인화할 수 있는 **선택적 특징형상**을 설치합니다. 디스패처는 파일 내용 요청을 관리하는 데 사용됩니다.

Solr 시작

Solr은 **색인화 엔진** 기능이 설치된 시스템에 설치됩니다. *INDEXING-ENGINE-ROOT***solr-***version**에 설치됩니다.

색인화 엔진을 서비스로 설치하도록 선택한 경우 설치 중에 **Active Workspace 색인화 서비스**가 생성되고 시작됩니다. 해당 **시작 유형**은 **자동**입니다.

Windows 서비스로 색인화 엔진을 설치하지 않았거나 Linux 시스템에 있을 경우 수동으로 시작해야 합니다. 다음에서 **RunSolr.bat** 또는 **runSolr.sh** 실행:

INDEXING-ENGINE-ROOT\solr-*version*

Solr이 실행 중인지 확인하려면 웹 브라우저를 열고 Solr 페이지에 액세스합니다.

http://*host*:*port*/solr/admin

*host*는 Solr가 설치되는 시스템입니다.

*port*는 Solr에서 사용하는 포트입니다. 기본값은 **8983**입니다.

색인화 엔진을 설치할 때 정의한 Solr 관리자 이름과 암호로 로그인합니다.

TcFTSIndexer 연결 테스트

runTcFTSIndexer 유틸리티를 실행하는 Teamcenter 사용자는 데이터베이스에 로그인할 수 있어야 하며 모든 색인화 시스템이 실행되어야 합니다.

© 2024 Siemens

1. 유틸리티를 실행하는 기본 사용자는 *TC_ROOT/TcFTSIndexer/conf/TcFtsIndexer.properties* 파일의 **Tc.user** 설정에 정의되어 있습니다.

   또한, Security Services 보호 환경에서 실행하는 경우 이 사용자는 Security Services이 LDAP 서버에서 인증할 수 있는 유효한 사용자여야 합니다.

   a. 여러 TCCS Security Services 응용 프로그램 ID를 사용하는 경우, 올바르게 구성되어 있는지 확인합니다.

      Teamcenter Environment Manager(TEM)의 **클라이언트 통신 시스템의 환경 설정** 패널을 사용하여 여러 응용 프로그램 ID를 구성할 수 있습니다.

      TCCS 환경을 생성하는 데 필요한 정보를 입력합니다.

값	설명
이름	TCCS 환경의 이름을 지정합니다. 구성이 완료되면 이 이름이 TCCS 로그온 다이얼로그에 표시됩니다.
URI	TCCS 환경의 URL을 지정합니다. 이는 웹 계층 배포에 대한 끝점 URI입니다(예: **http://***host***:***port***/tc**). 네트워크에서 IPV6(128비트) 또는 IPV4(32비트) 주소를 사용하는 경우, 사용해야 하는 IP 주소를 DNS(도메인 이름 시스템)가 결정할 수 있도록 URI에서 호스트 이름을 사용하고 리터럴 주소를 사용하지 마십시오.
태그	TCCS 환경에 대한 문자열 ID를 지정합니다. Rich Client를 설치할 때, 필터와 일치하는 해당 환경에 대한 Rich Client에 표시된 환경의 리스트를 필터링하도록 **클라이언트 태그 필터** 값을 선택적으로 제공할 수 있습니다. 예를 들어, **클라이언트 태그 필터** 값이 **9***인 경우, **9**로 시작하는 **태그** 값을 가진 모든 TCCS 환경은 클라이언트 호스트에서 사용할 수 있습니다. **태그** 값이 **10**으로 시작하는 환경은 사용할 수 없습니다.
SSO 응용 프로그램 ID	TCCS에서 사용할 Security Services 응용 프로그램의 ID를 지정합니다.
SSO 로그인 URL	Security Services에서 사용할 TCCS 응용 프로그램의 URL을 지정합니다. 애플릿 없는 모드에서 Security Services을(를) 사용할 경우, URL의 끝에 **/tccs**를 포함합니다. 예를 들면 다음과 같습니다. **http://***host***:***port***/***app-name***/tccs**

   b. *TC_ROOT\TcFTSIndexer\conf\TcFtsIndexer.properties* 파일의 **Tc.user** 설정을 통해 정의한 사용자가 LDAP 서버 및 Teamcenter 데이터베이스에서 유효한 사용자인지 확인합니다. 필요한 경우

© 2024 Siemens

두 위치에서 사용자를 생성하거나 기존의 유효한 활성 사용자를 선택해 **runTcFTSIndexer** 유틸리티를 실행합니다.

새 사용자를 생성하는 경우 콘솔 내 환경 변수를 암호 값으로 설정해 암호화된 암호 파일을 생성합니다. 예:

set mytcenv=*password*

그런 다음 **encryptPass.bat/sh** 유틸리티를 **-tc** 인수와 함께 실행해 실행된 환경 변수 이름을 지정합니다. 예:

encryptPass -tc *mytcenv*

그런 다음 **TcFTSIndexer** 사용자가 사용하는 LDAP 암호에 대해 이 유틸리티를 실행합니다. **암호화 Tpass .bat/sh** 유틸리티는 *TC_ROOT\TcFTSIndexer\bin* 디렉터리에 있습니다.

암호화된 암호 파일을 생성한 후 암호 변수를 제거합니다.

2. 다음이 실행 중인지 확인합니다.
   - Teamcenter 데이터베이스
   - Solr
   - Teamcenter 웹 계층 애플리케이션을 호스트하는 웹 애플리케이션 서버
   - Active Workspace 웹 애플리케이션을 호스트하는 웹 애플리케이션 서버
   - 서버 관리자
3. **색인기** 기능이 설치된 컴퓨터에서 명령 프롬프트를 엽니다.
4. **TcFTSIndexer**의 디렉터리(예: *TC_ROOT\TcFTSIndexer\bin*)로 이동합니다.
5. **TcFTSIndexer**에 대한 연결성을 테스트하려면 다음 명령을 실행합니다.

   ```
   runTcFTSIndexer -task=objdata:test
   ```

6. 오류가 없는지 확인합니다.

Teamcenter 및 Solr 스키마 병합

데이터를 색인화하려면 먼저 Solr 스키마를 Teamcenter 스키마와 병합해야 합니다. 색인화 엔진 설치 중에 스키마 파일(*TC_DATA\ftsi\solr_schema_files*)을 병합하지 않은 경우, 스키마를 수동으로 병합해야 합니다.

© 2024 Siemens

색인화에 영향을 미치는 Business Modeler IDE를 변경한 후에도 Solr 스키마를 병합하고 업데이트해야 합니다. 예를 들어, 새 비즈니스 개체 또는 새 색인화 가능한 속성을 추가한 후 Solr스키마를 병합하고 업데이트해야 합니다.

이 단계는 Solr을 **독립형 모드**로 실행하는지 또는 **SolrCloud 구성**에서 실행하는지에 따라 다릅니다.

독립형 모드에서 병합 및 업데이트

1. 실행 중인 경우 Solr 서비스를 중지합니다.

   사용자 환경에서 Solr은 **Active Workspace 색인화 서비스**로 실행되거나 **runSolr** 명령을 사용하여 실행되었을 수 있습니다.

2. 기업 서버의 *TC_SOLR_SCHEMA.xml*TC_DATA*TC_ACE_SOLR_SCHEMA.xml 디렉터리에서* 및 *\ftsi\solr_schema_files* 파일을 찾습니다. 스키마를 수동 병합하려면 Solr이 설치된 시스템에서 이들 파일을 사용할 수 있어야 합니다.

   *TC_ACE_SOLR_SCHEMA.xml*은 **활성 내용 구조** 기능이 설치된 경우에만 표시됩니다.

   Solr은 **색인화 엔진** 기능과 동일한 시스템에 설치됩니다. 기업 서버가 아닌 시스템에 이 기능을 설치한 경우, 이 파일을 Solr 호스트의 임시 위치로 복사해야 합니다.

3. 명령 프롬프트를 열고 *SOLR_HOME* 디렉터리로 변경합니다.

   기업 서버 구성에 색인화 기능을 설치한 경우, *SOLR_HOME* 디렉터리는 *TC_ROOT*\solr-\solr-*version* 디렉터리입니다.

   별도의 시스템에 색인화 기능을 설치한 경우 *SOLR_HOME* 디렉터리는 *indexing-engine-install-dir*\solr-\solr-*입니다*.

4. Solr 스키마를 업데이트하려면 이 명령을 실행합니다.

   TcSchemaToSolrSchemaTransform.bat *LOCAL-DIR*

   *local-dir*은 *TC_SOLR_SCHEMA.xml* 및 *TC_ACE_SOLR_SCHEMA.xml* 파일을 포함하는 로컬 디렉터리입니다.

   이 명령이 "파일을 찾을 수 없음" 오류를 표시하면 *JAVA_HOME* 폴더를 찾을 수 없습니다. 이 *.bat* 파일에는 *JAVA_HOME*에 대한 하드 코드된 값이 있습니다. 사용자 환경과 이 값이 일치하는지 확인하거나, 사용자 시스템에서 *JAVA_HOME* 환경 변수를 설정한 경우 이 행을 제거합니다.

   이 명령은 업데이트된 스키마에 이러한 사용자 정의가 포함되지 않은 경우 사용자 정의가 포함된 Solr 스키마의 백업을 생성합니다.

5. 모든 사용자 정의에 대한 백업 스키마를 검토하고 업데이트된 스키마를 동일하게 변경합니다.

© 2024 Siemens

6. **Active Workspace 색인화 서비스**를 재시작하거나 시작 명령을 실행하여 Solr을 다시 시작합니다.

   ```
   SOLR_HOME\runSolr.bat
   ```

SolrCloud 모드에서 병합 및 업데이트

SolrCloud 스키마 업데이트의 필수항목

- 이미 사용 가능하거나 실행 중인 SolrCloud 구성
- 식별된 *Leader1_SOLR_HOME* 설치

병합 및 업데이트 절차

다음과 같이 *schema.xml* 또는 *solrconfig.xml*과 같은 구성 파일을 업데이트합니다.

1. Zookeeper에서 컬렉션에 대한 Configset를 다운로드합니다.

   Configset에는 각 컬렉션에 사용되는 모든 구성 파일이 포함되어 있습니다. 독립 실행형 모드에서는 구성 파일이 Solr의 컬렉션 내에 저장됩니다. SolrCloud에서는 ZooKeeper에 저장됩니다.

   a. 관리자로 명령 프롬프트를 열 수 있습니다.

   b. *Leader1_SOLR_HOME\bin* 디렉터리로 이동합니다.

   c. 각 컬렉션에 대해 다음 명령을 실행합니다. 이렇게 하면 *conf* 디렉터리가 지정된 임시 경로로 다운로드됩니다.

      solr zk downconfig -n *collection_name* **-d** *Temp_path\collection_name* **-z** *hostname:port*

      여기서:

-n	컬렉션의 이름을 지정합니다.
-d	임시 디렉터리의 경로를 지정합니다.
-z	ZooKeeper 시스템 및 포트를 지정합니다.

      예:

      ```
      solr zk downconfig -n collection1 -d C:\SolrTest\collection1 -z
      server1:2181
      ```

2. 파일을 Teamcenter 스키마와 병합합니다.

   Leader1_SOLR_HOME 디렉터리로 이동하고 각 컬렉션에 대해 다음 명령을 실행하여 Teamcenter 스키마를 Solr 스키마와 병합합니다.

© 2024 Siemens

TcSchemaToSolrSchemaTransform.bat *TC_DATA\ftsi\solr_schema_files Temp_path\collection_name\conf*

예:

```
TcSchemaToSolrSchemaTransform.bat
C:\apps\TC\TC_DATA\ftsi\solr_schema_files
C:\SolrTest\collection1\conf
```

이 명령은 업데이트된 스키마에 이러한 사용자 정의가 포함되지 않은 경우 사용자 정의가 포함된 Solr 스키마의 백업을 생성합니다.

3. 모든 사용자 정의에 대한 백업 스키마를 검토하고 업데이트된 스키마를 동일하게 변경합니다.

4. Configset를 ZooKeeper로 다시 업로드합니다.

   Leader1_SOLR_HOME\bin 디렉터리로 이동하고 각 컬렉션에 대해 다음 명령을 실행합니다. 이렇게 하면 지정된 경로에서 *conf* 디렉터리가 업로드됩니다.

   solr zk upconfig -n *collection_name* **-d** *Temp_path\collection_name* **-z** *hostname:port*

   예:

   ```
   solr zk upconfig -n collection1 -d C:\SolrTest\collection1 -z
   server1:2181
   ```

5. 모든 업로드가 완료되면 SolrCloud 구성 내의 모든 노드를 다시 시작하여 스키마 변경 사항이 적용되도록 합니다.

TcFTSIndexer 인스턴스 최적화

최적화를 시작하기 전에

TcFTSIndexer 환경의 최적화를 시작하기 전에 다음을 수행하십시오.

1. **하드웨어 고려 사항**을 검토합니다.

2. **TcFTSIndexer 연결**을 테스트합니다.

최적화 이해

색인화 최적화는 반복 프로세스입니다:

- 소량의 데이터 집합을 색인화하여 전체 데이터 집합을 색인화하려면 어느 정도의 시간이 걸릴지 파악합니다.

© 2024 Siemens

- 환경에서 병렬 스레드가 더 많이 또는 더 적게 필요한지 여부를 결정하는 데 이 정보를 사용하여 CPU, 메모리 및 I/O의 소비를 해석합니다.

이 정보를 평가하면 추가 시스템 구성뿐 아니라 **TcFTSIndexer** 설정을 재구성하는 데 도움이 됩니다.

TcFTSIndexer Orchestrator는 쿼리, TIE 내보내기, 변환 및 순차적인 패턴으로 실행되는 독립 실행형 프로세스로 로드하기를 지원합니다. 이러한 프로세스는 독립 실행형 모드에서의 구성 수준이 한 가지이며, 이것이 최대 연결 설정(**tc.maxConnections** 속성)입니다. 이 설정은 정해진 시간에 열리는 Teamcenter 연결의 최대 개수를 지정합니다. 이 수는 풀 관리자에서 사용 가능한 원 **tcserver** 인스턴스의 수를 초과하지 않아야 합니다. 최소값은 **2**이며 기본값은 **3**입니다.

tc.maxConnections 값을 배포 센터 또는 Teamcenter Environment Manager(TEM)에서 변경할 수 있습니다.

시간 슬라이스 및 일괄 작업 크기 최적화

여러 시스템과 각 시스템의 컴포넌트를 정의한 후 최적의 시간 단위 및 배치 크기를 결정합니다. 그 다음, 연결 개수를 조정합니다.

1. 색인화할 개체의 작은 리스트를 식별합니다. 색인화는 시간 단위에 기반하므로, 총 개체 수의 약 5%가 포함된 시작 시간과 종료 시간을 식별해야 합니다. 대략적 시간을 얻기 위해 Teamcenter Rich Client에서 색인화 가능한 개체에 대한 쿼리를 실행할 수 있습니다.

   다음 파일에 시작 및 종료 시간을 설정합니다.

   *TC_ROOT***TcFTSIndexer\conf***TcFTSIndexer_objdata.properties*

2. 색인화 컴포넌트가 실행 중이고 **TcFTSIndexer** Orchestrator가 **runTcFTSIndexer** 명령을 사용하여 색인화 컴포넌트에 연결할 수 있는지 테스트합니다.

   ```
   runTcFTSIndexer -task=objdata:test
   ```

3. Solr, 풀 관리자, **TcFtsIndexer** Orchestrator 및 데이터베이스 등 설치된 색인화 컴포넌트가 있는 각 컴퓨터에서 Windows Task Manager를 시작합니다.

4. 다음 명령을 실행하여 개체의 5%로 구성된 집합을 색인화합니다.

   ```
   runTcFTSIndexer -task=objdata:index
   ```

   - 색인화 컴포넌트가 설치된 시스템에서 Task Manager를 사용하여 메모리 사용 및 CPU 프로세스를 모니터링합니다. 추가 **tcserver** 연결을 위한 공간이 있는지 평가할 수 있습니다.

   - 데이터베이스와 시스템 중 하나가 오버로드되지 않도록 실행되고 있는 데이터베이스와 시스템 모두를 모니터링합니다.

© 2024 Siemens

- **TcFTSIndexer** 실행 종료 시 생성된 보고서에서 색인화하는 데 걸린 전체 시간을 보고합니다. 보고서는 콘솔에서 사용할 수 있으며 *TC_ROOT\TcFTSIndexer\logs*의 **TcFTSIndexer.log** 및 **TcFTSIndexer_objdata.log**에 저장됩니다.

5. 초기 실행 이후, 모든 시스템이 해당 리소스를 90% 이하로 사용했다면 원 **tcservers**의 최대 수를 초과하지 않는 한 **tc.maxConnections** 값을 늘립니다. 그렇다면 연결 수를 늘리기 전에 풀에 사용할 수 있는 원 Teamcenter 서버 수를 늘립니다.

   > 팁:
   >
   > 활성 색인화 실행 중에 **tc.maxConnections** 설정을 동적으로 수정할 수 있습니다.
   >
   > a. 새 Teamcenter 명령 윈도우를 열고 *TC_ROOT\TcFTSIndexer\bin* 디렉터리로 이동합니다.
   >
   > b. 다음과 같은 **runTcFTSIndexer** 유틸리티 명령을 실행합니다.
   >
   > ```
   > runTcFTSIndexer -maxConnections=connections
   > ```
   >
   > *connections*는 원하는 연결 수이지만, 사용 가능한 **tcservers**의 수를 초과하지 않아야 합니다.

6. 100%에서 항상 최적화되는 것은 아니지만 리소스 소비가 90% 이상에 도달하여 **tc.maxConnections** 설정이 최적화될 때까지 이전 두 단계를 반복합니다.

7. 최적의 연결 설정을 발견한 후 Teamcenter Environment Manager(TEM)을 사용하여 이 설정을 색인화 시스템에서 모든 색인화 세션에 대해 영구적으로 만듭니다.

   a. Teamcenter Environment Manager(TEM)을 열고 **구성요소 유지보수** 패널이 나올 때까지 **다음**을 클릭합니다.

   b. **색인기 설정 업데이트**를 선택한 후 **색인기 설정** 패널이 나올 때까지 **다음**을 클릭합니다.

   c. **최대 Teamcenter 연결**을 찾고 값을 변경합니다.

8. **tc.maxConnections** 설정을 최적화한 후에 **exportbatchsizestep.baseBatchSize** 속성을 최적화합니다.

   > 주:
   >
   > 이 단계는 색인화할 개체가 100만 개 미만인 소규모 환경에 대해서는 선택 사항이지만, 색인화할 개체가 100만 개 이상인 중대형 환경에서는 사용이 권장됩니다.

   기본 **exportbatchsizestep.baseBatchSize** 값은 1000입니다.

   - 성능을 개선하기 위해 **exportbatchsizestep.baseBatchSize**의 값을 증가시킬 수 있는데, 이는 스레드당 Teamcenter 개체 수를 증가시킵니다.

© 2024 Siemens

- 내부 테스트에서는 색인 크기에 따라 2000-5000으로 설정된 값이 최적화된 결과를 산출했습니다.
- 너무 큰 값으로 증가시킬 경우 성능에 악영향을 줄 수 있습니다.

9. **TcFTSIndexer**가 최적화되어 적절한 크기가 된 후에는 데이터의 전체 색인화를 시작할 수 있습니다.

색인 데이터 및 필터 정의

기본적으로 **ItemRevision** 비즈니스 개체의 자식은 해당 비즈니스 개체의 특정 속성 뿐만 아니라 색인화를 위해서도 표시됩니다. Business Modeler IDE에서 색인화할 비즈니스 개체 및 속성을 정의하는 사용자 정의 템플릿을 생성할 수 있습니다. 또한 전역 상수를 사용하여 검색 필터 동작을 정의할 수도 있습니다. 상수 관련 작업에 대한 자세한 내용은 Teamcenter 문서의 비즈니스 개체 상수 소개를 참조하십시오.

검색 및 필터링에는 다음 속성이 필요합니다. 이 속성은 기본적으로 색인화되며 제거할 수 없습니다.

- **POM_application_object.last_mod_date**
- **POM_application_object.creation_date**
- **WorkspaceObject.object_type**

검색 및 필터링을 위해 영구 속성을 색인화할 수 있습니다. 검색 및 필터링을 위해 다음 속성 유형을 색인화할 수 없습니다.

- **Runtime** 속성.
- **런타임** 속성 또는 통과 경로에서 일시적으로 표시된 속성을 가진 **복합** 속성입니다.
- 고유한 저장소 클래스가 없는 폼 데이터 파일을 참조하는 **복합** 속성입니다.
- **LongString** 속성 유형이 있는 속성.
- **TypedReference** 특성 유형이 있는 속성은 다음과 같은 속성 상수가 설정된 경우만 색인화될 수 있습니다.
  - **Awp0SearchRefTypesNames**
  - **Awp0SearchPropFromRefType**

> 팁:
>
> 변경 및 받은 편지함에 대해 개별 필터링을 설정할 수 있습니다.
>
> 속성을 필터로 설정하는 경우 개체 ID와 같이 많은 수의 고유 값이 있는 속성을 사용하지 않도록 하십시오. 사용자가 해당 필터를 사용하려고 하면 작업 결과로 찾은 결과가 없습니다 오류 메시지가

© 2024 Siemens

표시될 수 있습니다. 예를 들어 검색에서 별표(*) 와일드카드를 사용하고 필터가 ID 속성을 지정하는 경우 반환된 아이템 수가 많으면 사용자가 ID를 필터링하려고 할 때 오류가 발생합니다. 필터를 구성할 때는 고유한 값이 적은 속성을 선택하십시오.

1. Active Workspace(**aws2**) 템플릿을 Business Modeler IDE 프로젝트로 가져옵니다.

   템플릿 가져오기에 대한 자세한 내용은 Teamcenter Business Modeler IDE 도움말의 템플릿 소개를 참조하십시오.

2. Business Modeler IDE에서 색인화할 비즈니스 개체를 엽니다.

3. 색인화할 비즈니스 개체를 정의하려면 **Awp0SearchIsIndexed** 비즈니스 개체 상수를 사용합니다.

   계층에서 색인 값이 상속됩니다. 예를 들어 **ItemRevision** 비즈니스 개체를 색인화하도록 표시할 경우, 그 아래에 있는 모든 아이템 리비전(파트 리비전, 문서 리비전 등)도 색인화하도록 자동으로 표시됩니다. 그러나 상속된 값을 재정의하도록 속성을 **false**로 설정해 하위 단계 개체를 색인화하지 않도록 선택할 수 있습니다.

   동일한 **Awp0SearchIsIndexed** 비즈니스 개체 상수를 사용하여 색인화할 데이터 집합 비즈니스 개체를 정의해야 합니다. 예를 들어, 색인화할 *.jpeg* 파일을 정의하려면 Business Modeler IDE에서 **JPEG** 비즈니스 개체를 열고 **Awp0SearchIsIndexed** 상수를 **true**로 설정합니다.

4. 색인화할 속성을 정의하려면 **Awp0SearchIsIndexed** 속성 상수를 사용합니다.

   계층에서 색인 값이 상속됩니다. 예를 들어 **ItemRevision** 비즈니스 개체의 **object_type**을 색인화하도록 표시할 경우, 그 아래에 있는 모든 아이템 리비전(파트 리비전, 문서 리비전 등)도 색인화하도록 자동으로 표시됩니다.

5. (선택 사항) **Awp0IndexableFileTypes** 전역 상수에 추가 파일 확장 유형을 입력하거나 **AW_Indexable_File_Extensions** 환경설정에 전역 상수 및 추가 값의 모든 파일 확장자를 입력합니다.

   전역 상수 및 환경설정은 색인화할 파일 유형을 정의합니다. 검색 색인화 중에 텍스트 내용 추출을 위해 파일 확장자를 지정합니다. 유효한 값은 파일 확장자 이름(예: *.txt*, *.doc*, *.zip*, *.jpeg*, *.png*)입니다.

   이 환경설정에 대해 값이 지정되지 않은 경우 **Awp0IndexableFileTypes** 전역 상수 아래에 나열된 파일 확장자가 사용됩니다.

   애니메이션된 *.gif* 파일에서는 텍스트 내용 추출을 위해 순서의 첫 번째 이미지만 읽습니다.

   텍스트 내용 추출에는 *.heic* 및 *.avif* 파일 형식이 지원되지 않습니다.

6. 적절한 전역 상수, 비즈니스 개체 상수 또는 속성 상수를 사용해 검색 필터를 정의합니다.

© 2024 Siemens

예를 들어, 검색 결과 필터 리스트에 표시할 속성을 지정하려면 **Awp0SearchCanFilter** 속성 상수를 사용합니다. 필터 리스트에 속성이 표시되는 순서를 설정하려면 **Awp0SearchFilterPriority** 속성 상수를 사용합니다. 계층에서 색인 값이 상속됩니다.

필터가 사용자 인터페이스에 제대로 표시되려면 **Awp0SearchCanFilter** 및 **Awp0SearchFilterPriority** 속성 상수를 소스 비즈니스 개체의 속성에 대해 설정해야 합니다.

> 팁:
>
> 하나의 속성에 대해 **Awp0SearchCanFilter** 속성 상수를 true로 설정하면 반환되는 모든 개체에 해당 속성이 있는 경우 해당 속성만 검색 결과 필터링 리스트에 표시됩니다. 반환된 개체의 모든 속성이 표시된 경우 필터 리스트는 유용하게 사용하기에는 너무 길어질 수 있습니다. 소유한 사용자, 마지막 수정 날짜, 릴리스 상태 및 유사한 속성은 모든 개체에 공통된 속성이므로 항상 검색 결과 필터 리스트에 표시됩니다.

속성이 **테이블** 형식 속성인 경우 참조된 **TableRow** 유형의 모든 유효한 속성을 필터로 사용할 수 있습니다. 이러한 모든 테이블 행 속성은 필터 우선 순위가 동일하기 때문에 함께 나타나며 필터 패널에 수직으로 나열됩니다.

테이블 속성의 필터 카테고리 이름에는 구분 기호(:)와 테이블 행 속성 표시 이름이 추가된 테이블 속성 표시 이름이 포함되어 있습니다. 예를 들어, **내용**은 테이블 속성 표시 이름이고 **비용**, **처리 날짜** 및 **값**은 테이블 행 속성 표시 이름입니다.

7. 기본적으로 Solr는 필터 리스트의 각 속성 아래에 최대 100개의 값을 표시합니다. 제한 사항을 제거하고 모든 개체를 나열할 수 있습니다. *TC_ROOT***solr-***version***server\solr\collection1\conf\solrconfig.xml** 파일을 열고, **<requestHandler name="/tcfts" class="solr.SearchHandler">**를 검색합니다. 다음을 **<lst name="defaults">** 섹션에 추가합니다.

```
<requestHandler name="/tcfts" class="solr.SearchHandler">
   <!-- Request handler for Teamcenter search -->
   <lst name="defaults"
      <str name="echoParams">explicit</str>
      <int name="rows">10</int>
      <str name="df">TC_0Y0_FTS</str>
      <str name="q.op">OR</str>
      <str name="facet.limit">-1</str>
   </lst>
```

특정 속성에 대한 제한도 제거할 수 있습니다. 예를 들어, 모든 개체 유형을 나열하려면 다음을 **<lst name="defaults">** 섹션에 추가합니다.

```
<str name=”f.TC_0Y0_WorkspaceObject_0Y0_object_type.facet.limit”>-1</str
<str name=”facet.sort”>count</str>
```

8. **사용자 정의 템플릿을 업데이트한 후 색인기 실행**.

© 2024 Siemens

대체 ID 색인화 및 검색

대체 ID는 서로 다른 투시에서 동일한 파트에 대한 정보(파트 번호 및 속성)를 저장합니다. 이를 사용해 여러 유형의 사용자가 개체를 만든 사용자의 규칙에 따르는 대신 자체 규칙에 따라 아이템을 표시할 수 있습니다. 항목 비즈니스 개체 또는 자식만 대체 ID를 사용합니다.

ItemRevision 비즈니스 개체에 있는 **altid_list** 속성은 런타임 속성이며, 런타임 및 관계 속성은 색인에 지원되지 않으므로 기본적으로 대체 ID를 검색에 색인화할 수 없습니다. 속성, 복합, 테이블 및 참조 속성만 색인화됩니다.

대체 ID를 색인화 및 검색하려면 비즈니스 개체 유형에서 대체 ID를 추가하는 복합 속성을 정의해야 합니다. 복합 속성은 관계별 참조를 사용하여 소스 속성에 도달합니다.

예를 들어, **IdentifierRev** 개체를 찾는 것으로 가정합니다. 기본적으로 **Identifier** 및 **IdentifierRev** 비즈니스 개체에는 **Identifier** 저장소 클래스가 있습니다. 따라서 다음과 같이 복합 속성을 정의해야 합니다.

```
<TcCompoundPropertyRule destTypeName="ItemRevision" destPropertyName="dev3AltID"
   sourceTypeName="Identifier" sourcePropertyName="idfr_id" isReadOnly="false"
   pathToSource="REFBY(altid_of,Identifier).REF(suppl_context,Identifier).idfr_id"
   description=""/>
```

이것은 뒤 방향 참조를 사용하여 **altid_of** 관계를 통해 **IdentifierRev** 개체를 찾은 다음 **suppl_context** 참조 속성을 통해 **Identifier** 개체의 모든 참조를 찾습니다. **idfr_id** 속성은 이러한 **식별자** 개체에서 가져옵니다.

이제 **MyIdentifier** 및 **MyIdentifierRev**와 같은 사용자 정의 **Identifier** 비즈니스 개체가 있다고 가정합니다.

Identifier
  |- **IdentifierRev**
  |- **MyIdentifier**
    |- **MyIdentifierRev**

다음과 같이 복합 속성을 정의합니다.

```
<TcCompoundPropertyRule destTypeName="ItemRevision" destPropertyName="dev3AltID"
   sourceTypeName="MyIdentifier" sourcePropertyName="idfr_id" isReadOnly="false"

pathToSource="REFBY(altid_of,MyIdentifierRev).REF(suppl_context,MyIdentifier).idfr_id"
   description=""/>
```

검색 중 대체 ID로 검색하면 해당 키워드와 일치하는 **ItemRevision** 개체가 반환됩니다. 복합 속성에 대한 속성별 검색도 지원됩니다.

© 2024 Siemens

> 주:
>
> Business Modeler IDE에는 제한이 있습니다. Business Modeler IDE로부터 이러한 유형의 복합 속성을 추가할 수 없습니다. 이들을 템플릿에 수동으로 추가하고 Business Modeler IDE에서 해당 템플릿을 다시 로드하여 유효성을 검사해야 합니다.

AW_FTSIndexer_skip_modifications_via_relations 환경설정은 **TcFTSIndexer**가 동기화 플로 동안 관계 조회를 실행하여 수정 사항을 찾을 수 있는지 여부를 제어합니다. 기본값은 **true**이며, 이는 조회를 건너뜁니다. 값을 **false**로 설정하여 아이템에서 대체 ID를 색인화합니다.

검색 색인 속성 구성

TcFTSIndexer 검색 속성은 *TC_ROOT***TcFTSIndexer\conf**에 있는 *TcFTSIndexer_objdata.properties* 파일에 저장됩니다.

비동기화 파일 내용 색인화 플로를 사용하는 경우 *Dispatcher_Root***Module\Translators\TcFTSIndexer\conf**에 있는 *TcFTSIndexer_objdata.properties* 파일과 *TcFTSIndexer_filecontent.properties* 파일 모두에서 속성을 구성합니다.

대부분의 검색 속성은 **TcFTSIndexer** 컴포넌트 설치 중 정의됩니다. 미리 정의한 속성을 편집하거나 파일에 나열된 추가 속성을 정의할 수 있습니다.

데이터 집합 색인화

데이터 집합 유형이 색인화로 표시되는 경우 독립 데이터 집합 및 해당 파일 내용이 색인화됩니다. 환경설정 **AWS_FullTextSearch_Index_Dataset_File_Content**를 **on**으로 설정하여 데이터 집합 파일 내용을 색인화합니다.

데이터 집합에 대해 부모 비즈니스 개체 반환 여부를 구성할 수 있습니다.

텍스트 내용 추출을 위해 이미지 파일을 색인화하려면 *TC_ROOT***TcFTSIndexer\conf**에 있는 *TcFtsTikaConfig.xml* 파일에서 **extractInlineImages** 매개변수 값을 **true**로 설정합니다.

텍스트 내용 추출에 지원되는 최대 이미지 크기는 32,767픽셀 x 32,767픽셀입니다.

> 주:
>
> 이미지 해상도가 낮으면 텍스트 내용이 부정확하게 추출될 수 있습니다. 텍스트 내용 추출을 위해 여러 이미지를 한 번에 색인화하기 전에 테스트를 수행하는 것이 좋습니다.

TcFTSIndexer를 실행하는 사용자 변경

검색 색인기(**TcFTSIndexer**)를 실행하는 사용자는 색인화 엔진 설치 중에 설정됩니다. 사용자를 변경하려는 경우 다음 단계를 수행합니다. 색인 사용자는 개체 데이터, 데이터 집합 및 텍스트 내용을 색인화할 관련 파일에 대한 읽기 액세스 권한이 있어야 합니다.

© 2024 Siemens

1. *TC_ROOT***TcFTSIndexer\conf\TcFtsIndexer.properties** 파일로 이동하여 엽니다.

2. **Tc.user** 설정에서 사용자를 데이터베이스 권한이 있는 승인된 Teamcenter 사용자로 변경합니다.

3. 콘솔에서 암호 값에 대한 환경 변수를 설정합니다.

   ```
   set mytcenv=password
   ```

4. *TC_ROOT***TcFTSIndexer\bin** 디렉터리의 **encryptPass.bat/sh** 유틸리티를 실행하여 이 사용자에 대한 암호화된 암호 파일을 생성합니다.

   -tc 인수를 사용하여 **encryptPass.bat/sh** 유틸리티를 실행하고 이전 단계에서 생성한 환경 변수를 지정합니다. 예를 들어,

   ```
   encryptPass -tc mytcenv
   ```

5. 암호화된 암호 파일을 생성한 후에 환경 변수 값을 제거합니다.

   ```
   set mytcenv=
   ```

파일 내용에서 조각 표시

이미지를 포함하여 아이템에 첨부된 파일 내용 내에서 검색 용어의 위치를 표시할 수 있습니다. 첨부된 데이터 집합 파일의 일치하는 내용이 있는 전역 검색 결과는 텍스트의 조각을 표시합니다. 반환된 조각은 개체 아래에 구문으로 표시됩니다. 스니펫을 표시하면 특히 결과가 이름이나 설명과 정확하게 일치하지 않는 경우 사용자가 결과에서 검색 용어와 일치하는 항목을 찾는 데 도움이 될 수 있습니다. 조각은 검색어를 표시할 때 유사한 단어를 일치시키는 형태소 분석 규칙을 따릅니다.

1. 색인화 엔진(Solr)에서는 Business Modeler IDE에서 전역 상수를 설정하여 Solr 스키마에 대한 데이터 집합 필드가 저장되었는지 확인해야 합니다. 전역 상수 **Awp0StoreDatasetContent**를 **true**로 설정합니다.

2. Teamcenter 명령 프롬프트에서 다음을 실행합니다.

   ```
   bmide_modeltool -u=user -p=password -g=dba -tool=all -mode=upgrade
      -target_dir="%TC_DATA%"
   ```

3. **Teamcenter와 Solr 스키마를 병합합니다**.

   Solr을 다시 시작하고 데이터 집합을 포함하여 데이터를 다시 색인화합니다.

4. **AWC_Search_Enable_Snippets** 환경설정은 강조 표시된 구문이 전역 검색에 표시되는지 여부를 제어하고 데이터 집합이 색인화되는지 여부도 확인합니다. 사이트의 환경 설정이 활성화되어 있지만, 사용자는 이를 **false**로 설정하여 결과 리스트에 보이는 아이템 수를 늘릴 수 있습니다.

5. **AWC_Search_Trim_Snippets_Size** 환경설정을 설정하여 조각에 표시되는 주변 내용의 양을 제어할 수 있습니다. 파일 내용에서 일치하는 검색 조건의 양쪽에 표시할 단어 수를 지정합니다. 조각이

© 2024 Siemens

400자보다 많은 경우에만 환경설정이 적용됩니다. 예를 들어, 반환된 조각 크기가 450자인 경우 이 환경설정의 값이 적용됩니다. 기본값은 일치하는 용어의 왼쪽과 오른쪽에 있는 10개의 단어입니다.

조각이 활성화되어 있지만 첨부된 데이터 집합 내에서 검색을 사용할 수 없는 경우에는 환경설정 제어 조각이 무시됩니다.

지역화된 디스플레이 값 구성

디스플레이 이름을 지역화한 후 색인화하면 지역화된 속성 값을 사용자가 사용할 수 있습니다.

- 디스플레이 이름을 사용하여 이러한 참조 속성을 검색할 수 있습니다.
  - **object_type**에 대해 **유형**을 사용합니다.
  - **release_status**에 대해 **릴리스 상태**를 사용합니다.
  - **project_list**에 대해 **프로젝트**를 사용합니다.

색인화하려는 문자열 속성이 지역화 가능으로 표시되어 있는지 확인합니다.

1. Business Modeler IDE에서 **Awp0IndexLocalizedValues** 전역 상수를 **true**로 변경합니다.
2. Teamcenter 명령 프롬프트에서 다음을 실행합니다.

   ```
   bmide_modeltool -u=user -p=password -g=dba -tool=all -mode=upgrade
      -target_dir="%TC_DATA%"
   ```

3. **Teamcenter와 Solr 스키마를 병합합니다**.
4. *TC_ROOT*/**TcFTSIndexer/conf/***TcFTSIndexer_objdata.properties*로 이동합니다. 파일을 열고 **objdatasaxtransformstep.supportedLocalesForIndexing** 속성에 색인화하려는 로케일 리스트가 포함되어 있는지 확인합니다.
5. 색인 데이터를 지우려면 **runTcFTSIndexer -task=objdata:clear**를 실행합니다. 옵션을 입력하라는 메시지가 표시될 때 옵션 **1** 또는 옵션 **4**를 지정합니다.

   옵션 **1**을 선택하면 색인기 캐시가 지워집니다.

   옵션 **4**를 선택하면 캐시, Solr 색인 및 책임 테이블에서 개체 데이터 색인화 레코드가 지워집니다.

6. 데이터를 다시 색인화합니다.

© 2024 Siemens

예제:

TcFTSIndexer_objdata.properties 파일에서 지원되는 언어를 설정합니다.

```
objdatasaxtransformstep.supportedLocalesForIndexing=
   en_US,de_DE,it_IT,ja_JP,cs_CZ,es_ES,fr_FR,ko_KR,pl_PL,pt_BR,
   ru_RU,zh_CN,zh_TWen_US
```

값이 비어 있으면 서버의 마스터 로캘만 색인화됩니다.

색인기 서비스 시작

색인기가 서비스로 설치된 경우 **objdata:sync** 플로와 **objdata: build_suggester** 플로를 Windows 또는 Linux의 서비스로 실행할 수 있습니다. 이러한 각 동기화 플로에 대한 간격은 설치 중에 설정됩니다.

Windows에서 색인기 서비스 시작

1. Windows 서비스 리스트에서 새로 설치한 다음 서비스 중 하나를 찾습니다.
   - Teamcenter 전역 검색 색인화 서비스
   - Teamcenter 제안 빌더 서비스
2. **서비스** 윈도우에서 서비스를 마우스 오른쪽 버튼으로 클릭하고 **시작**을 선택합니다.
3. 원하는 서비스가 모두 시작될 때까지 이전 프로세스를 반복합니다.

주:

다음 두 가지 방법 중 하나를 사용하여 서비스의 간격을 변경할 수 있습니다.

- 배포 센터의 색인기 컴포넌트 섹션에서 **동기화 간격** 필드를 업데이트하고 배포 스크립트를 생성하고 영향 받는 시스템에 소프트웨어를 배포합니다.
- Registry Editor에서 **간격** 매개변수를 업데이트하고 서비스를 다시 시작합니다.

Linux에서 색인기 서비스 시작

rc scripts 또는 **systemd scripts**를 사용하여 Linux에서 색인기 서비스를 실행할 수 있습니다.

다음 테이블에서는 색인화 서비스 및 사용된 스크립트 유형에 따라 서비스를 시작하는 방법에 대해 설명하고 있습니다.

© 2024 Siemens

스크립트 유형	색인화 서비스	서비스 시작 방법	간격 변경 방법
rc scripts	Teamcenter 전역 검색 색인화 서비스	**./rc.indexerObjDataSync start**	색인기 데몬 실행을 중지하고 **rc.indexerObjdataSync**에서 간격을 업데이트하고 서비스를 시작합니다.
	Teamcenter 제안 빌더 서비스	**./rc.indexerSuggBuilder start**	색인기 데몬 실행을 중지하고 **rc.indexerSuggBuilder**에서 간격을 업데이트하고 서비스를 시작합니다.
systemd scripts	Teamcenter 전역 검색 색인화 서비스	**TcFTSIndexer/bin**에서 *indexerObjdata_systemd.service.readme* 파일을 참조합니다.	**indexerObjdata_systemd.service**에서 간격 매개변수를 업데이트하고 서비스를 시작합니다.
	Teamcenter 제안 빌더 서비스	**TcFTSIndexer/bin**에서 *indexerSuggBuilder_systemd.service.readme* 파일을 참조합니다.	**indexerSuggBuilder_systemd.service**에서 간격 매개변수를 업데이트하고 서비스를 시작합니다.

오른쪽에서 왼쪽으로 언어 색인화

Solr에서 지원하는 아랍어 및 히브리어 같은 RTL 언어의 경우, 파일 내용에서 오른쪽에서 왼쪽으로 텍스트를 색인화할 수 있습니다. 색인기 및 Solr 구성은 지원되는 RTL 언어가 포함된 PDF와 같은 데이터 집합을 색인화하는 기능을 제공합니다.

색인화를 위한 RTL 언어 지원 설정

RTL 언어 파일은 색인화 지원을 제공합니다. 언어 프로파일 파일이 없으면 Solr 설치에서 얻을 수 있습니다.

1. Solr의 **\contrib\langid\lib** 디렉터리에서 Solr 언어 프로파일 JAR 파일 *langdetect*를 찾습니다. 다른 디렉터리로 복사하고 JAR 파일을 추출합니다.

   추출된 **프로파일** 디렉터리에서 RTL 언어의 프로파일 파일을 찾습니다(예: 히브리어의 경우 *he*).

2. 다음 디렉터리에 (자체 또는 Solr의) 언어 프로파일 파일을 추가합니다.

   TC_ROOT/**TcFTSIndexer/conf/languageProfiles**

3. *SOLR_HOME***server\solr\collection1\conf***schema.xml**을 엽니다.

4. RTL 언어에 대한 필드 유형을 생성합니다. 이 예제에서는 히브리어로 설정합니다.

```
<fieldType class="solr.TextField" name="tc_text_he"
positionIncrementGap="0">
   <!-- Hebrew (he) -->
    <analyzer>
```


© 2024 Siemens

```
        <tokenizer class="solr.ICUTokenizerFactory"/>
      </analyzer>
    </fieldType>
```

5. RTL 언어에 대한 필드를 생성합니다.

```
<field indexed="true" multiValued="true" name="TC_DS_file_content_he"
   required="false" stored="false" type="tc_text_he"/>
```

6. Solr를 다시 시작합니다.

7. RTL 언어 세트를 지정하려면 **AWC_Additional_Supported_Languages_Dataset_Indexing** 환경설정을 생성합니다. 언어 값 리스트로 설정합니다. 언어 값은 언어 프로필 파일 이름과 일치해야 합니다(예: 히브리어의 경우 *he*).

8. RTL 언어로 내용을 검색할 수 있도록 데이터 집합을 다시 색인화해야 합니다.

보고 마이크로서비스 구성

보고서 기능을 설치하면 **내 대시보드**에서 검색 조건에 대해 직접 색인화 엔진(Solr)을 검색하는 보고서를 생성할 수 있습니다. 이러한 보고서는 필터 등록 정보를 반환하지만 개체 자체를 검색하지 않으므로 더 빠르게 실행됩니다. 이 유형의 보고서에는 GraphQL 마이크로서비스가 필요합니다.

Active Workspace를 설치하거나 업데이트할 때 Windows 또는 Linux에 마이크로서비스를 설치하고 색인화 엔진(Solr)과 통신하도록 GraphQL 마이크로서비스를 구성해야 합니다. GraphQL 마이크로서비스는 Active Workspace를 기본적으로 사용할 수 있습니다.

TEM 사용	Active Workspace를 설치하거나 업그레이드하는 경우 Active Workspace URL 및 **색인화 엔진** 사용자 이름 및 암호를 제공하여 **GraphQL 마이크로서비스** 기능을 구성합니다. Active Workspace에 대한 **구성요소 유지보수** 업데이트를 수행하는 경우 **Teamcenter GraphQL 서비스**를 찾고 **설정 업데이트**를 선택합니다. 색인화 엔진 사용자 이름과 암호를 제공해야 합니다.
배포 센터 사용	Active Workspace를 설치하거나 업그레이드하는 경우 필요한 **Teamcenter GraphQL 서비스**를 포함하는 기본 **마이크로서비스 노드** 컴포넌트를 구성합니다. 또한 **색인화 엔진** 컴포넌트 사용자 이름과 암호를 제공해야 합니다.

올바른 **색인화 엔진 사용자 이름 및 암호**가 업데이트되었을 수 있으므로 이를 확인합니다.

구성을 완료한 후 Active Workspace의 **내 대시보드**에서 사용 가능한 **내 수정된 개체** 또는 **내 그룹 릴리스된 개체** 보고서를 실행하여 연결을 테스트할 수 있습니다.

© 2024 Siemens

비동기화 파일 내용 색인화 구성

비동기화 파일 내용 색인화 구성 개요

전제 조건

Teamcenter 데이터 집합에 대한 비동기화 파일 내용 색인화를 실행하거나 CAD 파일에 대한 비동기화 파일 내용 색인화를 구성하려면 먼저 다음 사항을 확인해야 합니다.

- **비동기화 파일 내용 색인기** 응용 프로그램이 설치됩니다.

  디스패처 모듈 변환기 설정 섹션에서 원하는 추출기 위치가 지정되어 있어야 합니다.

- 개체 데이터 색인화가 올바르게 구성되어 있습니다.

- 디스패처 스케줄러, 디스패처 모듈 및 디스패처 클라이언트 응용 프로그램이 실행되고 있습니다.

비동기화 파일 내용 색인화 구성

비동기화 파일 내용 색인화 응용 프로그램을 설치하면 개체 메타데이터와 별도로 데이터 집합을 색인화할 수 있습니다. 이 응용 프로그램을 구성하면 다음을 수행할 수 있습니다.

- NX 또는 Solid Edge CAD 필드 내용을 색인화하고 검색합니다.

- NX 파트 속성에 대한 색인화 및 검색 필터를 구성합니다.

- 조건에 따라 CAD 파일이 색인화되는 상황 제한합니다.

- 지정된 위치에서 ITAR 제어 데이터 집합을 처리합니다.

- 색인된 파일 내용에서 패싯 비활성화.

- 테스트 비동기화 파일 내용 색인화 설정.

비동기화 파일 내용 색인화 비활성화

TC_ROOT/TcFTSIndexer/conf/TcFTSIndexer_objdata.properties 파일에서 **enableAsyncFileContentIndexing** 속성 값을 **false**로 설정하여 기본 색인기의 개체 데이터 구성을 업데이트합니다.

NX 또는 Solid Edge CAD 파일에 대한 비동기화 파일 내용 색인화 구성

NX 및 Solid Edge CAD 파일을 색인화하려면 구성이 필요합니다. 독점 파일 유형만 색인화하는 경우 이 구성이 필요하지 않습니다.

© 2024 Siemens

1. NX 및 Solid Edge 파일을 색인화하려면 색인화 가능한 유형으로 표시하십시오.

파일 유형	수행할 작업...
NX 파트 파일	• **Awp0SearchIsIndexed** 상수를 활성화하여 **UGMASTER** 데이터 집합을 색인화 가능한 유형으로 표시합니다. • **AW_Indexable_File_Extensions** 환경설정 또는 **Awp0IndexableFileTypes** 전역 상수에 확장 *.prt*를 추가합니다.
Solid Edge 드래프트 파일	• **Awp0SearchIsIndexed** 상수를 활성화하여 **SE 드래프트** 데이터 집합을 색인화 가능한 유형으로 표시합니다. • **AW_Indexable_File_Extensions** 환경설정 또는 **Awp0IndexableFileTypes** 전역 상수에 확장 *.dft*를 추가합니다.

2. 초기 색인화, 동기화 또는 사용 가능한 개체 데이터 색인화 방법을 사용하여 NX **UGMASTER** 데이터 집합 및 Solid Edge **SE 드래프트** 데이터 집합을 색인화합니다.

   • 비동기화 파일 내용 색인화를 설치한 후 처음으로 색인화하는 경우 전체 색인화를 수행해야 합니다.

   • 비동기화 파일 내용 색인화를 이미 설치했고 NX 및 Solid Edge 파일에 대한 색인화를 활성화하는 경우 전체 색인화 또는 델타 다시 색인을 수행할 수 있습니다.

   이제 파일 내용 색인화가 디스패처에 제출됩니다.

3. 이제 NX 및 Solid Edge의 CAD 파일 내용을 검색할 수 있습니다.

NX 파트 속성에 대한 색인화 및 검색 필터 구성

NX CAD 파일의 경우 파트 속성 및 PMI(제품 제조 정보)를 색인화하여 사용자가 Active Workspace에서 도면 노트, 테이블 및 레이블의 정보를 검색할 수 있습니다. Active Workspace에서 검색 필터로 표시하도록 NX 파트 속성을 구성할 수도 있습니다.

전제 조건

최소 버전의 NX 2212가 필요합니다.

절차

1. **\Dispatcher_root\Module\Translators\TcFTSIndexer\conf\filecontent\extractors***NXExtractorConfig.json* 파일을 열고 파트 속성 및 PMI에 대한 다음 구성을 **\UGII\extractors\indexer\schema***SampleConfiguration.json* 파일에서 복사합니다.

```
"attribute" :{
    "object" :["part", "solid body"]
},
```


© 2024 Siemens

```
"pmi" :{
    "object" :["notes","tables","labels"]
}
```

2. **aw_search_config_manager** 유틸리티를 사용하여 NX 매핑 구성 파일을 내보내 최신 버전이 있는지 확인합니다.

```
aw_search_config_manager -export -config_id=nxconfig -output=c:\nxconfig.json
```

3. **\Dispatcher_root\Module\Translators\TcFTSIndexer\conf\filecontent\extractors***nxconfig.json* 파일을 열고 필터로 표시할 파트 속성에 대한 구성을 추가합니다.

   mappingConfig.schema.json 파일에서 매핑 구성의 정의를 검토할 있습니다.

   예를 들어, NX 볼륨 값을 필터로 표시하려면 다음 구성을 추가합니다.

```
"path": "attribute.part.[*]"
"id": "part attributes",
"ruleSet": [
    {
        "fieldName": "title",
        "fieldValue": "value",
        "fieldUnit": "unit",
        "type": "plm.doubleList",
        "measure": "Volume",
        "conditions": [
            {
                "fieldName": "NX Volume"
            }
        }
    }
```

4. **aw_search_config_manager** 유틸리티를 사용하여 NX 매핑 구성 파일의 변경사항을 데이터베이스로 가져옵니다.

```
aw_search_config_manager -import -dir=mapping_configs
```

5. **runTcFTSindexer** 유틸리티를 사용하여 파일 내용 색인화가 실행 중인지 확인합니다.

```
runTcFTSIndexer -task=filecontent:test -standalone
```

6. **runTcFTSindexer** 유틸리티를 사용하여 *aw_search_consolidated_mapping.json* 파일이 업데이트되었는지 확인합니다.

```
runTcFTSIndexer -task=objdata:test
```

조건에 따라 CAD 파일이 색인화되는 상황 제한

모든 도면 수정에 대해 CAD 파일을 색인화하는 데 리소스가 많이 소모될 수 있습니다. Teamcenter 속성 기반 조건을 생성하여 도면이 색인화되는 횟수를 줄일 수 있습니다.

© 2024 Siemens

전제 조건

NX 또는 Solid Edge CAD 파일을 색인화하도록 색인기를 구성하거나 다른 CAD 파일에 대한 사용자 정의 절차를 따르십시오.

속성 기반 Teamcenter 조건 생성

이 조건은 그룹화 구성 파일에 추가됩니다. 이 파일은 기본 색인기를 그룹화하고 색인화할 디스패터에 제출할 데이터 집합을 지정합니다.

1. 기본 색인기 설치 *TC_ROOT*/**TcFTSIndexer/conf/***DatasetGroupingConfig.json*에서 그룹화 구성 파일을 엽니다.

   UGMASTER의 경우 다음과 같이 제공된 기본 그룹화 구성에 주의하십시오.

   ```
   UGMASTER": {
           "UGPART": [
               {
                   "name": "tcftsindexerfilecontentnxindex",
                   "limit": 10
               }
           ]
    }
   ```

   이 구성에서 색인기는 데이터 집합 **UGMASTER**와 연결된 명명된 참조 **UGPART**를 검색할 때마다 색인화를 위해 이를 디스패처 서비스 **tcftsindexerfilecontentindex**로 전송합니다.

2. *Teamcenter-internal-property-name*:*internal-value* 구문으로 JSON 개체를 사용하여 조건을 추가합니다.

   조건이 있는 경우 색인기에서는 조건이 충족된 경우에만 파일을 색인화합니다. 이렇게 하면 파일이 색인화되는 횟수가 줄어듭니다.

   > 팁:
   > - 그룹화 구성의 속성 이름 및 값은 대소문자를 포함하여 내부 Teamcenter이름 및 값과 정확히 일치해야 합니다.
   > - 속성 값은 문자열이어야 합니다. 조건에서 원하는 속성 값과 현재 속성 값 간에 정확한 일치 비교가 수행됩니다.
   > - 날짜 및 시간 속성을 사용하는 조건은 지원되지 않습니다.
   > - 조건에서 참조 속성을 사용하는 경우 상수 **Awp0SearchPropFromRefType**가 가리키는 첫 번째 소스 속성을 사용합니다.

   Add a single condition

© 2024 Siemens

예를 들어 **release status** 속성이 **Released**인 경우에만 **UGMASTER** 데이터 집합의 내용을 색인화하려면 아래 조건을 추가합니다.

```
“UGMASTER”: {
        “UGPART”: [
            {
                “name”: “tcftsindexerfilecontentnxindex”,
                “limit”: 10,
             “conditions”:[
                 {
            “release_status_list”: [ “Released” ]
                    }
             ]
            }
        ]
 }
```

Add a single condition with multiple values in an AND relationship

예를 들어, 프로젝트 **ProjectA** 및 **ProjectB**에 할당된 데이터 집합을 색인화하려면 아래 조건을 추가합니다.

```
“conditions”: [
    {
        “project_list”: [ “ProjectA”, “ProjectB” ],
        “operation”: “AND”
    }
]
```

연산 필드가 비어 있으면 **OR**이 기본 연산자입니다.

Add a single condition with multiple values in an OR relationship

예를 들어, **ProjectA** 또는 **ProjectB**에 속하는 데이터 집합을 색인화하려면 아래 조건을 추가합니다.

```
“conditions”: [
    {
        “project_list”: [ “ProjectA”, “ProjectB” ]
    }
]
```

연산 필드가 지정되지 않았으므로 **OR**이 기본 연산자입니다.

Add multiple conditions with multiple values

예를 들어, **release_status_list**에 **Released** 및 **TCM Released**가 포함되어 있고 **owning_group**에 **design** 또는 **engineering**이 포함된 경우에만 데이터 집합을 색인화하려면 아래 조건을 추가합니다.

```
“conditions”: [
    {
        “release_status_list”: [ “Released”, “TCM Released” ],
        “operation”: “AND”
    },
    {
        “owning_group”: [ “design”, “engineering” ]
```


© 2024 Siemens

```
        }
    ]
```

여러 조건 사이에는 **AND** 관계가 있습니다.

지정된 위치에서 ITAR 제어 데이터 집합 처리

규정 준수를 유지하려면 ITAR 데이터에 대한 액세스를 제어해야 합니다. 특정 위치에서 처리해야 할 수 있습니다. 다음 절차를 사용하여 특정 지역을 기반으로 하는 서버의 특정 데이터 집합을 처리하도록 비동기화 파일 내용 색인화를 구성할 수 있습니다.

전제 조건

ITAR 호환 디스패처 설치 및 구성[5]를 사용할 수도 있습니다. 예를 들어, 미국 기반 볼륨에 액세스하는 미국 기반 디스패처를 구성합니다.

특정 데이터 집합을 특정 서버로 보내도록 디스패처 구성

이 예제에서는 미국 기반 서버에서 실행되는 것으로 가정되는 미국 기반 디스패처 서비스를 생성합니다. 그런 다음 처리를 위해 이 서비스로 보낼 파일을 지정합니다. 이 예제에서는 표준 파일(예: Microsoft Office 파일, PDF 및 텍스트 파일)을 지원하는 **tcftsindexerfilecontenttikaindex** 서비스를 사용합니다. 시스템 구성에 적합한 서비스를 선택하십시오.

색인화할 파일 유형...	검색할 서비스...
표준 파일(예: Microsoft Office 파일, PDF, 및 텍스트 파일)	**tcftsindexerfilecontenttikaindex**
NX	**tcftsindexerfilecontentnxindex**
Solid Edge	**tcftsindexerfilecontentsolidedgeindex**
기타 독점 파일	사용자 정의 서비스

1. *Dispatcher_Root***Module****conf***translator.xml* 구성 파일의 OOTB(out-of-box) **tcftsindexerfilecontenttikaindex** 서비스를 템플릿으로 사용하여 새 디스패처 변환기 서비스를 생성합니다.

   기존 **tcftsindexerfilecontenttikaindex** 서비스 정의를 복사하고 이름을 **tcftsindexerfilecontenttikaindex_us**로 변경합니다.

   원본 서비스 정의에 표시된 표준 카멜 대/소문자 구문과 일치하기 위해 XML 요소 이름을 변경합니다. **TcFtsIndexerFileContentTikaIndex_US** 표준 카멜 대/소문자 구문을 따릅니다.

   ```
   <TcFtsIndexerFileContentTikaIndex_US provider="SIEMENS"
        service="tcftsindexerfilecontenttikaindex_us"
        maxlimit="3" isactive="true">
        <TransExecutable dir="&MODULEBASE;/Translators/TcFTSIndexer/bin"
   name="runTcFTSIndexer.bat" />
   ```

5 에 대한 자세한 내용은 Teamcenter 문서의 디스패처 소개*를 참조하십시오*.

© 2024 Siemens

```
        <Options>
            <Option name="flow" string="-task=filecontent:tika"
    description="TcFtsIndexer flow to extract and index file contents using Apache
    Tika."/>
            <Option name="mode" string="-standalone" description="Switch to indicate
    that TcFtsIndexer needs to run in standalone mode."/>
            <Option name="inputpath" string="-i=" description="Full path to the input
    file or directory."/>
        </Options>
        <TransErrorExclStrings>
            <TransInputStreamExcl string="Error: 0"/>
        </TransErrorExclStrings>
    </TcFtsIndexerFileContentTikaIndex_US>
```

서비스 정의의 미리 알림은 기본 **tcftsindexerfilecontenttikaindex** 서비스와 동일하게 유지됩니다. 이 서비스는 플로 작업 입력이 **-task=filecontent:tika**로 설정된 상태에서 **runTcFTSIndexer** 유틸리티를 호출합니다.

isactive가 **true**로 설정되어 있어야 합니다.

2. 기본 색인기에서 지정된 데이터 집합을 새 서비스로 라우팅하는 그룹화 구성을 생성합니다.

*TC_ROOT***TcFTSIndexer\conf***DatasetGroupingConfig.json*를 수정하고 다음 규칙을 추가합니다.

```
{
    “MSWordX”: {
        “word”: [
            {
                “name”: “ tcftsindexerfilecontenttikaindex_us”,
                “limit”: 1000,
                “conditions”: [
                    {
                            “owning_group”: [ “engineering” ]
                    }
                ]
            }
        ]
    },
    “default”: {
        “name”: “tcftsindexerfilecontenttikaindex”,
        “limit”: 1000
    }
}
```

이 구성은 그룹 **엔지니어링**에서 소유한 모든 MSWordX 데이터 집합을 서비스 **tcftsindexerfilecontenttikaindex_us**에 라우팅합니다. 이 서비스는 미국 기반 서버에서 실행되는 것으로 가정됩니다. 다른 모든 색인화 가능한 데이터 집합은 기본 **tcftsindexerfilecontenttikaindex** 서비스에서 계속 처리됩니다.

색인된 파일 내용에 대해 패싯 비활성화

색인화된 파일 내용이 검색 필터에 패싯을 추가하지 않도록 패싯을 사용하지 않도록 설정할 수 있습니다.

© 2024 Siemens

절차

1. **필터** 속성을 **false**로 설정하여 매핑 구성 JSON 파일을 업데이트합니다. 매핑 사양 문서 *TC_ROOT*/**TcFTSIndexer/conf/specification/***Mapping_Specification.docx*을 참조하십시오.

2. **aw_search_config_manager** 유틸리티를 실행하여 새 매핑 규칙으로 구성 저장소를 업데이트합니다.

   예:

   ```
   aw_search_config_manager -u=admin -p=pwd -g=dba -import -dir=c:\mappi
   ngConfigs
   ```

3. 패싯 변경 사항을 표시하려면 새 Teamcenter 세션을 시작하십시오.

비동기화 파일 내용 색인화 연결 확인

비동기화 파일 내용 색인기 응용 프로그램을 설치하고 실행해야 합니다.

1. 비동기화 파일 내용 색인기 응용 프로그램이 설치되어 있는지 확인합니다.

   - **AWS_asynchronous_file_content_indexing_enabled** 환경설정이 **true**로 설정되어 있는지 확인합니다.

   - (선택 사항) 사용자 정의 구성을 통해 NX, Solid Edge 또는 기타 독점 파일에 대해 색인화가 활성화된 경우 **aw_search_config_manager** 유틸리티를 실행하여 매핑 구성이 설치되었는지 확인합니다.

     ```
     aw_search_config_manager -u=admin -p=pwd -g=dba -list
     ```

2. 비동기화 파일 내용 색인기 응용 프로그램은 디스패처 모듈의 일부로 **TcFTSIndexer** 클라이언트를 설치합니다. *Dispatcher_Root/Module/Translators/TcFTSIndexer/bin* 디렉터리에서 색인기의 **filecontent:test** 플로 작업을 실행하여 필요한 모든 전제 조건이 충족되었는지 확인합니다.

   ```
   runTcFTSIndexer.bat -task=filecontent:test
   ```

   이 플로 작업은 다음을 포함하는 상태 보고서를 표시합니다.

   - Teamcenter, FMS 및 Solr에 대한 연결 확인.

   - 모든 필수 구성 파일이 있는지 확인합니다.

© 2024 Siemens

데이터 색인화

개체 데이터의 전체 색인 수행

초기 색인을 실행하는 경우 **TcFTSIndexer의 인스턴스를 최적화**해야 합니다. 성능을 향상하기 위해 특정 시간 조각 또는 일괄 처리 크기를 색인화할 수 있습니다.

> 주:
>
> 1천만 개를 초과하는 개체를 단일 색인 실행으로 색인화하지 않을 것을 권장합니다. 천만 개를 초과하는 개체를 단일 색인 실행으로 색인화해야 하는 경우에는 개체가 천만 개 추가될 때마다 **TC_ROOT\TcFTSIndexer\bin\runTcFTSIndexer.bat**에서 **xmx***Ng*의 *N* 값을 1씩 높일 수 있습니다. (초기 기본값은 **3g**입니다.)

1. **TcFTSIndexer 연결 테스트** 결과 오류가 없으면 초기 인덱스를 실행할 준비가 된 것입니다. 다음이 실행 중인지 확인합니다.
   - Teamcenter 데이터베이스
   - Solr
   - Teamcenter 웹 계층
   - 서버 관리자(Server manager)
   - Active Workspace 게이트웨이
2. **runTcFTSIndexer -task=objdata:sync -interval=***seconds* 명령을 실행할 경우, 다음 명령을 사용하여 이를 일시적으로 종료합니다.

   ```
   runTcFTSIndexer -task=objdata:sync -stop
   ```

3. (선택 사항) 기존의 검색 데이터를 지우고 새 색인을 수행하려면 Teamcenter 명령 프롬프트를 열고 *TC_ROOT***\TcFTSIndexer\bin**으로 이동하여 다음과 같은 명령을 입력합니다.

   ```
   runTcFTSIndexer -task=objdata:clear
   ```

4. **색인기** 기능이 설치된 시스템에서 명령 프롬프트를 엽니다.
5. TcFTSIndexer의 **bin** 디렉터리(예: *TC_ROOT***\TcFTSIndexer\bin**)로 이동합니다.
6. 개체 데이터 색인화의 경우 다음 명령을 실행합니다.

   ```
   runTcFTSIndexer -task=objdata:index
   ```


© 2024 Siemens

데이터베이스에 기존 데이터가 있는 경우 초기 색인화를 실행하는 데 다소 시간이 걸릴 수 있습니다.

기본적으로 색인기에서 현재 날짜와 시간을 끝 점으로 사용합니다. 다른 날짜 및 시간을 지정하려면 TEM에서 TcFTSIndexer 설정을 사용하십시오.

다음에 정의된 날짜부터 데이터를 색인화하려면

*TC_FTS_INDEXER_HOME***conf***TcFTSIndexer_objdata.properties*

*TC_FTS_INDEXER_HOME***cache**에서 **TcFTSIndexerSessionobjdata.cache**를 삭제해야 합니다.

7. 색인 오류 발생 여부를 확인합니다. 콘솔에서 생성되어 *TC_ROOT***TcFTSIndexer****logs***TcFTSIndexer_objdata.log*에 저장된 보고서를 확인합니다.

8. 오류가 있는 경우 **복구** 옵션을 실행합니다.

   ```
   runTcFTSIndexer -task=objdata:recover
   ```

   모든 색인화 가능한 개체가 성공할 때까지 **objdata:recover**를 반복해서 실행하고 나머지 오류는 반복적으로 실패합니다. 색인 오류 해결 여부를 확인합니다.

   오류를 수정하거나 나중에 되돌아가서 수정할 수 있습니다. 그대로 두는 경우, 이러한 오류는 동기화 중에도 실패로 기록됩니다.

   runTcFTSindexer 유틸리티는 다양한 색인화 작업을 실행할 수 있습니다.

9. 다음이 실행 중인지 확인합니다.

   - Teamcenter 데이터베이스
   - 서버 관리자(Server manager)
   - Teamcenter 웹 계층
   - Teamcenter web client 게이트웨이

10. Active Workspace에서 검색을 실행하여 색인화가 작동되는지 테스트합니다. Active Workspace 게이트웨이에 대한 URL을 엽니다.

    검색 상자에 색인화된 알고 있는 용어를 입력하고 **검색**을 클릭합니다.

    입력한 용어와 관련된 결과가 표시되고 오류가 표시되지 않으면 Active Workspace 배치가 올바르게 작동됩니다.

11. 동기화 플로를 종료하거나 이 색인이 초기 색인인 경우 동기화 플로를 시작할 수 있습니다. 색인기를 서비스로 설치한 경우 **서비스를 시작**할 수 있습니다.

© 2024 Siemens

```
runTcFTSIndexer -task=objdata:sync
```

주의:

Teamcenter에서 개체 데이터가 추가, 수정 또는 삭제되었지만 동기화 플로가 실행되지 않는 경우 Active Workspace는 해당 변경 사항의 결과를 반환하지 않습니다.

복제된 환경에서 업데이트 및 색인화로 다운타임 최소화

전체 색인화가 필요한 Active Workspace 업그레이드 또는 패치의 경우 복제 및 병합 방법을 사용하여 다시 색인화 프로세스를 진행할 수 있습니다. 현재 생산 환경의 복사본을 만들고, 복제를 업그레이드하고, 다시 색인화를 수행한 다음, 다시 색인화된 파일을 업데이트된 생산 환경으로 병합합니다. 이후에 복제하는 경우와 병합할 때 사이에 수행된 모든 중간 변경 내용을 색인화합니다.

전제 조건

복제 또는 생산 환경을 업그레이드하기 전에 테스트 환경에서 업그레이드 또는 패치를 테스트합니다.

절차

다음을 실행하여 언제든지 생산 환경에서 복제 상태를 확인할 수 있습니다.

```
runTcFTSIndexer.bat -task=admin:status
```

1. 다음을 실행하여 생산 환경을 *Clone Ready*로 표시합니다.

   ```
   runTcFTSIndexer.bat -task=admin:cloneready
   ```

   이 작업은 *SOLR_HOME*\server\solr*admin*에서 생산 환경에 복제 상태를 적용합니다.

   환경이 *Clone Ready*로 표시되는 동안 **runTcFTSIndexer -task=objdata:clear** 실패를 실행하려고 시도합니다. 이 오류는 이 프로세스에서 실수로 색인을 지우는 것을 방지합니다.

2. 생산 환경에서 동기화 흐름을 실행하는 경우 다음을 실행합니다.

   ```
   runTcFTSIndexer -stop -task=objdata:sync
   ```

3. 복제된 환경으로 생산 환경을 복사합니다(여기서는 복제된 환경이라고 함).

   SOLR 색인 및 데이터베이스를 포함해야 하므로 전체 생산 환경의 정확한 복제본을 만들어야 합니다.

   SOLR_HOME\server\solr*collection-name*\data

4. 파일을 복사한 후 복제된 환경이 업데이트되는 동안 생산 환경에서 색인 동기화를 다시 시작합니다. 복제된 환경을 다시 업데이트된 생산 환경에 병합한 후 색인화 변경 델타를 선택합니다.

© 2024 Siemens

5. 복제된 환경을 업그레이드합니다. 일반 업그레이드 또는 패치 절차를 따릅니다. 생산 환경에 적용할 동일한 작업을 수행합니다.

   복제된 환경에 사용자 정의를 적용하지 마십시오. 복제된 환경이 업데이트된 생산 환경과 병합된 후에 이러한 내용을 적용할 수 있습니다.

6. 업그레이드 또는 업데이트에 Solr의 새 버전이 포함되어 있는 경우(예를 들어, Solr이 버전 7.7에서 8.6으로 이동하는 경우) 복제된 환경에서 이전 버전의 Solr에서 복제된 환경에 있는 Solr의 새 버전에서 동일한 위치로 **SOLR_HOME\server\solr\admin/data**를 복사합니다.

7. 복제된 환경에서 개체, 속성 또는 기타 값을 업데이트한 다음 복제된 환경에서 다시 색인화를 시작합니다.

   ```
   runTcFTSIndexer -task=objdata:index
   ```

8. 복제된 환경에 적용한 동일한 업그레이드 또는 패치로 생산 환경을 업그레이드합니다. 일반 업그레이드 또는 패치 절차를 따릅니다.

   현재 생산 환경에 사용자 정의를 적용하지 마십시오. 복제된 환경이 업데이트된 생산 환경과 병합된 후에 이러한 내용을 적용할 수 있습니다.

9. 복제된 환경에서 색인화가 완료되면 Solr 컬렉션 데이터를 복제된 환경에서 업데이트된 생산 환경의 동일한 위치로 복사합니다.

   SOLR_HOME\server\solr*name-of-collection*\data

   admin, **collection1**, **collection2**, **ProductScopeCollection** 등을 포함하여 모든 컬렉션을 복사해야 합니다.

10. 생산 환경에서 생산 환경에 복사한 Solr 데이터가 다음을 실행하여 예상되는 것과 일치하는지 확인합니다.

    ```
    runTcFTSIndexer -task=admin:clonevalidate
    ```

    Solr과 데이터베이스 사이에 날짜가 일치하지 않는 경우 오류 메시지는 문제 해결 방법에 대한 지침을 제공합니다. 이 유형의 유효성 검사는 데이터 충돌을 찾지 않습니다.

11. 유효성 검사가 완료되고 성공하면 시스템이 *복제 준비 완료*로 표시된 후 변경된 델타를 사용하여 생산 환경 색인을 업데이트합니다.

    ```
    runTcFTSIndexer.bat -task=objdata:index clone
    ```

12. 색인화가 실패하지 않고 나머지 변경 사항을 선택하면 생산 환경에서 *Clone Ready* 상태를 제거합니다.

    runTcFTSIndexer -task=admin:clonecomplete

© 2024 Siemens

```
runTcFTSIndexer -task=admin:clonecomplete
```

13. 생산 환경에서 동기화 흐름을 다시 시작합니다.

```
runTcFTSIndexer -task=objdata:sync
```

색인화에 대한 데이터 모델 변경 평가

데이터 모델 업데이트 후 **수정된 데이터**를 색인화해야 합니다. 델타 색인을 완료하여 변경사항을 포개할 수 있습니다. 그러나 전체 색인과 델타 색인은 데이터 모델 업데이트로 인한 영향 받는 유형 및 속성 변경 비율이 높기 때문에 처리 시간이 유사할 수 있습니다. 영향을 계산하려면 변경사항을 식별하고, 변경 수를 색인화 인프라에서 지원할 수 있는 개체 수와 비교하여 변경 불륨을 평가하고, 델타 색인이 적절한지 또는 전체 색인이 필요한지 여부를 판단합니다.

색인화 변경 내용이 유형 및 속성에 대한 추가, 수정 및 삭제인 경우, 전체 색인화가 아닌 델타 색인화 업데이트를 수행할 수 있습니다.

1. Teamcenter Rich Client 및 Active Workspace가 패치되지 않은 경우 **Teamcenter 및 Solr 스키마를 병합합니다**.

2. 색인기를 사용하여 동기화를 중지합니다(실행 중인 경우).

   ```
   runTcFTSIndexer -stop
   ```

3. 마지막 색인화 스키마와 현재 스키마 간의 변경 범위를 결정합니다.

   -delta -dryrun과 함께 awindexerutil 유틸리티를 실행하여 변경사항의 예상 델타 보고서를 가져옵니다. 예:

   ```
   awindexerutil -u=adminuser -p=password -g=group -delta -dryrun
   ```

   차이점은 로그 파일과 명령 윈도우에 출력됩니다.

4. 보고서를 평가한 후 변경 델타를 사용하여 색인화할 것인지 결정합니다.

   - 대량 또는 다양한 변경사항이 있는 경우 **전체 색인**을 완료합니다.
   - 색인화에 변경 델타를 사용하려는 경우 **awindexerutil** 유틸리티를 실행하여 색인화에 대한 보고서의 변경사항을 설정합니다.

     ```
     awindexerutil u=adminuser -p=password -g=group -delta
     ```

     식별된 업데이트는 동기화 플로와 함께 색인화됩니다. 모든 변경사항을 색인화하기 위해 동기화 플로의 인스턴스가 여러 개 걸릴 수 있습니다.

5. 색인기 테스트 플로를 실행하여 색인기 연결을 테스트합니다.

© 2024 Siemens

```
runTcFTSIndexer -task=objdata:test
```

6. runTcFTSIndexer 유틸리티를 사용하여 동기화 플로를 다시 시작합니다.

   ```
   runTcFTSIndexer -task=objdata:sync -interval=seconds
   ```

TcFTSIndexer 동기화 플로 설정

초기 색인화를 실행한 후 지정된 간격으로 동기화하도록 색인화를 설정할 수 있습니다. 색인기를 서비스로 설치한 경우 서비스를 시작할 수 있습니다.

1. (선택 사항) 동기화 플로를 사용하여 많은 양의 데이터를 색인화하는 경우 이러한 작업을 완료하여 일괄 처리 크기를 제한하고 모든 데이터가 성공적으로 색인화되도록 합니다.

   - 일괄 처리 크기 한계를 허용하려면 **TC_Indexer_Enable_Sync_In_Batch** 환경설정을 구성합니다.
   - 일괄 처리 크기 한계의 개체 수를 지정하려면 **TC_Indexer_Sync_Batch_Threshold**를 구성합니다.

2. 색인기가 설치된 시스템의 *TC_ROOT***TcFTSIndexer\bin** 디렉터리로 이동합니다.

3. 상황에 적용되는 옵션을 선택합니다.

   - 개체 색인화만 사용할 경우에는 다음 단계로 건너뜁니다.
   - 여러 유형의 색인화(예: 개체 색인화 및 구조 색인화)를 사용하는 경우 서비스 모드에서 **TcFTSIndexer**를 시작합니다.

     서비스 모드에서 TcFTSIndexer를 사용하려면 -background 옵션을 사용합니다. **-background** 옵션을 사용하여 **TcFTSIndexer** 프로세스를 지속적으로 실행하고 여러 유형의 색인화 오퍼레이션을 동시에 실행할 수 있습니다.

     예제:

     runTcFTSIndexer -background

4. **runTcFTSIndexer -task=objdata:sync -interval=***x*를 입력합니다. 여기서 *x*는 동기화 오퍼레이션을 다시 실행하기 전에 대기하는 시간(초)으로 **0**보다 큰 수입니다.

   예를 들어, 25초 동안 대기시키려면 다음을 입력합니다.

   runTcFTSIndexer -task=objdata:sync -interval=25

   -background 옵션을 이전에 사용한 경우, 해당 명령이 서비스로 전송되고 서비스에서 해당 명령을 실행합니다. 그렇지 않은 경우, 현재 프로세스에서 작업이 실행됩니다.

5. **TcFTSIndexer** 프로세스를 중지하려면 다음 중 하나를 선택합니다.

© 2024 Siemens

- 모든 플로가 실행을 중지한 후 프로세스를 중지하려면 **runTcFTSIndexer -stop**을 입력하고 **runTcFTSIndexer -status**를 입력합니다.

  이 명령을 수행했을 때 실행 중인 플로가 반환되지 않으면 프로세스가 중단됩니다. 프로세스를 중단할 시간을 부여하려면 잠시 기다렸다가 이 명령을 반복해야 할 수 있습니다. 서비스가 계속해서 실행됩니다.

- 서비스를 중지한 후 종료하려면 **runTcFTSIndexer -shutdown**을 입력합니다.

  플로의 완료가 허용되면 서비스가 중단됩니다.

6. 새 윈도우에서 TcFTSIndexer 서비스를 실행할 수 있습니다.

   원하는 간격을 사용하여 개체 데이터 동기화를 시작합니다. 다음 예에서 간격은 25초입니다.

   runTcFTSIndexer -task=objdata:sync -interval=25

   > 팁:
   >
   > 대량의 데이터를 마이그레이션하는 경우 마이그레이션된 데이터를 동기화 플로와 별도로 색인화하여 성능을 향상시킬 수 있습니다.
   >
   > 데이터베이스 에서 **fast_sync_add_trigger**를 비활성화하여 마이그레이션된 데이터가 동기화 플로에서 인덱싱되지 않도록 하고 마이그레이션 중에 동기화 플로를 중지합니다. 마이그레이션이 완료되면 **fast_sync_add_trigger**를 활성화하고 동기화 플로를 재개합니다. 마이그레이션 동안 **타임 슬라이스를 사용**하여 별도의 **TcFTSIndexer** 인스턴스에서 마이그레이션된 데이터를 색인화인덱싱합니다.

동기화는 풀 관리자 및 Solr에 연결된 단일 Java SOA 클라이언트로 실행되어 델타 색인화를 수행합니다. 동기화에서는 간격에 사용할 추가 인수를 갖습니다. 이 간격은 동기화 오퍼레이션이 다시 실행 가능해질 때까지 대기해야 하는 시간을 지정합니다.

동기화는 다음 작업을 수행합니다.

- 이전 동기화 동안에 색인화에 실패한 개체를 조회합니다.

  실패한 개체가 있는 경우 경고 메시지가 로그에 기록됩니다.

  ***NNN* 실패 개체가 있습니다. 이런 오류 중 일부는 이후 동기화 호출에서 고칠 수도 있습니다. 이런 오류가 지속되면 이를 수동으로 고쳐야 하며 "indexsyncfailures 플로"를 다시 실행해 이런 오류를 색인화해야 합니다.**

  이 경우 **TcFtsIndexer**를 **-task=objdata:indexsyncfailures** 옵션과 함께 실행합니다.

  다음 동기화는 실패한 개체를 선택하고 이에 대한 복구를 시도합니다.

© 2024 Siemens

- 리비전 규칙이 변경된 모든 개체를 조회하고 TIE 내보내기 및 변환을 통해 개체 리스트를 실행합니다. 그런 다음 Solr에 로드합니다.

  이전에 색인화된 리비전만 동기화됩니다.

  예를 들어, 리비전 A, B, C가 있고 B와 C만 색인화된다고 가정합니다. 새 리비전 생성, 기존 리비전 삭제 또는 기존 리비전 상태 변경과 같은 변경 사항은 모든 형제를 영향 받은 리비전 규칙으로 확인하고 동기화되어야 합니다. 이 단계에서는 색인에 있는 형제만 동기화합니다. 이 경우 리비전 B와 C만 동기화됩니다.

- 삭제된 개체를 조회하고 Solr에 연결한 후 해당 개체를 삭제합니다.

- 새 개체를 조회하고 TIE 내보내기 및 변환을 통해 개체 리스트를 실행합니다. 그런 다음 Solr에 로드합니다.

- Access Manager 규칙 변경의 영향을 받는 개체를 조회하고 개체 리스트, 변환 및 Solr에 로드하는 READ 수식을 가져옵니다.

- 수정된 개체를 조회하고 변환을 통해 개체 리스트를 실행한 후 Solr로 로드합니다.

- 개체를 새로 고칩니다.

  이 단계 동안 시스템은 **:FTS_REFRESH** 응용 프로그램 ID가 마지막으로 처리된 날짜보다 이전에 내보냈던 색인화된 개체를 조회합니다. 이러한 개체를 다시 색인화해야 합니다.

  refreshBatchSize 속성 값은 **TcFtsIndexer_objdata.properties** 파일에서 구성됩니다. 조회되고 색인화되는 개체의 최대 수는 기본적으로 **10000**으로 설정됩니다. 새로 고칠 개체 수가 10,000개를 초과하거나 해당 파일의 제한 값을 초과하면, 다음 동기화 호출에서 추가 개체가 처리됩니다.

- 복제 보류 중인 개체를 조회합니다.

  이들은 이전 단계 중에 TIE 내보내기, 변환, 로드 과정에서 오류가 발생했을 수 있는 개체입니다.

  이러한 개체에 대한 색인화는 3번 다시 시도됩니다. 계속 실패하는 경우 이들은 *import failed*로 표시되며 다음 동기화 호출에서 처리됩니다.

- 이벤트 통보 요청한 모든 응용 프로그램에서 사용되었던 개체 데이터를 정리합니다.

동기화되는 개체 수는 콘솔 및 *TC_ROOT***TcFTSIndexer\logs\TcFtsIndexer.log** 파일에 출력됩니다. 동기화 실행 간격은 동기화 사례에서 처리하는 일반적인 개체 수를 기반으로 정의해야 합니다. 개체 수가 너무 많으면 동기화 간격을 줄이십시오. 또한 사용자가 이러한 변경된 개체에 액세스하는 속도를 고려하십시오.

동기화 오퍼레이션이 진행 중이고 간격에 따라 다음 번의 예약된 동기화 시간이 된 경우, 시스템은 다음 번 예약된 동기화를 건너뛰고 현재 동기화 오퍼레이션이 완료될 때 진행합니다.

© 2024 Siemens

사용자 정의 템플릿을 업데이트한 후 색인기 실행

사용자 정의 템플릿 패키지를 변경할 때마다 데이터베이스에 다시 배포하고 색인기를 실행해야 합니다.

1. 사용자 정의 템플릿 패키지를 설치합니다. Business Modeler IDE의 템플릿에 대한 자세한 내용은 Teamcenter 도움말에서 템플릿 배포 소개를 참조하십시오.

   주:

   핫 배포 또는 실시간 업데이트를 사용하여 변경한 경우, 나중에 유틸리티를 수동으로 실행해야 합니다. **TC_ROOT** 및 **TC_DATA**가 정의되어 있는 Teamcenter 명령 윈도우에서 다음 명령을 실행합니다.

   ```
   bmide_modeltool.bat -u=username -p=password -g=group -tool=all
      -mode=upgrade -target_dir="TC_DATA"
   ```

2. **Teamcenter 및 Solr 스키마를 병합합니다**.

3. **검색 데이터를 색인화합니다**.

   사용자는 서버에 설정된 마스터 언어로 검색할 수 있으며, 이 언어는 **SiteMasterLanguage** 전역 상수로 정의됩니다. **SiteMasterLanguage** 전역 상수 값을 변경하는 경우, 해당 언어에 대해 새로운 분석기 집합으로 데이터를 다시 색인화해야 하기 때문에 **objdata:index** 옵션을 사용하여 색인기를 다시 실행해야 합니다.

다중 사이트 게시 개체 색인화

runTcFTSIndexer 유틸리티를 실행해 공유 게시 레코드를 색인화합니다. **runTcFTSIndexer 유틸리티**의 **-task=multisite** 플로 작업을 사용해 게시된 개체를 색인화합니다. 중복 게시된 레코드는 색인화 중 제거됩니다.

- **-task=multisite:test**

  환경이 올바르게 설정되었는지 확인한 후 색인화합니다.

- **-task=multisite:index**

  기존 색인을 지우지 않고 *TcFTSIndexer_multisite.properties*에 지정된 기간 동안 색인화를 실행합니다.

- **-task=multisite:recover**

  실패한 다중 사이트 개체에서 색인화를 실행합니다.

- **-task=multisite:sync**

  이전 실행과 현재 시간 중 발생한 변경 사항에 대한 다중 사이트 색인을 업데이트합니다.

© 2024 Siemens

선택적 **-interval** 인수는 초 단위로 지정된 간격에서 **동기화** 작업을 반복합니다. **동기화**를 한 번만 실행하려면 **-interval** 인수를 생략합니다.

- **-task=multisite:indexsyncfailures**

  수동 개입이 필요한 다중 사이트 동기화 실패에 대하여 색인화를 실행합니다.

- **-task=multisite:clear**

  기존 다중 사이트 게시 레코드 색인 데이터 및 캐시 파일을 지웁니다.

- **-task=multisite:indexuids**

  특정 개체 디렉터리 서비스(ODS) 사이트에 대한 UID를 색인화합니다. ODS 사이트 이름 및 *uid.txt* 파일로의 경로를 지정합니다. 예:

  ```
  runTcFTSIndexer -task=multisite:indexuids ODS_Site_name D:\UID_dir
  ```

다중 사이트에 대한 검색 구성을 확인하여 사용자가 다중 사이트 복제 데이터를 사용할 수 있도록 합니다.

색인화 유틸리티 참조

색인화 유틸리티

유틸리티	다음 사용
runTcFTSIndexer	작업 플로를 사용하여 데이터를 색인화합니다.
awindexerutil	동기화 플로를 실행할 때 색인된 개체의 색인을 새로 고침합니다.
aw_search_config_manager	CAD 파일과 같은 비표준 데이터 집합에서 추출된 데이터를 색인화하는 데 사용되는 매핑 구성을 관리합니다.

© 2024 Siemens

runTcFTSIndexer

Solr 색인화 엔진에 데이터를 색인화합니다. *FTS_INDEXER_HOME* 디렉터리(예: *TC_ROOT***TcFTSIndexer\bin**)에서 이 명령을 실행합니다. 이 명령은 특정 환경에서 실행할 필요는 없습니다.

구문

runTcFTSIndexer -debug -maxconnections -status -stop -service -shutdown -task=[objdata | multisite | structure | fourgd]:*flow-action* **-h**

주:	
운영 체제 고려사항:	
Linux	*.sh* 확장이 필요합니다. 예: **./runTcFTSIndexer.sh**. UID에 특수 문자가 포함되어 있으면 이를 굵은 작은따옴표로 묶어야 합니다. UID는 **A-Z**, **a-z**, **0-9**, **$** 및 **_**을 포함할 수 있습니다.
Windows	**runTcFTSIndexer**에는 *.bat* 확장이 필요하지 않습니다. Windows에서 UID는 굵은 작은따옴표로 묶지 마십시오.

인수

-debug

진행 중인 플로의 요약을 연관된 연결 및 로그를 포함하여 명령 윈도우 및 **TcFTSIndexer\logs**내 **TcFTSIndexer** 로그에 보고합니다. 별도의 명령 윈도우에서 **-debug**를 실행합니다.

```
runTcFTSIndexer -debug
```

-maxconnections

다음 예와 같이, 최대 **tcserver** 연결에 대한 새로운 값을 설정해 항상 사용할 수 있도록 합니다.

```
runTcFTSIndexer -maxconnections=5
```

-status

색인기 상태를 확인하고 색인기에서 실행 중인 플로를 보고합니다. 아래 예와 같은 경우, 모든 플로에 대한 상태가 표시됩니다.

```
runTcFTSIndexer  -status
```

이 인수는 **-task** 인수와 함께 사용할 수도 있습니다. 예를 들어, 다음 명령에는 **objdata:reindex** 플로의 상태가 표시됩니다.

```
runTcFTSIndexer -status -task=objdata:index
```


© 2024 Siemens

-stop

간격을 사용하는 색인화 플로를 중단합니다.

```
runTcFTSIndexer -stop
```

이 인수는 **-task** 인수와 함께 사용할 수도 있습니다. 예를 들어, 다음 명령을 실행하면 간격이 있는 **objdata:sync** 플로가 중단됩니다.

```
runTcFTSIndexer -stop -task=objdata:sync
```

-service

다음 예와 같이, 콘솔에서 색인기를 Java RMI(Remote Method Invocation) 서비스로 시작합니다.

```
runTcFTSIndexer -service
```

이 인수는 **-task** 인수와 함께 사용해 서비스 시작 시 플로를 실행하도록 할 수 있습니다. 다음 예를 참조하십시오.

```
runTcFTSIndexer -service -task=objdata:index
```

-shutdown

모든 플로를 중지한 후 서비스를 종료합니다.

```
runTcFTSIndexer -shutdown
```

-task

다음 형식으로 색인화 타스크를 실행합니다.

`-task=`*`type`*`:`*`flow-action`*

플로 작업은 타스크 유형에 따라 다릅니다.

objdata	개체 데이터 색인화를 지원하는 색인화 작업
admin	복제 및 병합 색인화를 지원하는 관리 작업
fourgd	4GD를 지원하는 색인화 작업
multisite	다중 사이트 게시 레코드를 지원하는 색인화 작업
structure	구조화된 내용 색인화를 지원하는 색인화 작업

-h

이 유틸리티의 도움말을 표시합니다.

© 2024 Siemens

OBJDATA 플로 작업

- **objdata:build_suggester**

  Solr에 대한 제안 사전을 재구성합니다. 이 플로는 동기화 플로와는 별개입니다. **TcFtsIndexer_objdata.properties** 파일의 **objdatasyncstep.build_suggester_inline** 속성이 **false**로 설정된 경우 **objdata:build_suggester**를 실행합니다.

  - 이 작업은 예를 들어, **-interval=***seconds* 인수를 사용하여 5분(300초)마다 제안 사전을 재구성할 수 있습니다.

    ```
    runTcFTSIndexer -task=objdata:build_suggester -interval=300
    ```

    간격 값은 300 이상이어야 합니다. 이 플로 작업을 간격으로 실행하는 것이 좋습니다.

  - 예를 들어, 다음과 같은 제안 사전을 한 번 재구성할 수 있습니다.

    ```
    runTcFTSIndexer -task=objdata:build_suggester
    ```

- **objdata:clear**

  기존 색인화된 데이터, 레코드 및 캐시된 파일을 지웁니다. 이 옵션은 **objdata:index**를 실행하기 전에 가장 자주 사용됩니다. **-task=objdata:clear**를 실행할 때 다음 메시지가 표시되면 이러한 옵션 중 하나를 지정합니다.

  1은 색인기 캐시를 지웁니다.

  2는 Solr 색인을 지웁니다.

  3은 책임 테이블에서 레코드를 색인화하는 개체 데이터를 지웁니다.

  4는 옵션 1, 2 및 3으로 다뤄진 데이터를 지웁니다.

  5는 책임 테이블에서 통과된 UID를 지웁니다.

  5를 지정하는 경우 UID를 포함하는 텍스트 파일의 전체 경로를 제공합니다.

  6은 지우기 플로를 취소합니다.

  7은 스크래치 테이블 레코드를 정리합니다. 여기서 마지막으로 저장된 날짜는 구독의 마지막으로 처리된 날짜 이전입니다.

  실수로 색인이 지워지지 않도록 **복제 준비 완료**로 설정된 환경에서 **-task=objdata:clear**를 실행하는 것이 실패하게 됩니다.

© 2024 Siemens

예제:

UID 리스트를 지우려면 다음 명령을 실행합니다.

```
runTcFTSIndexer -task=objdata:clear
```

프롬프트에서

```
위의 리스트[1-7]에서 명확한 옵션을 제공하십시오.
```

5를 입력합니다. 프롬프트는 다음을 반환합니다.

```
입력 UID 텍스트 파일에 대한 전체 경로를 제공합니다.
```

전체 경로 및 파일 이름을 입력합니다.

```
c:\my_uid_directory\uids.txt
```

- **objdata:errors**

  색인화 실패에 대한 오류를 지정된 디렉터리로 반환합니다. 디렉터리가 비어 있어야 합니다. 반환할 오류의 수를 선택적으로 지정할 수 있습니다. 기본값은 50개의 오류입니다.

  오류 디렉터리에는 실패한 개체의 각 UID에 대한 디렉터리가 포함되어 있습니다. 각 UID 디렉터리에는 다음 정보가 포함되어 있습니다.

  - 개체 이름 및 유형과 같은 속성은 실패한 개체에 대한 정보를 제공합니다.
  - 디버깅할 TC XML 및 Solr XML 파일이 생성되고 저장됩니다.
  - Syslog 정보입니다.
  - 오류가 있는 **TcFTSIndexer** 로그 파일입니다.

  이 예제에서는 처음 100개의 오류가 지정된 출력 디렉터리로 전송됩니다.

  ```
  runTcFTSIndexer -task=objdata:errors d:\TcFTSIndexer\errors 100
  ```

- **objdata:index**

  기존 색인을 지우지 않고 *TC_ROOT***TcFTSIndexer\conf\TcFTSIndexer_objdata.properties**에 지정된 기간 데이터를 색인화합니다.

  깔끔한 시작을 위해 먼저 **objdata:clear**를 사용해 색인화된 데이터와 캐시된 파일을 정리합니다. 그러나 색인을 실수로 지우는 것을 방지하기 위해 **복제 준비 완료** 환경에서의 실행은 실패하게 됩니다.

© 2024 Siemens

- **objdata:indexclone**

  기존 색인을 지우지 않고 지정된 시간 기간 동안 색인화합니다. 시작 시간은 **복제 준비 완료** 상태가 설정된 시기로 설정됩니다. 끝 시간은 **conf\objdata.properties** 파일에 설정됩니다.

  이 색인화 플로는 복제 및 병합 색인화 방법을 지원하며, 이를 통해 복제된 환경을 다시 색인화한 다음 업데이트된 생산 환경으로 병합하여 업그레이드 및 패치를 보다 효율적으로 관리할 수 있습니다.

  objdata:indexclone 타스크는 복제된 환경이 색인화되는 동안 생산 환경에서 발생한 색인화 변경 내용의 델타를 선택합니다. 이 플로는 **관리 플로 작업**에 따라 달라집니다.

- **objdata:indexlargeload**

  기존 색인을 지우지 않고 *TC_ROOT***TcFTSIndexer\conf\TcFTSIndexer_objdata.properties**에 지정된 기간 데이터를 청크 단위로 조회, 단계화 및 색인화합니다.

  Java 힙 제한으로 인해 색인기에 문제가 발생할 수 있는 대용량 데이터를 보다 효율적으로 처리하기 위해 이 플로를 사용할 수 있습니다.

- **objdata:indexsyncfailures**

  수동 개입이 필요한 색인 동기화 실패입니다.

- **objdata:recover**

  실패한 색인화된 개체를 복구합니다.

  초기 색인화 중에 실패가 보고되는 경우 초기 색인화가 완료된 후 복구 흐름을 실행합니다. 실패가 보고되지 않거나 일관적으로 실패가 발생해 수정이 필요할 때까지 **objdata:recover**를 반복 실행해 초기 색인화에서 실패를 복구합니다.

  오류를 수정하거나 나중에 수정하기 위해 그대로 두고 동기화 흐름을 계속 진행할 수 있습니다. 그대로 두는 경우, 이러한 오류는 동기화 중에 실패로 기록됩니다. 이런 경우 나중에 되돌아와서 문제를 해결할 수 있습니다. 복구 플로 작업은 시간 조각별로 작동합니다. 복구 플로 작업은 실패한 모든 개체를 함께 처리합니다.

- **objdata:recoverlargeload**

  실패한 색인화된 개체를 복구합니다.

  indexlargeload 플로 작업 중에 오류가 보고된 경우 **indexlargeload** 작업이 완료된 후 **objdata:recoverlargeload** 플로 작업을 실행합니다. 실패가 보고되지 않거나 일관적으로 실패가 발생해 수정이 필요할 때까지 **objdata:recoverlargeload**를 반복 실행해 **indexlargeload** 작업에서 실패를 복구합니다.

  오류를 수정하거나 나중에 수정하기 위해 그대로 두고 동기화 흐름을 계속 진행할 수 있습니다. 그대로 두는 경우, 이러한 오류는 동기화 중에 실패로 기록됩니다.

© 2024 Siemens

- **objdata:show**

  지정된 색인화 상태인 개체들을 두 개의 텍스트 파일로 반환합니다. **-task=objdata:show** *n uid_file_dir* 형식을 사용하여 출력 파일에 대한 상태 코드와 디렉터리를 지정합니다.

  상태를 지정하는 **show** 번호는 **1**, **2** 또는 **3**입니다.

  1은 복제 보류 상태에서 개체를 반환합니다.

  2는 색인화 완료 상태에서 개체를 반환합니다.

  3은 색인화 실패 상태에서 개체를 반환합니다.

  이 예제에서는 출력 파일에 색인화에 실패한 개체가 포함되어 있습니다.

  ```
  runTcFTSIndexer -task=objdata:show 3 d:\TcFTSIndexer\uid_output
  ```

  출력은 다음 두 개의 텍스트 파일을 반환합니다.

  - **uid.txt**

    지정된 상태와 일치하는 개체의 UID를 포함합니다. 각 UID는 하나의 행에 있습니다.

    objdata:sync *uid_file_dir*을 사용해 개체를 동기화할 수 있습니다. 예를 들어, **uid.txt** 파일이 포함된 디렉터리를 지정합니다.

    ```
    runTcFTSIndexer.bat -task=objdata:sync d:\TcFTSIndexer\uid_output
    ```

  - **uid_prop.txt**

    예를 들어 출력 형식 *UID*|*object_string*|*object_type*에 UID 포함:

    ```
    TRa3S5qdSDyB | Breaker Panel Anchor Plate/AP02-A | Physical Part Revision
    ```

- **objdata:sync**

  이전 실행과 현재 실행 사이에 발생한 데이터 변경 사항으로 색인을 업데이트합니다.

  - **sync** 작업은 **-interval=***seconds* 인수를 가질 수 있습니다.

    예를 들어, 다음과 같이 독립 실행형 색인기를 300초(5분)마다 사용해 개체 데이터를 동기화하는 경우입니다.

    ```
    runTcFTSIndexer -task=objdata:sync -interval=300
    ```

    이 값은 **0**보다 커야 합니다.

© 2024 Siemens

동기화를 한 번 실행하려면 **-interval=***seconds*를 생략합니다.

- **동기화** 작업은 **uid.txt** 파일에 대한 *파일 경로*를 취할 수 있습니다.

  objdata:sync *filepath*를 사용해 개체를 다시 동기화할 수 있습니다. 예를 들어, **uid.txt** 파일이 포함된 디렉터리를 지정합니다.

  ```
  runTcFTSIndexer.bat -task=objdata:sync d:\TcFTSIndexer\uid_output
  ```

- **objdata:test**

  색인기 실행 전에 환경이 올바르게 설정되었는지 확인합니다.

관리 플로 작업

- **admin:cloneready**

  복제된 환경을 생성할 준비를 위해 생산 환경을 **복제 준비 완료**로 설정합니다. 이 작업이 적용된 후 복제된 환경을 업그레이드 또는 패치하고 **objdata:index**를 사용하여 색인화할 수 있습니다.

- **admin:clonevalidate**

  업그레이드된 생산 환경 및 업그레이드된 복제된 환경의 색인화 정보가 동일한지 확인합니다.

  이 플로 작업을 실행하여 생산 환경이 색인화 시스템을 병합할 준비가 되었는지 확인합니다.

  병합 후 **objdata:index clone**을 사용하여 업그레이드된 생산 환경에서 색인화 변경 내용의 델타를 선택합니다.

- **admin:clonecomplete**

  업그레이드된 생산 환경 내 색인화 시스템에서 **복제 준비 완료** 설정을 제거합니다. 해당 환경은 이전에 **복제 준비 완료**로 설정되어 있어야 합니다.

- **admin:status**

  생산 환경의 현재 복제 상태를 확인하는 수단을 제공합니다.

다중 사이트 플로 작업

- **multisite:clear**

  기존 다중 사이트 게시 레코드에 대해 색인화된 데이터 및 캐시화된 파일을 지웁니다.

  multisite:index을 실행하기 전에 사용합니다.

© 2024 Siemens

- **multisite:index**

  기존 색인화된 다중 사이트 게시 레코드를 지우지 않고 *TC_ROOT***TcFTSIndexer\conf\TcFtsIndexer_multisite.properties** 파일에 지정된 기간에 대한 색인입니다. 예:

  ```
  runTcFTSIndexer -task:multisite:index
  ```

  색인화하는 동안 중복 게시 레코드가 제거됩니다.

  깔끔한 시작을 위해 **multisite:clear**를 사용해 다중 사이트 게시 레코드를 위한 색인화된 데이터와 캐시된 파일을 정리합니다.

- **multisite:indexuids**

  특정 개체 디렉터리 서비스(ODS) 사이트에 대한 UID를 색인화합니다.

  예를 들어, **uid.txt** 파일이 포함된 ODS 사이트 이름 및 디렉터리를 지정합니다.

  ```
  runTcFTSIndexer -task=multisite:indexuids ODS_Site_name D:\UID_dir
  ```

- **multisite:indexsyncfailures**

  수동 개입이 필요한 색인 동기화 실패입니다.

- **multisite:sync**

  이전 실행과 현재 실행 사이에 발생한 게시 레코드 데이터 변경 사항으로 색인을 업데이트합니다.

  동기화 작업은 **-interval=***seconds* 인수를 사용해 지정된 간격에서 **동기화** 작업을 반복합니다. 예를 들어, 다중 사이트 게시 레코드 데이터를 300초마다(5분) 동기화하려면:

  ```
  runTcFTSIndexer -task=multisite:sync -interval=300
  ```

  이 값은 **0**보다 커야 합니다.

  multisite:sync를 한 번 실행하려면 **-interval=***seconds*를 생략합니다.

- **multisite:recover**

  실패한 색인화된 게시 레코드 개체를 복구합니다.

  색인화 중에 일부 게시 레코드가 실패하는 경우 다중 사이트 색인화가 완료된 후 복구 플로를 실행합니다. 실패가 보고되지 않거나 일관적으로 실패가 발생해 수정이 필요할 때까지 **multisite:recover**를 반복 실행해 실패를 복구합니다.

© 2024 Siemens

- **multisite:test**

  다중 사이트 색인기 실행 전에 환경이 올바르게 설정되\돼는지 확인합니다.

구조 플로 작업

- **structure:recoverfailures**

  상태가 오류로 되어 있는 모든 제품 구성을 **ReadyToIndex** 상태 또는 **MarkedForDeletion** 상태로 변경합니다. 예:

  ```
  runTcFTSIndexer -task=structure:recoverfailures
  ```

- **structure:reset** *product-configuration-UID*

  제공된 제품 구성 UID 설정을 **ReadyToIndex** 상태 또는 **MarkedForDeletion** 상태로 재설정합니다. 예:

  ```
  runTcFTSIndexer -task=structure:reset 'goZRkWxoqd$DyB'
  ```

- **structure:resetall**

  최신 변환 및 스키마 파일을 다운로드하고, 모든 활성 제품 구성을 **ReadyToIndex** 상태로 재설정하고, 모든 삭제된 제품 구성을 **MarkedForDeletion** 상태로 재설정합니다. 예:

  ```
  runTcFTSIndexer -task=structure:resetall
  ```

- **structure:show**

  다음 예와 같이, 모든 구성된 제품 구성의 요약 내용을 표시합니다.

  ```
  runTcFTSIndexer -task=structure:show
  ```

- **structure:sync**

  다음 예와 같이, 동기화를 큐에 대기시키고 모든 제품 구성에 대한 작업을 삭제합니다.

  ```
  runTcFTSIndexer -task=structure:sync
  ```

- **structure:syncone** *product-configuration-UID*

  다음 예와 같이, 동기화를 큐에 대기시키고 단일 제품 구성 UID에 대한 작업을 삭제합니다.

  ```
  runTcFTSIndexer -task=structure:syncone goZRkWxoqd12DyB
  ```

- **structure:test**

  색인기를 실행하기 전 환경이 올바르게 설정됐는지 확인합니다. 예:

© 2024 Siemens

```
runTcFTSIndexer -task=structure:test
```

4GD 플로 작업

4GD 제품 구조는 처음에 **bomindex_admin** 유틸리티를 사용해 색인화 됩니다.

- **fourgd:reset** *product-configuration-UID*

  해당 4GD 제품 구성 UID 설정을 **ReadyToIndex** 상태로 재설정 합니다. 예:

  ```
  runTcFTSIndexer -task=fourgd:reset goZRkWxoqd$DyB
  ```

- **fourgd:resetall**

  최신 변환 및 스키마 파일을 다운로드하고 모든 활성 4GD 제품 구성을 **ReadyToIndex** 상태로 재설정합니다. 예:

  ```
  runTcFTSIndexer -task=fourgd:resetall
  ```

- **fourgd:show**

  다음 예와 같이, 모든 4GD 구성된 제품 구성의 요약을 표시합니다.

  ```
  runTcFTSIndexer -task=fourgd:show
  ```

- **fourgd:sync**

  다음 예와 같이, 동기화를 큐에 대기시키고 모든 4GD 제품 구성에 대한 작업을 삭제합니다.

  ```
  runTcFTSIndexer -task=fourgd:sync
  ```

- **fourgd:syncone** *product-configuration-UID*

  다음 예와 같이, 동기화를 큐에 대기시키고 단일 4GD 제품 구성 UID에 대한 작업을 삭제합니다.

  ```
  runTcFTSIndexer -task=fourgd:syncone goZRkWxoqd12DyB
  ```

  ```
  ./runTcFTSIndexer.sh -task=strcture:syncone 'Abed$b8dBhGXHA'
  ```


© 2024 Siemens

awindexerutil

색인화된 개체를 변경하고 동기화 플로를 통해 해당 개체에 대한 색인을 새로 고치려는 경우 해당 개체를 새로 고칩니다. **awindexerutil** 유틸리티는 다음 동기화 흐름 일괄 처리 중에 이러한 개체가 선택되도록 표시합니다. 이렇게 하면 전체 색인 흐름의 중단 없이 색인을 업데이트할 수 있습니다.

완료된 마지막 동기화 이후의 변경 델타에 대해서만 색인을 새로 고칠 수 있습니다. 델타 업데이트를 수행하여 추가, 수정 또는 제거된 유형 및 속성에 대한 변경 사항을 색인화할 수 있습니다.

awindexerutil을 실행하면 현재 동기화 흐름을 방해하지 않습니다.

TC_ROOT\bin 디렉터리에서 유틸리티를 실행합니다 (설정된 경우 *TC_BIN*).

구문

awindexerutil -u=user-id -p=password -g=group [**-refresh**] [**-delta** [**-dryrun**] [**-daterange**]] **-h**

인수

-u

사용자 ID를 지정합니다. 사용자는 관리 권한이 있어야 합니다.

> 주:
>
> 서버에 대해 보안 서비스 싱글 사인-온 (SSO)이 활성화되어 있는 경우 user 및 password 인수가 Teamcenter 데이터베이스에 대해서가 아닌 SSO를 통해 외부적으로 인증됩니다. 이러한 인수를 제공하지 않으면 유틸리티는 기존 SSO 세션에 가입을 시도합니다. 세션을 찾을 수 없는 경우 사용자 ID와 암호를 입력하라는 메시지가 표시됩니다.

-p

암호를 지정합니다.

-g

사용자와 연결된 그룹을 지정합니다.

-새로 고침

일괄 처리 크기에 따라 동기화 흐름을 시작합니다.

-delta

완료된 마지막 동기화 이후의 개체 유형 및 속성 변경 델타에 대해 색인을 업데이트합니다. **-delta -dryrun**을 실행해 변경 사항을 평가한 후 **-delta** 색인 업데이트를 실행하십시오.

하려면 이 방법을 사용합니다.

-dryrun

© 2024 Siemens

-delta와 함께 사용하면 개체 유형 및 속성의 변경 내용을 비교해 델타를 다시 색인화할 수 있습니다. 색인화는 수행되지 않습니다. 차이는 명령 윈도우 및 로그 파일에 출력됩니다. 오퍼레이션이 완료된 후에는 *.log* 파일 경로가 반환됩니다.

-daterange

-delta 또는 **-delta -dryrun**과 함께 사용해 업데이트된 스키마에서 색인화할 수 있는 개체 유형 및 속성에 대한 날짜 범위를 설정합니다. 날짜 범위를 설정하려면 다음과 같이 년-월-일 형식을 사용하십시오.

YYYY/mm/dd HH:MM:SS-YYYY/mm/dd HH:MM:SS

시간 (**HH:MM:SS**)은 선택사항입니다. 쉼표로 구분된 리스트에 여러 범위를 지정할 수 있습니다. 지정된 날짜 범위에 공백이 있는 경우 전체 사양을 따옴표로 묶으십시오.

-classification

-delta 인수와 함께 사용해 스키마 파일을 평가하며 다음 예정된 색인화 동기화 실행 시 색인화용 분류 데이터의 변경 사항을 표시합니다. **awindexerutil** 실행 시 색인으로 표시될 변경 리스트를 받으려면 **-classification -delta -dryrun** 인수로 유틸리티를 실행합니다.

-h

도움말 정보를 표시합니다.

예

- 데이터 개체를 새로 고치는 동기화 플로를 시작합니다.

```
awindexerutil -u=admin -p=pwd -g=group -refresh
```

- 색인화 변경 델타의 연습 실행 비교를 시작합니다. 신규, 수정 및 삭제된 유형과 속성의 모든 인스턴스가 식별됩니다. 새로 색인화된 유형의 경우, 지정된 날짜 범위 (2012/12/12 - 2016/12/31) 동안 변경된 인스턴스만 포함됩니다. 보고서만 반환되며 색인화 변경은 수행되지 않습니다.

```
awindexerutil -u=admin -p=pwd -g=group -delta -dryrun
   -daterange=2012/12/12-2016/12/31
```

- 변경 사항의 델타에 대한 재색인화를 시작합니다. 새로 만들어지고 수정되고 삭제된 모든 유형과 속성을 업데이트합니다.

```
awindexerutil -u=admin -p=pwd -g=group -delta
```

- 변경 사항의 델타에 대한 재색인화를 시작합니다. 새로 만들어지고 수정되고 삭제된 모든 유형과 속성을 업데이트합니다. 새로 추가된 유형의 경우, 지정된 날짜 범위에서 마지막으로 수정된 개체만 색인화됩니다.

```
awindexerutil -u=admin -p=pwd -g=group -delta -daterange=
   ”2015/12/12 15:30:00 - 2015/12/31 22:00:00, 2016/02/01 07:00:00
   - 2016/12/31 14:00:00”
```


© 2024 Siemens

- 색인화를 위해 분류 데이터의 변경 사항을 표시합니다.

```
awindexerutil -u=admin -p=pwd -g=group -delta -classification
```


© 2024 Siemens

aw_search_config_manager

비동기화 파일 내용 색인화 사용자 정의 시 사용되는 매핑 구성을 관리합니다. 매핑 구성은 CAD 파일과 같은 비표준 파일에서 추출된 데이터를 Solr로 색인화하는 방법을 지정합니다.

TC_ROOT\bin 디렉터리에서 유틸리티를 실행합니다. (또한 구성된 경우 *TC_BIN* 디렉터리를 사용할 수 있습니다.)

구문

aw_search_config_manager **-u=***user-ID* {**-p=***password* | **-pf=***password-file*} **-g=***group*
[**-import -dir=** *config-directory*]
[**-export -config_id=** *config-id* **-output=** *output-file-path*]
[**-remove -config_id=** *config-id*]
[**-list**]
[**-h**]

인수

-u

사용자 ID를 지정합니다. 사용자에게 관리 권한이 있어야 합니다.

> 주:
>
> 서버에 대해 보안 서비스 싱글 사인-온 (SSO)이 활성화되어 있는 경우 user 및 password 인수가 Teamcenter 데이터베이스에 대해서가 아닌 SSO를 통해 외부적으로 인증됩니다. 이러한 인수를 제공하지 않으면 유틸리티는 기존 SSO 세션에 가입을 시도합니다. 세션을 찾을 수 없는 경우 사용자 ID와 암호를 입력하라는 메시지가 나타납니다.

-p

사용자 암호를 지정합니다. 이 인수는 **-pf** 인수와 함께 사용할 수 없습니다.

-pf

암호 파일을 지정합니다. 이 인수는 **-p** 인수와 함께 사용할 수 없습니다.

-g

사용자와 연결된 그룹을 지정합니다.

-import

지정된 디렉터리에서 구성 파일을 가져옵니다.

-dir

-import와 함께 사용하여 구성 파일을 가져올 디렉터리를 지정합니다.

-export

지정된 디렉터리에서 구성 파일을 가져옵니다.

© 2024 Siemens

-output

-export와 함께 사용하여 JSON 구성을 저장할 파일 이름을 포함한 전체 경로를 지정합니다.

-remove

매핑 구성을 제거합니다.

-config_id

- **-export**와 함께 사용하면 내보낼 구성의 ID를 지정합니다.
- **-remove**와 함께 사용하면 제거할 구성의 ID를 지정합니다.

-list

저장된 구성 ID의 리스트를 표시합니다.

-h

도움말 정보를 표시합니다.

예

예를 들어, 매핑 구성 파일 *C:\mappingConfigs\extConfig.json*에는 ID가 *extConfig*인 매핑 구성이 포함되어 있습니다. 이 ID(*extConfig*)에는 필드 내용 변환 프로세스가 외부 소스에서 추출한 데이터를 색인화 및 검색할 수 있는 형식으로 변환하도록 하는 규칙이 포함되어 있습니다.

다음 명령을 실행하여 *C:\mappingConfigs* 디렉터리에 있는 모든 구성을 업로드합니다.

```
aw_search_config_manager -u=admin -p=pwd -g=dba -import -dir=c:\mappingConfigs
```

extConfig.json 내부에 있는 구성을 이제 변환 프로세스에서 사용할 수 있습니다.

- 사용 가능한 모든 구성 나열:

  ```
  aw_search_config_manager -u=admin -p=pwd -g=dba -list
  ```

- JSON 파일로 구성 내보내기:

  ```
  aw_search_config_manager -u=admin -p=pwd -g=dba -export -config_id=extConfig
  -output=c:\exportedConfigs\extConfig.json
  ```

- 기존 구성 제거:

  ```
  aw_search_config_manager -u=admin -p=pwd -g=dba -remove -config_id=extConfig
  ```


© 2024 Siemens

색인화 결과 검토

색인기 및 검색 관리 대시보드를 사용하여 색인화 상태 보기

*색인기 및 검색 관리 대시보드*를 사용하여 개체 데이터 및 파일 내용 색인화 상태를 볼 수 있습니다.

색인기 및 검색 관리 대시보드 타일을 클릭하여 **활성 관리** 작업 영역의 대시보드에 액세스할 수 있습니다.

요약에서 색인화 개요 정보 검사

요약에서 색인기 및 색인화된 개체에 대한 개요 정보를 검토할 수 있습니다.

건강 상태	색인기 및 색인화 엔진(Solr)이 기능하는 방법에 대한 정보를 제공합니다.
색인화 상세 정보	최근 색인화 및 서브스크리싱 응용 프로그램에서 새로 생성되고 새로 삭제된 데이터에 대한 정보를 제공합니다. 개체의 추가 또는 지우기 구독을 했지만 최근에 처리하지 않은 응용 프로그램은 동기화가 느려질 수 있습니다.
개체 데이터 색인화 상태	가장 최근의 색인화 동기화에서 **통과**, **보류** 및 **실패함** 개체에 대한 개수를 제공합니다. 해당 상태의 처음 50개 개체에 대한 상세정보를 보려면 옵션을 클릭합니다.
파일 내용 색인화 상태	가장 최근의 색인화 동기화에서 **통과**, **보류** 및 **실패함** 파일 내용에 대한 개수를 제공합니다. 해당 상태의 처음 50개 개체에 대한 상세정보를 보려면 옵션을 클릭합니다.

진단에서 색인화 상세정보 검사

진단에서 특정 개체 데이터에 대한 색인화 상태를 검토할 수 있습니다. 조회를 선택하고 특정 검색 조건을 입력한 후 **색인 상태 가져오기**를 클릭합니다.

개체 데이터 색인화 상태	각 색인화 상태에 대한 개수가 있는 **상태 요약**을 제공합니다. • **통과**: 개체가 성공적으로 색인화됩니다. • **보류**: 개체가 색인화 대기 중입니다. • **실패**: 개체를 색인화할 수 없습니다. • **색인화되지 않음**: 개체가 아직 색인화 대기열에 없는 경우 또는 색인화에 선택되지 않습니다. **결과** 섹션에서 해당 상태의 개체 데이터를 표시하려면 옵션을 선택합니다.

© 2024 Siemens

결과

색인화 상태 조회의 결과를 제공합니다. 처음에는 조회의 모든 결과가 표시됩니다. **색인화 상태**에서 상태 옵션을 선택하면 해당 상태의 결과만 표시됩니다.

설정 ⚙을 클릭하여 결과에서 상세정보를 구성합니다. **텍스트 래핑**을 선택하여 긴 설명이 있는 열의 너비를 줄이거나 **정렬**을 선택하여 열 정렬 옵션을 표시하고 정렬을 저장할 수 있습니다.

↺ **재설정**

열을 기본 순서로 반환합니다.

위로 이동

열 이름을 클릭한 다음 **위로 이동** ⤒을 클릭하여 열 순서를 변경합니다.

아래로 이동

열 이름을 클릭한 다음 **아래로 이동** ⤓을 클릭하여 열 순서를 변경합니다.

열 정렬

저장된 열 정렬 중 하나를 선택하여 뷰에 적용합니다.

› **추가**

사용 가능한 열 리스트에서 테이블 열을 선택합니다. 그런 다음 **추가** ›를 클릭하여 표시된 열 리스트에 추가합니다.

‹ **제거**

표시된 열 리스트에서 테이블 열을 선택합니다. 그런 다음 **제거** ‹를 클릭하여 표시된 열 리스트에서 제거합니다.

색인화 관련 로그 찾기

로깅은 수행 중인 서로 다른 유형의 오퍼레이션에 대해 여러 위치에서 사용할 수 있습니다. 다양한 로깅 사양을 설정하여 오류 및 문제를 디버그할 수 있습니다.

> 주:
>
> 성능 문제가 발생하지 않도록 하려면 로깅 수준 및 변수를 디버깅 완료 후 기본 설정으로 복구합니다. 문제 해결을 마친 후에는 시스템을 다시 시작해야 합니다.

TcFTSIndexer 로그

이러한 로그는 *TC_ROOT***TcFTSIndexer\logs** 디렉터리에 있습니다. 로그에는 개체 데이터 색인화 및 구조 데이터 색인화에 대한 메시지가 포함되어 있습니다. 이러한 로그는 지원되는 모든 **TcFTSIndexer** 유형의 로그를 포함하는 색인기 및 색인기 프레임워크에 대한 기본 정보도 제공합니다.

© 2024 Siemens

- *TcFTSIndexer.log*에는 프레임워크 및 모든 TcFTSIndexer 플로와 관련된 모든 메시지가 포함됩니다.
- *TcFTSIndexer_objdata.log*에는 개체 색인화 플로와 관련된 개체 데이터 메시지가 포함됩니다. 여기에는 동기 파일 내용 색인화 플로를 사용할 때 색인화된 파일과 관련된 메시지도 포함됩니다.
- *TcFTSIndexer_filecontent.log*에는 비동기 파일 내용 색인화 플로를 사용할 때 색인화된 파일과 관련된 메시지도 포함됩니다.
- *TcFTSIndexer_structure.log* 구조 데이터 색인화 플로와 관련된 메시지를 포함합니다.

로깅 수준을 변경하여 디버그 로깅을 설정할 수 있습니다.

1. *TC_ROOT***TcFTSIndexer****conf***log4j.properties* 파일을 열 수 있습니다.
2. **DEBUG** 로깅 수준을 설정합니다.

   ```
   log4j.logger.com.siemens.teamcenter.ftsi=DEBUG
   ```

3. **TcFTSIndexer**를 다시 시작합니다.

tcserver 로그

TcFTSIndexer는 색인 데이터에 대한 서버 호출을 만드는 Java SOA 클라이언트입니다. SOA 호출에 실패하면 **tcserver** 및 **TcFTSIndexer**에서 실패에 대한 제한된 메시지를 제공합니다. 오류를 조사하는 경우 다음을 수행하십시오.

- 로깅 수준을 **DEBUG**로 설정하고 오류를 유발하는 사용 사례를 실행합니다. 이전에 *TcFTSIndexer 로그*에서 설명한 대로 지침을 따르십시오.
- **tcserver syslog** 파일을 확인합니다. 서버측 실패에 대한 자세한 내용은 SOA 호출과 연결된 **syslog** 파일에서 확인하십시오.

  syslog 파일은 *Temp* 디렉터리에 있습니다. 해당 위치에 파일이 없으면 **TC_KEEP_SYSTEM_LOG**가 true로 설정되어 있는지 확인하십시오.

  - 디버그 로깅의 경우 다음 파일을 엽니다.

    ```
    TC_LOGGER_CONFIGURATION\logger.properties
    ```

    다음에 대한 로깅 수준을 **DEBUG**로 변경합니다.

    ```
    logging.rootLogger
    logging.logger.Teamcenter
    logging.logger.Teamcenter.Soa.Communication
    ```

  - SQL 디버그 로깅의 경우, 환경 변수를 설정할 수 있습니다.

© 2024 Siemens

1. *tc_profilevars* 파일을 엽니다.
2. **TC_SQL_DEBUG=PT**를 설정합니다.
3. 풀 관리자 및 **tcserver**를 다시 시작합니다.
4. **syslog** 파일에서 **DEBUG** 메시지를 확인합니다.

- **저널** 파일을 검사하여 **tcserver**의 낮은 수준 해석을 가져옵니다. 다음 환경 변수를 설정하여 호출된 모든 방법을 추적할 수 있습니다.

  ```
  TC_JOURNAL=FULL
  TC_JOURNALLING=ON
  TC_JOURNAL_LINE_LIMIT=0
  ```

- *.pjl* 파일을 확인합니다. 환경 변수를 설정하여 성능 피드백을 가져올 수 있습니다.

  ```
  TC_JOURNAL_PERFORMANCE_ONLY=true
  ```

  풀 관리자 및 **tcserver**를 다시 시작합니다.

  이들 파일은 매우 커질 수 있으므로 조사 중인 사용 사례만 실행하십시오. 저널 로깅은 해석하기 어려울 수 있는 대량의 데이터를 생성합니다. **TC_JOURNAL_PERFORMANCE_ONLY** 환경 변수를 재설정하여 정상 오퍼레이션을 복원하십시오.

Solr 로그

Solr에 오류가 있는 경우 *TC_ROOT***solr**-version**server\logs***solr.log**에 로그를 찾습니다.

TcFTSIndexer 개체 데이터 스테이징 로그

스테이징 로그 파일에는 데이터 문제에 대한 정보가 포함됩니다. 이들 파일은 다음 위치에 있습니다.

Staging_Directory\TcFTSIndexer_objdata

스테이징 디렉터리는 다음에 있는 **Staging.Dir** 속성에 의해 정의됩니다.

TC_ROOT\TcFTSIndexer\conf\TcFTSIndexer.properties

기본적으로 스테이징 디렉터리 위치는 다음과 같습니다.

TC_ROOT\TcFTSIndexer\working\TcFTSIndexer_objdata

- *Tcxml*.xml* 파일에는 Teamcenter에서 내보낸 낮은 수준의 TC XML 가져오기/내보내기 데이터가 포함되어 있습니다.

© 2024 Siemens

- *Solr_pool_*.xml* 파일에는 Solr로 로드할 Solr 형식의 내보낸 데이터가 포함되어 있습니다.

기본적으로 오류가 있으면 생성된 스테이징 파일이 삭제되지 않습니다. 오류가 없으면 해당 파일을 그대로 유지할 수 있습니다.

*TC_ROOT\TcFTSIndexer\conf\TcFTSIndexer.properties*를 열고 다음을 설정합니다.

```
Indexer.keepGeneratedFiles=always
```

syslog 파일의 TcFTSIndexer 개체 데이터 색인화 오류

syslog 파일 또는 *TcFTSIndexer_objdata.log*에서 개체 데이터 색인화 오류를 찾습니다.

- **runTcFTSIndexer -debug**를 실행하여 **TcFTSIndexer syslog** 파일 리스트를 가져옵니다.
- *TC_ROOT***\TcFTSIndexer\logs***TcFTSIndexer_objdata.log*를 엽니다.

  ERROR -가 포함된 구문을 검색합니다. 다음과 같은 **syslog** 파일 위치를 찾으십시오.

예제:

```
2017-02-09 01:55:19,977 ERROR - Please see the log file
     'C:\tc_logs\tcserver.exe59d00782.syslog'
     on the server 'xyz' for more information.


    2017-02-09 01:55:19,978 ERROR - The request XML:
    <?xml version="1.0" encoding="UTF-8"?>
    <ExportObjectsInput tcGSMessageId="TCpool-1-thread-104_196181486623316423"
         synchronize="false"
         xmlns="http://teamcenter.com/Schemas/Internal/GlobalMultiSite/2007-06/
    ImportExport"
    2017-02-09 01:55:19,978 ERROR - The response XML:
    2017-02-09 01:55:19,978 ERROR - SYSLOG FILE: C:\tc_logs\tcserver.exe59d00782.syslog



    2017-02-09 01:55:19,981 ERROR - TieExport failed. Files are located in
         D:\tc\tcroot\TcFTSIndexer\working\TcFtsIndexer_objdata\
         U159c8055a56130e1761114495\result
```

생성된 색인화 로그

각 색인화 오퍼레이션에는 타스크 ID (예: **U159c8055a56130e1761114495**)가 있습니다. 실패한 오퍼레이션에 대한 타스크 ID를 생성하면 *Staging_Directory\TcFTSIndexer_objdata*내 하위 디렉터리에서 생성된 파일을 찾을 수 있습니다.

오류가 있는 경우 오류 메시지를 통해 디렉터리 위치가 제공됩니다. 예:

```
2017-02-09 01:55:19,981 ERROR - TieExport failed. Files are located in
D:\tc\tcroot\TcFTSIndexer\working\TcFtsIndexer_objdata\U159c8055a56130e1761114495\result
```


© 2024 Siemens

오류가 없지만 생성된 파일을 계속 표시하려는 경우:

1. **색인화** 플로의 작은 시간대를 사용하거나, 색인화해야 하는 개체를 추가, 수정 또는 삭제한 후 **동기화** 플로를 실행하여 데이터의 양을 제한합니다.

2. *TC_ROOT***TcFTSIndexer\conf***TcFTSIndexer.properties*를 열고 다음을 설정하십시오.

   ```
   Indexer.keepGeneratedFiles=always
   ```

3. **색인화** 플로 또는 **동기화** 플로를 실행합니다.

4. 플로가 실행된 날짜 및 시간과 함께 *Staging_Directory_path***TcFTSIndexer_objdata**에서 생성된 파일을 찾습니다.

조회 오류 로그

runTcFTSIndexer -debug를 실행하여 **TcFTSIndexer syslog** 파일 리스트를 가져옵니다.
TcFTSIndexer_objdata.log 파일에서 **syslog** 파일 정보를 살펴보십시오.

SQL 디버그 로깅의 경우 *tc_profilevars* 파일을 엽니다.

1. **TC_SQL_DEBUG=PT**를 설정합니다.

2. 풀 관리자를 다시 시작합니다.

3. **syslog** 파일에서 **DEBUG** 메시지를 확인합니다.

오류 로그 내보내기

TcFTSIndexer -debug를 실행하여 **TcFTSIndexer syslog** 파일 리스트를 가져옵니다.
TcFTSIndexer_objdata.log 파일에서 **syslog** 파일 정보를 살펴보십시오.

변환 오류 로그

*TC_ROOT\TcFTSIndexer\logs\TcFTSIndexer_objdata.log*에서 오류를 찾습니다. 생성된 파일 디렉터리를 오류에서 참조하는 경우 이러한 로그 파일을 평가하십시오.

로드 오류 로그

*TC_ROOT***TcFTSIndexer\logs***TcFTSIndexer_objdata.log* 및 *TC_ROOT***solr-***version***\server\logs***solr.log* 파일에 오류를 찾습니다.

© 2024 Siemens

파일 내용 색인화 상태 검토

모든 데이터 집합 개체의 색인화 상태를 조회할 수 있습니다. 또한 데이터 집합 대신 부모 비즈니스 개체에 대해 파일 내용이 색인화될 때 부모 비즈니스 개체의 상태를 조회할 수 있습니다. 상태를 표시하려면 Teamcenter 데이터베이스의 **ACCT_TABLE**에서 다음 속성을 조회합니다.

이 상태의 개체를 나열하려면...	이 SQL 조회 실행...
보류 중인 데이터 집합	`SELECT island_anchor_uid FROM ACCT_TABLE WHERE app_id=':FTS_FILECONTENT' AND state=1`
성공적으로 색인화됨	`SELECT island_anchor_uid FROM ACCT_TABLE WHERE app_id=':FTS_FILECONTENT' AND state=2`
실패한 데이터 집합	`SELECT island_anchor_uid FROM ACCT_TABLE WHERE app_id=':FTS_FILECONTENT' AND state=4`

실패 검토 및 해결

관련 스테이징 및 로그 파일을 검토하여 **동기화** 플로 및 **복구** 플로에서 실패한 개체를 검사합니다.

1. *스테이징 디렉터리**TcFTSIndexer_objdata* 및 *TC_ROOT**TcFTSIndexer**logs*의 파일을 별도의 디렉터리로 복사합니다.

2. *TC_ROOT***TcFTSIndexer**\conf**TcFTSIndexer.properties*를 열고 다음을 설정하십시오.

   Indexer.keepGeneratedFiles=always

3. 흐름에 적합한 작업을 선택합니다.

   - 개체 데이터에 대해 **복구** 플로를 사용하는 경우 **runTcFTSIndexer -task=objdata:recover**를 실행합니다.
   - 보류 중인 데이터 집합에 대해 **복구** 플로를 사용하는 경우 **runTcFTSIndexer -task=objdata:filecontentrecover**를 실행합니다.
   - **동기화** 플로를 사용하는 경우 **runTcFTSIndexer -task=objdata:sync -interval=***seconds*를 실행합니다.

     완료 후 **runTcFTSIndexer -task=objdata:indexsyncfailures**를 실행하여 실패한 개체를 처리합니다.

4. 평가를 위해 TcFTSIndexer 생성 로그, syslog 파일 및 Solr 로그에 관련된 로그를 찾습니다.

5. 실패한 개체 리스트를 가져오려면 모든 개체 데이터를 처리하고 최종 상태(성공 또는 실패)가 된 후 **표시** 플로를 사용하여runTcFTSIndexer 유틸리티를 실행합니다. 개체 상태 및 출력 디렉터리에 대한 코드를 지정하십시오.

© 2024 Siemens

```
runTcFTSIndexer -task=objdata:show 3 d:\TcFTSIndexer\uid_output
```

이 예제에서 상태 **3**은 **실패한 색인화**를 지정하며 출력 디렉터리에는 개체를 나열하는 *uid.txt* 및 *uid_prop.txt* 파일이 포함되어 있습니다.

- *uid.txt*

  개체의 UID를 포함합니다. 다음과 같이 **objdata:sync** *uid_file_dir*을 사용하여 개체를 동기화할 수 있습니다.

  ```
  runTcFTSIndexer.bat -task=objdata:sync d:\TcFTSIndexer\uid_output
  ```

- *uid_prop.txt*

  UID|*object_string*|*object_type* 출력 형식으로 UID를 포함합니다. 예를 들면 다음과 같습니다.

  ```
  TRa3S5qd$DyB | Breaker Panel Anchor Plate/AP02-A | Physical Part Revision
  ```

파일 내용 색인화 오류 해결

파일 내용 색인화 오류 해결

경우에 따라 색인화 프로세스 중에 파일을 색인화할 때 오류 메시지가 나타날 수 있습니다. 영향 받는 파일에 대해 다음 제안된 복구 절차를 사용하여 파일을 색인화할 수 있습니다. 복구할 수 있는 데이터의 양은 파일 및 특정 오류에 따라 다릅니다.

로그 파일 또는 색인화가 실행되는 명령 윈도우에서 오류 메시지를 볼 수 있습니다. 메시지의 값은 시스템에 따라 다를 수 있습니다.

텍스트 파일에 영향을 미치는 오류

다음 오류에 대해 **텍스트 복구 절차**를 수행하십시오.

```
Caused by: java.lang.IllegalStateException: Potential loop detected -
Block 0 was already claimed but was just requested again
```

PDF 파일에 영향을 미치는 오류

다음 오류의 경우 파일이 손상되었거나 암호로 보호되어 있어 복구 절차를 사용할 수 없습니다.

- `org.apache.pdfbox.pdmodel.encryption.InvalidPasswordException: Cannot decrypt PDF, the password is incorrect`
- `java.io.IOException: Missing root object specification in trailer`
- `java.io.IOException: Page tree root must be a dictionary`

© 2024 Siemens

- `java.lang.IllegalArgumentException: The end (65642) must not be before the start (65648)`

재저장 복구 절차

다음 오류가 발생하면 **PDF 파일 재저장 절차**를 수행하십시오.

- `java.lang.IllegalStateException: Expected 'Page' but found COSName{Font}`
- `Unknown dir object c='>' cInt=62 peek='>' peekInt=62 at offset 195699`
- `java.lang.IllegalStateException: Expected 'Page' but found COSName{Font}`
- `Caused by: java.io.IOException: Invalid font type: COSName{Font}`

이 절차를 수행한 후 *error-text* **in font** *font-name*에 대한 WARN No Unicode 매핑 오류를 수신할 수 있습니다. 이는 *error-text*를 색인화할 수 없음을 나타냅니다.

Microsoft PowerPoint, Excel 및 Word 파일에 영향을 주는 오류

다음 오류의 경우 파일이 Microsoft 보안 센터에 의해 차단되었기 때문에 사용할 수 있는 복구 절차가 없습니다.

```
org.apache.poi.poifs.filesystem.NotOLE2FileException: The supplied data appears to be an old Word version 2 file. Apache POI doesn't currently support this format
```

재저장 복구 절차

다음 오류가 발생하면 **Microsoft Office 파일 재저장 절차**를 수행하십시오.

- `Table Stream '0Table' wasn't found - Either the document is corrupt, or is Word95 (or earlier)`
- `Initialization of record 0x809(BOFRecord) left 4082 bytes remaining still to be read.`
- `This paragraph is not the first one in the table`
- `java.lang.ArrayIndexOutOfBoundsException`
- `java.lang.IllegalArgumentException: The end (65642) must not be before the start (65648)`

OOXML 파일 재저장 절차

© 2024 Siemens

다음 오류가 발생하면 **OOXML 파일 재저장 절차**를 수행하십시오.

```
Strict OOXML isn't currently supported, please see bug #57699
```

OOXML 관계 해결

다음 오류가 발생하면 **OOXML 파일 해결 관계**를 수행하십시오.

```
No part found for relationship id=rId3 -
container=org.apache.poi.openxml4j.opc.ZipPackage@6a539bca -
relationshipType=http://schemas.openxmlformats.org/officeDocument/
2006/relationships/oleObject - source=/ppt/slides/slide64.xml -
target=/ppt/slides/NULL,targetMode=INTERNAL
```

텍스트 복구

다음 오류가 발생하면 **텍스트 복구 절차**를 수행하십시오.

- `Caused by: java.lang.IllegalStateException: Potential loop detected - Block 0 was already claimed but was just requested again`
- `Caused by: java.io.IOException: The text piece table is corrupted, expected byte value 2 but had 0`

텍스트 파일 복구

다음 절차를 수행하여 텍스트 파일을 복구하고 오류를 해결합니다.

절차

1. 영향 받는 파일에 대한 Microsoft 응용 프로그램을 엽니다.
2. **열기→찾아보기**를 클릭합니다.
3. 영향 받는 파일의 위치로 이동합니다.
4. 파일 이름을 선택합니다.

   모든 파일 드롭다운에서 **모든 파일에서 텍스트 복구**를 선택합니다.
5. **열기**를 클릭합니다.

   복구 표시 윈도우가 열립니다.
6. **복구 표시** 윈도우를 닫습니다.

   이 절차에는 필요하지 않습니다.

© 2024 Siemens

7. 파일을 저장합니다.

8. 이름 변경된 파일을 Teamcenter에 추가하고 **runTcFTSIndexer** 유틸리티를 사용하여 이름 변경된 파일을 색인화합니다.

PDF 파일 다시 저장

다음 절차를 수행하여 PDF 파일을 다시 저장하고 오류를 해결합니다.

절차

1. 영향 받는 파일을 엽니다.

2. 표시되면 **지금 수정**을 클릭한 다음 5단계로 진행합니다.

   그렇지 않은 경우 3단계를 계속합니다.

3. **파일→다른 이름으로 저장**을 클릭합니다.

4. 파일의 이름을 변경합니다.

5. 파일을 저장합니다.

6. 이름 변경된 파일을 Teamcenter에 추가하고 **runTcFTSIndexer** 유틸리티를 사용하여 이름 변경된 파일을 색인화합니다.

Microsoft Office 파일 다시 저장

다음 절차를 수행하여 Microsoft Office 파일을 다시 저장하고 오류를 해결합니다.

절차

1. 각각의 Microsoft 응용 프로그램을 사용하여 영향 받는 파일을 엽니다.

2. **파일→다른 이름으로 저장→찾아보기**를 클릭합니다.

3. 파일의 이름을 변경합니다.

4. **파일 유형** 드롭다운 리스트에서 최신 Microsoft Office 파일 형식을 선택합니다.

   파일 확장자가 표시되면 *.docx*, *.pptx*, 또는 *.xlsx*와 같은 확장자를 선택합니다.

5. **저장**을 클릭합니다.

   파일이 최신 파일 형식으로 업그레이드되었음을 알리는 새 윈도우에 표시됩니다.

© 2024 Siemens

6. **확인**을 클릭합니다.

   파일의 레이아웃을 변경할 수 있습니다.

7. 이름 변경된 파일을 Teamcenter에 추가하고 **runTcFTSIndexer** 유틸리티를 사용하여 파일을 색인화합니다.

OOXML 파일 다시 저장

다음 절차를 수행하여 OOXML 파일을 다시 저장하고 오류를 해결합니다.

절차

1. 각각의 Microsoft 응용 프로그램을 사용하여 영향 받는 파일을 엽니다.

2. **파일→다른 이름으로 저장→찾아보기**를 클릭합니다.

3. 파일의 이름을 변경합니다.

4. **파일 유형** 드롭다운 리스트에서 최신 Microsoft Office 파일 형식을 선택합니다.

   파일 확장자가 표시되면 *.docx*, *.pptx*, 또는 *.xlsx*와 같은 확장자를 선택합니다.

5. **저장**을 클릭합니다.

6. 이름 변경된 파일을 Teamcenter에 추가하고 **runTcFTSIndexer** 유틸리티를 사용하여 이름 변경된 파일을 색인화합니다.

OOXML 파일 관계 해결

다음 절차를 수행하여 OOXML 파일 관계를 해결하고 오류를 해결합니다.

절차

1. 각각의 Microsoft 응용 프로그램을 사용하여 영향 받는 파일을 엽니다.

2. 오류 메시지에서 강조표시된 문제 관계로 이동합니다.

3. 문제 관계 아이템을 복사합니다.

4. 아이템을 그림으로 붙여넣습니다.

   마우스 오른쪽 버튼을 클릭하고 **붙여넣기→그림**을 선택합니다.

5. 모든 관계 문제 아이템에 대해 2단계에서 4단계까지 반복합니다.

© 2024 Siemens

6. 파일을 저장합니다.

7. 이름 변경된 파일을 Teamcenter에 추가하고 **runTcFTSIndexer** 유틸리티를 사용하여 이름 변경된 파일을 색인화합니다.

검색에 표시되지 않는 색인된 파일 내용 검토

검색에서 색인화된 파일 내용이 있는 데이터 집합을 반환하지 않는 경우 다음을 검증합니다.

- 데이터 집합 유형은 색인화 가능으로 표시됩니다.
- 파일 확장자 또는 명명된 참조는 색인화 가능한 파일 확장자 리스트에 포함됩니다.
- **비동기화 파일 내용 색인화가 활성화됩니다**.
- 색인기가 실행 중입니다.
- 모든 디스패처[6] 컴포넌트가 실행 중입니다.
  - Dispatcher Scheduler
  - 디스패처 클라이언트
  - Dispatcher Module

    적절한 서비스가 설치되어 있고 구성 파일 *Dispatcher_Root*/**Module/conf/***translator.xml*에서 활성으로 설정되어 있는지 확인하십시오.

색인화할 파일 유형...	**검색할 서비스...**
표준 파일(예: Microsoft Office 파일, PDF, 및 텍스트 파일)	**tcftsindexerfilecontenttikaindex**
NX	**tcftsindexerfilecontentnxindex**
Solid Edge	**tcftsindexerfilecontentsolidedgeindex**
기타 독점 파일	사용자 정의 서비스

- (선택 사항) 다음을 실행하여 데이터 집합에 대한 매핑 구성을 설치합니다.

```
aw_search_config_manager -u=admin -p=pwd -g=dba -list
```

Microsoft Office 파일, PDF 파일 및 텍스트 파일과 같은 표준 파일만 색인화하는 경우 매핑 구성 파일이 필요하지 않습니다.

6 Teamcenter 문서의 디스패처 소개를 참조하십시오.

© 2024 Siemens

- (선택 사항) 구성 파일 *Dispatcher_Root***Module****Translators****TcFTSIndexer****conf***ExtractorConfig.json*에는 추출기의 올바른 경로가 포함되어 있습니다.

  Microsoft Office 파일, PDF 파일 및 텍스트 파일과 같은 표준 파일만 색인화하는 경우 추출기가 필요하지 않습니다.

- 색인기 로그에서 오류를 확인하여 데이터 집합 개체가 성공적으로 색인화되었습니다.

표시되지 않는 파일 내용 패싯 검토

패싯이 올바르게 표시되지 않으면 다음을 확인하십시오.

- 매핑 구성 파일이 정확합니다. 매핑 사양 문서 *TC_ROOT*/**TcFTSIndexer/conf/specification/***Mapping_Specification.docx*을 참조하십시오.

  - 패싯 속성은 매핑 항목에서 **true**로 설정됩니다.

    이렇게 하면 패싯 속성에 대한 필터링이 활성화됩니다.

  - 표시 이름은 로케일에 대해 설정됩니다.

    로케일에 대해 표시 이름이 설정되어 있지 않으면 패싯 이름이 예상대로 표시되지 않을 수 있습니다.

  발견된 문제를 수정하고 **aw_search_config_manager** 유틸리티를 실행하여 매핑 구성 파일을 구성 저장소로 로드합니다.

- **AWS_asynchronous_file_content_indexing_enabled** 환경설정이 **true**로 설정됩니다.

색인화 성능 향상

타이밍 정보를 수집하고 해석할 수 있습니다. 또한 Teamcenter와 색인기 간의 연결을 평가할 수도 있습니다.

ACCT_TABLE 잠금 문제 또는 Microsoft SQL Server 데이터베이스의 동기화 지연

다음 제안 사항을 사용하여 색인화 중 **ACCT_TABLE** 잠금 문제를 방지하고 동기화 중 조회 성능 문제를 방지하십시오.

- *색인기 최적화* 섹션에 제안된 대로 **tc.maxConnections** 속성을 사용하여 Teamcenter 연결 수를 줄이십시오.

- 색인화 단계에서 **querytimesslicestep.maxQueryTimeSpan**의 값을 줄여 각 개체 데이터 조회에 대한 조회 시간 스팬을 줄입니다.

  기본값은 20000분 또는 약 2주입니다. 이 값을 낮추면 개별 시간 조회의 로드를 줄일 수 있습니다.

© 2024 Siemens

- 다음 제안은 수정된 개체를 식별하기 위해 실행되는 동기화 조회의 성능을 향상시킵니다.
  - **Index_Sync_Mssql_Hint** 환경설정을 **true**로 설정합니다.
  - Microsoft SQL Server의 호환성 모드를 **2012**로 설정합니다.

성능 정보 수집

초기 색인화에서는 각 오퍼레이션에 대한 기간 요약 정보를 제공합니다. 예를 들면 다음과 같습니다.

```
2017-03-01 18:23:07,358 INFO  - Step Summary
2017-03-01 18:23:07,358 INFO  -   objdataquerystep
2017-03-01 18:23:07,359 INFO  -      Status: Created: 0   Started: 0   Done: 189
Error: 0
2017-03-01 18:23:07,359 INFO  -      Total Time   68.32   Total Count 90899

2017-03-01 18:23:07,359 INFO  -   objdataexportstep
2017-03-01 18:23:07,359 INFO  -      Status: Created: 0   Started: 0   Done: 125
Error: 0
2017-03-01 18:23:07,359 INFO  -      Total Time 19092.57   Total Count 90899

2017-03-01 18:23:07,360 INFO  -   objdatasaxtransformstep
2017-03-01 18:23:07,360 INFO  -      Status: Created: 0   Started: 0   Done: 125
Error: 0
2017-03-01 18:23:07,360 INFO  -      Total Time  213.68   Total Count 90899

2017-03-01 18:23:07,360 INFO  -   objdataloadstep
2017-03-01 18:23:07,361 INFO  -      Status: Created: 0   Started: 0   Done: 125
Error: 0
2017-03-01 18:23:07,361 INFO  -      Total Time  133.21   Total Count 90899
```

runTcFtsIndexer -status -task=objdata:index를 실행하여 **색인화** 플로의 모든 오퍼레이션에 대한 성능 및 상태를 보고하는 경우 **-status** 인수를 추가할 수 있습니다.

runTcFTSIndexer -task=objdata:sync 플로를 실행하면 각 조회 횟수 및 조회하는 동안 발견된 개체의 색인화 횟수가 보고됩니다.

색인기 최적화

Teamcenter 및 색인기 사이의 연결은 **색인기를 설치**할 때 설정됩니다. **TcFTSIndexer 인스턴스를 최적화**하면 Teamcenter 및 색인기 간에 연결을 설정할 수 있습니다.

또한 **TcFtsindexer\conf\TcFtsIndexer.properties** 파일의 **Tc.maxConnections** 속성에서 연결의 최대 수를 조정할 수 있습니다.

serverPool.properties 파일의 **PROCESS_WARM** 매개변수를 사용하여 풀의 웜 서버 수를 조정할 수 있습니다. 자세한 내용은 Teamcenter 도움말의 *시스템 관리*를 참조하십시오.

© 2024 Siemens

환경설정 값 확인

AW_FTSIndexer_skip_modifications_via_relations 환경설정은 **TcFTSIndexer** 동기화 플로 동안 관계 조회를 실행하여 **TcFTSIndexer**가 수정 사항을 찾을 수 있는지 여부를 제어합니다. 기본값은 **true**이며, 이는 조회를 건너뜁니다. 환경설정 값이 **false**로설정되어 있는지 여부를 확인합니다. 예를 들어 **아이템에서 교체 ID가 색인화**된 경우 또는 **문서 리비전 유형의 개체에 첨부된 데이터 집합 개체**(또는 해당 하위 유형)가 이미 색인화된 경우, 환경설정 값을 **false**로 설정할 수 있습니다.

패치에서 업그레이드한 후 느린 동기화

시스템이 올바르게 구성되지 않으면 이전 패치 버전에서 Teamcenter 플랫폼을 업그레이드한 후, 개체 데이터 색인화의 첫 번째 동기화를 완료하는 데 오랜 시간이 걸릴 수 있습니다. 다음 이유로 인해 문제가 발생할 수 있습니다.

- 동기화 오퍼레이션은 **설치** 유틸리티의 **TC_TIMESTAMP_THRESHOLD** POM 매개변수에 의해 지정된 임계값에 의존합니다. 이는 **POM_timestamp** 테이블이 타임스탬프를 갖는 시간입니다. 설정되지 않은 경우 기본값은 96시간입니다.

  (개체 데이터 색인화는 서버에서 발생하는 TC XML 내보내기에 의존합니다.)

- 두 동기화 오퍼레이션 간의 시간이 임계값을 초과하는 경우(예: 업그레이드 동안 시스템이 다운될 경우):

  - POM 개체 테이블의 크기가 클 경우 동기화 논리는 POM 개체 테이블을 사용하여 오래 걸릴 수 있는 동기화 후보를 식별합니다.

  - 업그레이드의 경우, 두 동기화 오퍼레이션 간의 시간이 임계값을 초과하면 문제가 발생할 수 있습니다.

- 동기화 간의 시간이 임계값보다 적을 경우(예: 정상적인 동작 조건):

  - 더 작고 더 빨리 실행될 것으로 예상되므로 동기화 논리는 **POM_timestamp** 테이블을 사용합니다.

  - 일반적으로 동기화 시작 시 동기화 간격은 임계값보다 작은 300초로 설정됩니다. 따라서 문제가 발생하지 않습니다.

일시적으로 업그레이드 전보다 더 높은 임계값을 설정하여 업그레이드할 때 이 문제를 해결할 수 있습니다.

1. **install -ask_pom_param**을 실행하여 현재 값을 봅니다.

2. **install -set_pom_param**을 실행하여 더 높은 값을 설정합니다.

3. 첫 번째 동기화 오퍼레이션 이후 매개변수를 이전 값으로 재설정합니다.

© 2024 Siemens

설치 유틸리티를 실행하는 방법에 대한 지침은 Teamcenter 도움말의 *Teamcenter Utilities*를 참조하십시오.

색인화 사용자 정의

색인기 사용자 정의 개요

TcFTSIndexer:

- 유형, 플로 및 단계를 실행할 수 있는 Java 응용 프로그램입니다.
- 이는 Teamcenter에 연결하여 데이터를 추출하고 Solr에서 데이터를 색인화하는 SOA 클라이언트입니다.
- 모든 기존 단계 및 플로를 수정하여 고객의 요구 사항을 충족할 수 있습니다.
- 또한 이를 사용자 정의하여 외부 시스템 데이터를 추출하고 Solr에서 색인화를 수행할 수도 있습니다.
- 사용자 정의 단계에서 사용할 수 있는 유틸리티를 제공합니다.

  이러한 유틸리티에 대한 상세정보는 *TC_ROOT***TcFTSIndexer\docs\javadocs** 디렉터리의 Javadocs에서 제공됩니다.

TcFTSIndexer 설치 레이아웃

TcFTSIndexer 설치의 디렉터리 구조는 다음과 같습니다(*TC_ROOT***TcFTSIndexer**).

- **bin**

  TcFTSIndexer를 시작하는 스크립트가 있습니다.

- **cache**

  모든 관련 캐시 파일을 저장합니다.

- **conf**

  모든 구성 파일이 있습니다.

- **docs**

  게시된 API에 대한 Javadocs가 있습니다.

- **lib**

© 2024 Siemens

TcFTSIndexer를 실행하는 데 필요한 모든 JAR 파일이 있습니다.

- **logs**

  색인기 로그 파일이 있습니다.

- **sample**

  사용자 정의를 지원하는 샘플 파일이 있습니다.

TcFTSIndexer 확장성 프레임워크

TcFTSIndexer는 유형, 플로 및 단계를 실행하는 Java 응용 프로그램입니다.

- 유형

  서로 다른 통합 또는 사용자 정의 항목이 **TcFTSIndexer**에 표시됩니다(예: 개체 데이터 및 구조). 유형에는 플로가 포함됩니다.

- 플로

  특정한 유형에 대해 지원되는 오퍼레이션을 나타냅니다. 예를 들어, 개체 데이터에 대한 몇 가지 플로(**ObjData**)로는 지우기, 색인화, 복구 및 동기화를 들 수 있습니다. 플로에는 서로 연결된 단계가 포함됩니다.

- 단계

  특정 동작을 정의하는 메서드가 포함됩니다. 각 단계에는 정의된 입력 및 출력이 있어야 합니다. 이러한 단계는 입력 및 출력 요구사항을 기준으로, 다양한 플로 및 유형 전반에 걸쳐 재사용할 수 있습니다. 단계는 플로에 정의된 순서대로 실행됩니다. 한 단계의 출력은 다음 단계의 입력이 됩니다.

  예를 들어,개체 데이터 (**ObjData**) 색인화 플로에는 조회, TIE 내보내기, 변환, 로드 단계가 포함됩니다.

  단계의 종류는 다음 세 가지입니다.

  - 단순 단계

    입력을 기준으로 단계를 실행하고, 처리할 다음 단계에 대한 출력 데이터를 반환합니다.

  - 분할 단계

    입력 데이터 리스트를 리스트의 크기를 기준으로 이를 여러 단순 단계로 분할합니다. 개체 데이터(**ObjData**)의 경우 이 단계를 사용하여 조회를 시간 단위로 분할합니다. 플로에는 다중 분할 단계가 포함될 수 있습니다.

  - 집계 단계

© 2024 Siemens

모든 분할 단계가 완료될 때까지 기다리며, 분할 단계에서 처리되는 모든 출력 데이터를 통합합니다. 플로에서, 모든 분할 단계에는 연결된 집계 단계가 있어야 합니다.

색인기 필수항목 사용자 정의

- **TcFTSIndexer** 설치가 독립 실행형 모드에서 작동되어야 함.
- **TcFTSIndexer** 아키텍처에 대한 심도 있는 이해.
- 사용자 정의할 플로의 각 단계와 연결된 입력 및 출력 개체에 대한 이해.
- 플로와 연결된 속성에 대한 이해.
- *TC_ROOT***TcFTSIndexer\sample** 디렉터리의 단계에 대한 샘플 코드를 검토해야 함.
- 게시된 메서드 및 언급된 클래스에 대한 내용을 Javadocs에서 참조해야 함.
- *TC_ROOT***TcFTSIndexer\sample***TcFTSIndexer_sample1.properties* 파일 및 *TC_ROOT***TcFTSIndexer\conf***TcFTSIndexer_objdata.properties* 파일에서 예제 구성을 참조해야 함.
- 새 요구 사항의 경우:
  - 기능의 높은 수준 설계를 생성합니다.
  - 입력 및 출력 개체가 정의된 단계 리스트를 생성합니다. 재사용 가능한 기존 단계가 있는지 확인합니다.
  - 새 플로를 생성하거나 기존 플로를 수정하여 이러한 단계를 연결합니다.
  - 플로가 기존 유형에 속하는지 또는 새 유형에 속하는지 확인합니다.

새 검색 색인 유형 추가

유형은 모든 플로 및 단계 구성을 포함하는 속성 파일을 생성하여 정의됩니다. *TC_ROOT***TcFTSIndexer\conf** 디렉터리에서 이 속성 파일을 생성합니다. 파일 이름에는 **TcFTSIndexer_***TypeName***.properties** 구문이 있어야 합니다. **TcFTSIndexer**는 속성 파일 이름을 읽고 이러한 속성과 연결된 유형 이름을 확인합니다.

유형은 특정 동작이 필요한 경우를 제외하고 기본 동작을 사용합니다. 이 경우 유형 클래스는 **TcFtsIndexer_***TypeName***.properties** 파일에서 다음과 같이 정의할 수 있습니다.

```
type=full-class-name
```

예를 들면 다음과 같습니다.

```
com.siemens.teamcenter.ftsi.objdata.TcFTSIndexerObjDataType
```


© 2024 Siemens

이 사용자 정의 구현 클래스는 **TcFTSIndexerType** 클래스를 확장해야 하며 일반적으로 다음 메서드의 재정의가 필요합니다.

```
public void initialize( Object zData )
```

이 메서드는 유형을 실행하는 도중 호출되며, 이 유형에 특화된 개체의 초기화 또는 입력의 유효성 검사를 지원합니다. 예를 들어, 개체 데이터 유형 구현 클래스는 초기화 메서드를 재정의하여 언제든지 하나의 개체 데이터 유형만 실행되도록 합니다.

*TC_ROOT***TcFTSIndexer\conf\log4j.properties**에서 **objdata**와 유사한 사용자 정의 유형에 대한 로깅을 설정하십시오.

기존 검색 색인화 유형 수정

수정하기 전에 기존 유형을 백업하고, 기존 유형의 사본으로 작업을 수행합니다.

1. **TcFTSIndexer**가 설치되어 있고 독립 실행형 모드에서 작동 중인지 확인합니다.
2. *TC_ROOT***TcFTSIndexer\conf\TcFTSIndexer_***type-name1**.properties*를 *TC_ROOT***TcFTSIndexer\conf\TcFTSIndexer_***type-name2**.properties*로 복사합니다.
3. 파일의 새 이름에 대한 유형별 정의를 변경합니다.
4. 새 유형에 불필요한 플로 및 단계를 삭제합니다.
5. 플로와 연관된 모든 단계를 추가, 수정 또는 삭제합니다.
6. 속성 파일 정의를 기준으로 새 단계를 구현합니다.
7. 모든 구현 클래스가 포함된 사용자 정의 JAR 파일을 생성합니다.
8. 새 사용자 정의 JAR 파일을 *TC_ROOT***TcFTSIndexer\lib** 디렉터리에 복사합니다.
9. **runTcFTSIndexer -?** 명령을 실행하여 새 사용자 정의 유형 및 플로가 콘솔에 표시되는지 확인합니다.
10. 출력 콘솔 메시지를 바탕으로 모든 구성 및 구현 관련 문제를 해결합니다.
11. 다음 명령을 사용하여 플로를 실행합니다.

    ```
    runTcFTSIndexer -task=type:flow-action
    ```

검색 색인 플로 추가

플로에는 내부 이름 및 최종 사용자가 명령으로 실행한 관련 플로 작업이 포함됩니다. 플로 작업과 내부 플로 이름 간의 매핑은 다음과 같이 정의됩니다.

© 2024 Siemens

internal-flow-name.action=*flow-action*

예를 들면 다음과 같습니다.

```
reindexflow.action=reindex
```

플로의 경우 특정한 플로 동작을 지원해야 하는 경우를 제외하고는 기본 동작을 사용합니다. 다음과 같이 속성을 생성하여 사용자 정의 플로 클래스를 정의할 수 있습니다.

internal-flow-name=full-class-name

예를 들면 다음과 같습니다.

```
reindexflow=com.siemens.teamcenter.ftsi.objdata.TcFtsIndexeObjDataFlow
```

이 사용자 정의 구현 클래스는 **TcFtsIndexerFLow** 클래스를 확장해야 하며, 일반적으로 다음 메서드를 재정의해야 합니다.

```
public void initialize( Object zData )
```

이 메서드는 플로를 실행하는 도중 호출되며, 이 플로에 특화된 개체의 초기화 또는 입력의 유효성 검사를 지원합니다. 예를 들어, 개체 데이터 플로 구현 클래스는 성능상의 이유로 초기화 메서드를 재정의하여 캐시 파일의 Teamcenter 환경설정을 캐시합니다.

사용자는 **runTcFTSIndexer.bat/sh-status**명령을 입력해 지원되는 모든 플로에 대한 정보를 얻을 수 있습니다. 플로 설명이 다음과 같이 제공될 수 있습니다.

internal-flow-name.description=*description-text*

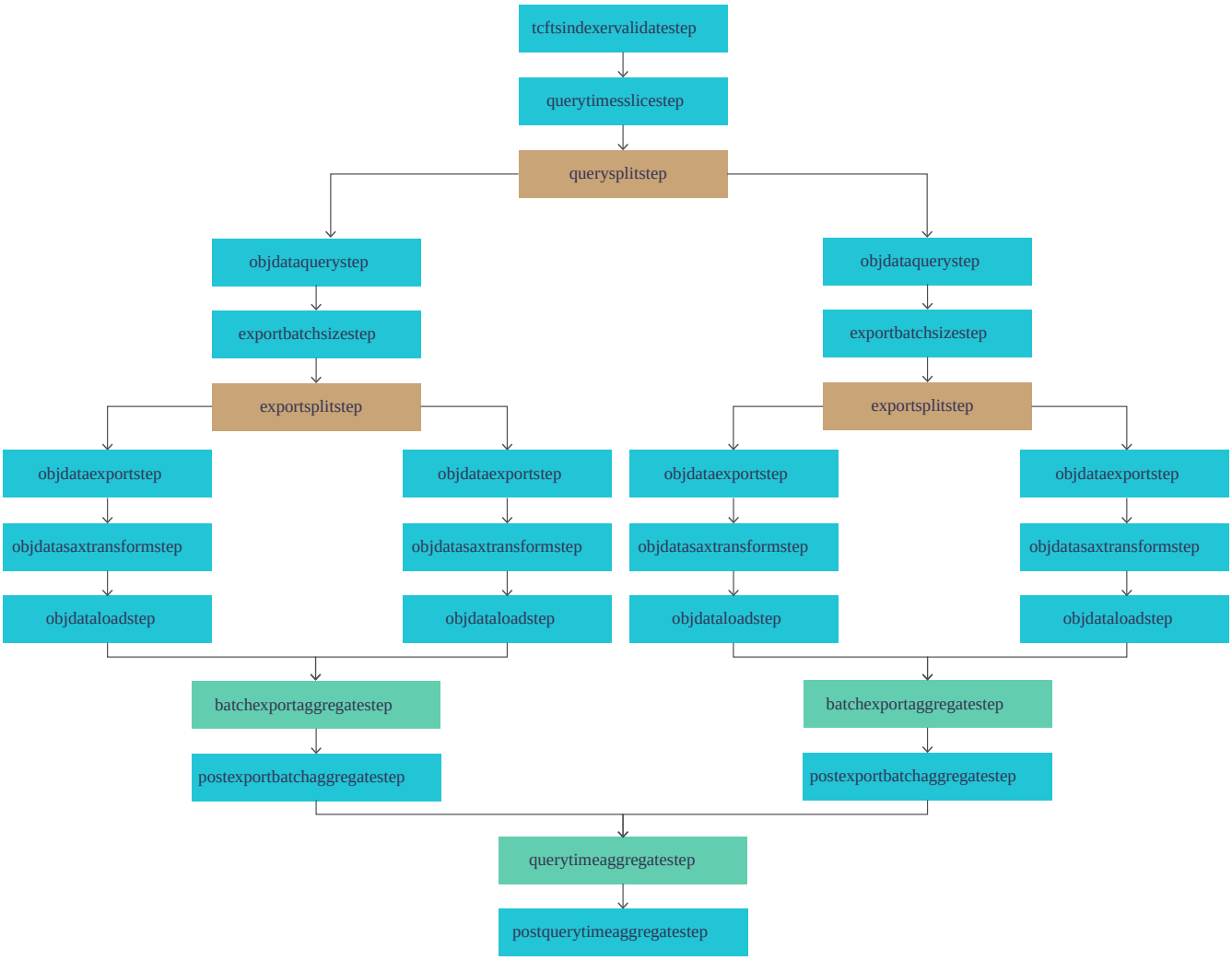
예를 들면 다음과 같습니다.

```
reindexflow.description=Clears the existing indexed data and performs index of data.
```

색인기 단계

다음은 **ObjData** 색인 플로의 단계입니다. 이러한 단계는 **ObjData** 복구 및 동기화 플로에서 재사용됩니다. 입력 및 출력 개체는 Javadocs에서 제공되는 각 단계와 연관됩니다.

© 2024 Siemens



© 2024 Siemens

검색 색인 단계 추가

- 구현 클래스

  각 단계는 구현 클래스와 연결해야 합니다. 이 클래스에는 명확하게 정의된 입력 및 출력 매개변수가 있어야 하며 이로 인해 필요한 경우 다른 플로에서 이 클래스를 재사용할 수 있어야 합니다. 단계는 속성 파일에서 다음과 같이 정의할 수 있습니다.

  step-name=full-class-name

  예를 들면 다음과 같습니다.

  ```
  querytimesslicestep=com.siemens.teamcenter.ftsi.objdata.steps.query.QueryTimeSliceStep
  ```

  단계의 유형을 기준으로 세 가지 옵션이 있습니다.

  - 단순 단계

    입력 개체를 가져와 이를 출력 개체로 변환하는 단순 단계의 경우, **TcFtsIndexerAbstractStep** 클래스를 확장하는 클래스를 구현합니다. (자세한 내용은 Javadocs를 참조하십시오.) 이를 위해서는 **process** 메서드를 구현해야 합니다.

    ```
    public ITcFtsIndexerStep.Status process( IStepInfo zStepInfo, IMessage zMessage )
       throws Exception
    ```

    각 단계의 **process** 메서드는 플로에 정의된 모든 단계에 대한 순서대로 호출됩니다. 프로세스의 출력은 다음 단계에 입력으로 전송됩니다. 프로세스에서는 **IStepInfo** 클래스를 인수로 취합니다. 이 클래스는 처리 중인 단계와 관련된 모든 상태를 추적합니다. 단계는 추적 정보를 설정할 수 있으며 도우미 메서드를 사용하여 **IStepInfo** 클래스를 사용하는 스테이징 디렉터리에 액세스할 수도 있습니다. 모든 **StepInfo** 개체는 색인화의 마지막에 또는 **-status** 명령 실행 도중 한 가지 상태로 인쇄됩니다.

    IMessage 개체는 입력/출력 개체를 유지하는 메시지 개체이며 플로의 전체 단계에서 액세스할 수 있습니다. 메시지 개체에는 Java 개체가 될 수 있는 데이터 개체가 포함됩니다. 단순 단계의 경우, 동일한 메시지 개체가 전달됩니다.

    TcFTSIndexer는 명령줄을 통해 전달되는 모든 인수를 구문 분석하며 프레임워크와 관련 없는 인수 리스트를 생성합니다. 이러한 인수는 유형별 인수로 간주됩니다. 이 리스트는 메시지 데이터로 처리하기 위해 플로의 첫 번째 단계에 전송됩니다.

    단계는 이 메시지 데이터 개체를 받아 업데이트할 수 있습니다. 또한 새 메시지 데이터 개체를 설정할 수도 있습니다. 예외가 발생하거나 **NoData** 상태가 반환될 경우 단계의 추가 처리가 중단될 수 있습니다.

    샘플 단계에 대한 자세한 내용은 *TC_ROOT***TcFTSIndexer\sample** 디렉터리를 참조하십시오.

  - 분할 단계

© 2024 Siemens

이 단계에서는 개별 요소에 대한 메시지 데이터 입력으로서 개체 리스트를 분할하며 리스트의 각 요소에 대해 새 메시지 개체를 생성합니다. **TcFTSIndexer**는 입력 메시지 데이터 리스트와 크기가 같은 병렬 스레드를 생성하고 새 메시지 개체로 다음 단계를 실행합니다. 분할 단계는 다음과 같이 정의할 수 있습니다.

step-name=com.siemens.teamcenter.ftsi.core.TcFTSIndexerSplitStep

예를 들어 개체 데이터 색인화를 수행하면 리스트의 리스트(**리스트<리스트<문자열>>**)가 분할됩니다. 이 리스트에는 특정한 배치 크기로 구성된 UID 리스트가 포함됩니다. 이 단계에서는 리스트의 리스트를 UID의 각 배치 크기 리스트를 처리하는 여러 개의 스레드로 나눕니다.

- 집계 단계

  이 단계에서는 모든 분할 스레드가 완료될 때까지 기다리며 성공적인 분할 단계 메시지 데이터를 새 메시지 개체에 통합하고 이를 다음 단계로 전달합니다. 집계 단계의 수는 분할 단계와 일치해야 합니다. 집계 단계는 다음과 같이 정의할 수 있습니다.

  step-name=com.siemens.teamcenter.ftsi.core.TcFTSIndexerAggregateStep

  예를 들어, 개체 데이터 집계 단계에서는 분할 단계에서 생성된 모든 스레드가 UID의 배치 크기를 처리할 때까지 기다립니다. 모든 분할 단계가 집계되면, 해당 데이터는 다음 단계에서 Solr에 커밋됩니다.

- 플로의 단계

  TcFTSIndexer는 지정된 순서대로 단계를 실행합니다. 플로에 정의된 단계의 순서는 다음과 같이 속성 파일에도 정의되어야 합니다.

  internal-flow-name.steps=*step-name-1*,*step-name-2*,*step-name-n*

  예를 들면 다음과 같습니다.

  ```
  reindexflow.steps=tcftsindexervalidatestep,querytimesslicestep,
     querysplitstep,objdataquerystep,exportbatchsizestep,exportsplitstep,
     objdataexportstep,objdatatransformstep,objdataloadstep,batchexportaggregatestep,
     postexportbatchaggregatestep,querytimeaggregatestep,postquerytimeaggregatestep
  ```

- 단계 상태

  기본적으로 **-status** 인수는 모든 단계의 상태를 출력합니다. 중요한 특정 단계에 대한 출력만 제어하는 경우, 다음 속성 형식을 사용할 수 있습니다.

  internal-flow-name.status=*step-name-1*,*step-name-4*,*step-name-7*

  예를 들면 다음과 같습니다.

  ```
  reindexflow.status=objdataquerystep,objdataexportstep,objdatatransformstep,
   objdataloadstep
  ```


© 2024 Siemens

새 검색 색인 유형 배치

전제 조건

- **TcFTSIndexer**가 설치되어 있고 독립 실행형 모드에서 작동 중인지 확인합니다.
- 새 검색 색인 유형 추가.
- 유형, 플로 및 단계에 대한 클래스를 속성 파일에 정의된 대로 구현합니다.

절차

1. 클래스 경로에 대한 *TC_ROOT***TcFTSIndexer\lib** 디렉터리에 JAR 파일을 추가하여 코드를 컴파일합니다.
2. 모든 사용자 정의 구현 클래스가 포함된 사용자 정의 JAR 파일을 생성합니다.
3. 새 사용자 정의 JAR 파일을 *TC_ROOT***TcFTSIndexer\lib** 디렉터리에 복사합니다.
4. runTcFTSIndexer **-status** 명령을 실행해 새 사용자 정의 유형 및 플로가 콘솔에 나타나는지 확인합니다.
5. 출력 콘솔 메시지를 바탕으로 모든 구성 및 구현 관련 문제를 해결합니다.
6. *TC_ROOT***TcFTSIndexer\conf\log4j.properties** 에서 **objdata**와 유사한 사용자 정의 유형에 대한 로깅을 설정합니다.
7. 다음 명령을 사용하여 플로를 실행합니다.

```
runTcFTSIndexer -task=type:flow-action
```

샘플 검색 색인화 플로 실행

TcFTSIndexer는 샘플 플로가 함께 제공됩니다. 다음 단계를 사용하여 이 샘플 플로를 실행할 수 있습니다.

1. *TC_ROOT***TcFTSIndexer\sample\TcFtsIndexer_sample1.properties** 파일을 *TC_ROOT***TcFTSIndexer\conf** 디렉터리에 복사합니다.
2. 명령을 실행합니다:

```
runTcFTSIndexer -task=sample1:sampleflow1
```

이 샘플 플로에는 다음이 포함됩니다.

© 2024 Siemens

- 4가지 단순 단계 구현 클래스(**SampleStep1.java**, **SampleStep2.java**, **SampleStep3.java**, **SampleStep4.java**). *TC_ROOT***TcFTSIndexer\sample** 디렉터리에 코드가 제공됩니다.
- 분할 단계 1개.
- 집계 단계 1개.
- 새 **samplestep5** 단계를 생성하는 데 재사용되는 **SampleStep4.java** 파일.

이 플로는 다음과 같이 연결됩니다.

1. **samplestep1**

   이 단순 단계에서는 리스트의 리스트(**List<List<String>>**)를 생성하고 이를 메시지 데이터에 추가합니다. 이 메시지 데이터 리스트에는 크기가 3인 샘플 문자열이 포함된 4개의 리스트가 있습니다. 이 메시지 데이터 리스트는 **StepInfo** 메시지의 추적 정보로 설정되어 상태를 표시합니다. 샘플 단계는 다음과 같습니다.

   ```
   [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3,
   Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3,
   Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3,
   Batch 4 MD 1, Batch 4 MD 2, Batch 4 MD 3]
   ```

2. **samplesplitstep**

   이는 네 개의 스레드를 생성하는 분할 단계이며, 이러한 스레드는 메시지 데이터 입력이 포함된 **samplestep2** 단계를 크기가 3인 문자열 내부 리스트로 호출합니다.

3. **samplestep2**

   이 단계는 크기가 3인 내부 리스트가 포함된 별도의 스레드를 사용하여 네 번 호출됩니다.

   ```
   Thread1 -> [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3]
   Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3]
   Thread3 -> [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3]
   Thread4 -> [Batch 4 MD 1, Batch 4 MD 2, Batch 4 MD 3]
   ```

4. **samplestep3**

   이 단계는 네 개의 각 스레드를 통해 호출되며 이전 단계에서 동일한 메시지 개체가 전송됩니다. 추가 문자열이 추가됩니다.

   ```
   Thread1 -> [Batch 1 MD 1, Batch 1 MD 2, Batch 1 MD 3, Message Data for Step3]
   Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
   Thread3 -> [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]
   Thread4 -> [Batch 4 MD 1, Batch 4 MD 2, Batch 4 MD 3, Message Data for Step3]
   ```


© 2024 Siemens

이 단계에서 한 번 실행하는 스레드는 **NoData** 상태로 반환되고 나머지에서는 예외가 발생합니다. 이러한 스레드에서는 더 이상 단계를 처리할 수 없으며 남은 스레드 두 개만 다음 단계로 이동됩니다.

5. **samplestep4**

   이 단계는 단 두 개의 스레드를 통해 호출되며 이 스레드와 연결된 메시지가 이전 단계에서 전달됩니다.

   ```
   Thread2 -> [Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
   Thread3 -> [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]
   ```

6. **sampleaggregatestep**

   이 단계에서는 네 가지 스레드가 모두 완료될 때까지 기다리며, 성공적인 스레드와 연결된 메시지 데이터를 집계합니다. 스레드 한 개에 오류가 발생했고 다른 스레드에는 데이터가 없으므로, 생성되는 통합 메시지 데이터는 두 가지 항목으로 구성된 리스트이며 이러한 두 가지 항목은 이전의 성공적인 메시지의 리스트입니다.

   ```
   [[Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
   [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]].
   ```

   이 단계에서는 **samplestep1** 분할 단계 메시지 개체 전의 메시지 개체를 가져오고 이를 다음 단계로 전달합니다.

7. **samplestep5**

   이 단계에서는 이전 집계 단계의 메시지 개체를 메시지 데이터와 함께 가져옵니다.

   ```
   Thread 5 -> [[Batch 2 MD 1, Batch 2 MD 2, Batch 2 MD 3, Message Data for Step3]
   [Batch 3 MD 1, Batch 3 MD 2, Batch 3 MD 3, Message Data for Step3]].
   ```

   메시지 데이터에 설정된 추적 정보는 **samplestep1** 단계에 설정된 것과 같습니다.

이 샘플 플로에서는 모든 단계마다 메시지 데이터를 출력하고 마지막 단계에서는 통합 상태를 제공합니다.

```
-------------------------------------------------------------------------------------------
-------
Status: Created: 0   Started: 0   Done: 1   Error: 0
Total Time    0.00   Total Count 12

Step Summary
  samplestep1
     Status: Created: 0   Started: 0   Done: 1   Error: 0

  samplestep2
     Status: Created: 0   Started: 0   Done: 4   Error: 0

  samplestep3
```


© 2024 Siemens

```
     Status: Created: 0   Started: 0   Done: 3   Error: 1

  samplestep4
     Status: Created: 0   Started: 0   Done: 2   Error: 0

  samplestep5
     Status: Created: 0   Started: 0   Done: 1   Error: 0

Total time for all Steps 0 sec
Overall Time 0.307 sec
Done processing Type: sample1 FlowAction: sampleflow1
```

저장된 쿼리에 ObjData 색인화 지원

저장된 조회의 UID에 대한 조회를 지원하기 위해 **ObjData** 유형에 샘플 **savedquery** 플로 작업이 추가됩니다. **savedquerystep** 단계에 대한 샘플 코드는 *TC_ROOT***TcFTSIndexer\sample** 디렉터리에서 제공됩니다. 이 샘플 단계는 저장된 쿼리 플로에 통합되어 있습니다. 플로 및 단계 구성은 *TC_ROOT***TcFTSIndexer\conf\TcFtsIndexer_objdata.properties** 파일에서 제공됩니다. 샘플 저장 쿼리와 관련된 속성의 이름은 **savedquerystep.***. 사용 및 상세정보에 대한 내용은 속성 설명을 참조하십시오.

이 저장된 조회를 실행합니다.

```
runTcFTSIndexer -task=objdata:savedquery
```

저장된 쿼리의 추가 요구사항은 샘플 코드를 변경하고 새 단계를 생성하여 처리할 수 있습니다. 이 새 단계는 기존 **savedquerystep** 단계를 대체합니다.

TcFTSIndexer 오류 해결

로그인 오류

runTcFTSIndexer 유틸리티를 실행할 때 SSO용으로 환경이 구성된 경우 오류가 발생할 수 있습니다. 잘못된 사용자 ID 또는 암호 오류가 발생한 경우 다음을 확인하십시오.

- Teamcenter client communication system SSO 앱 ID를 정확하게 구성하지 못할 수 있습니다. Teamcenter Environment Manager(TEM)의 **클라이언트 통신 시스템의 환경 설정** 패널을 사용하여 여러 응용 프로그램 ID를 구성할 수 있습니다.

- **runTcFTSIndexer** 유틸리티를 실행하는 사용자가 인증 시스템에서 인증되고 색인화할 데이터 집합 및 관련 파일에 대한 읽기 액세스 권한이 있는지 확인합니다.

- 사용자가 *TC_ROOT***TcFTSIndexer\conf***TcFTSIndexer.properties** 파일의 **Tc.user** 설정에 정의되었는지 확인합니다.

- 암호화된 암호를 사용하는 경우 올바르게 생성되었는지 확인해야 합니다. **TcFTSIndexer를 실행하여 사용자를 변경**할 수 있습니다.

© 2024 Siemens

색인기 액세스 오류

- 색인기에 액세스할 수 없거나 스키마가 올바르지 않습니다.

  Teamcenter 및 Solr 스키마를 병합합니다.

  SOLR_HOME\TcSchemaToSolrSchemaTransform.bat*TC_DATA*\ftsi\solr_schema_files

- Solr 게시물에 HTTP 응답 코드 **401**이 반환됩니다.

  Solr 자격 증명은 Teamcenter, 색인화 엔진(Solr) 및 **TcFTSIndexer**와 일치해야 합니다. 필요한 경우 Solr 자격 증명을 업데이트할 수 있습니다.

데이터베이스 테이블 오류

- 다음 예제 같은 오류를 참조하는 **syslog 파일에서 색인화 오류**를 발견한 경우, 이전 릴리스의 Teamcenter 데이터베이스에 **TIE_CLSF_DATA**라는 만료된 전역 임시 테이블이 있는 것입니다.

  - Syslog에 **ICM0UNITSYSTEM**이라는 데이터베이스 전역 임시 테이블에 추가 열이 있는 문제가 보고됩니다.

  - Syslog에 다음과 같은 오류가 보고됩니다.

    ```
    #######inside EIM_not_ok##########

    sqlca_error_code가 -904임
    EIM_db_error_code가 545001임
    롤백;
    ```

  데이터베이스에서 **TIE_CLSF_DATA** 테이블을 제거하여 문제를 해결할 수 있습니다. 색인화 또는 동기화 프로세스를 중지하고, 데이터베이스에서 테이블을 제거한 다음 색인화를 다시 시작합니다. 새 임시 테이블이 올바르게 생성됩니다.

- Syslog는 다음과 같은 전자 메일 알림에 대한 색인화 동기화 중 오류를 보고합니다.

  ```
  클린업 스크래치 테이블을 실행하는 중 오류 발생
  메일 서버에서 세션을 열 수 없음
  OS 메일 통보를 보내지 못했음
  ```

  오류는 동기화 중에 데이터베이스 구독 테이블에 대한 변경과 관련이 있습니다. 그러나 오류는 전자 메일 알림에만 적용됩니다. 다음 작업 중 하나를 수행할 수 있습니다.

  - 동기화 오퍼레이션을 방해하지 않으므로 오류를 안전하게 무시합니다.

  - **Subscription_Table_Notify_Level=2**(Teamcenter 도움말 항목 *사이트 통합을 통해 데이터 동기화 유지 관리 참조*)를 설정하여 전자 메일 알림을 억제하도록 선택합니다(예: 사이트에서 전자 메일 서버 또는 전자 메일 수신자가 구성되지 않을 경우).

© 2024 Siemens

데이터 집합 다운로드 오류

- 데이터 집합이 누락된 파일인 경우 **bmide_modeltool**을 실행하여 파일을 생성한 후 데이터 집합에 업로드합니다. 파일이 생성 또는 업로드되지 않는 경우 **bmide_modeltool** 로그를 참조하십시오. 자세한 내용은 Teamcenter 도움말의 *Business Modeler IDE* 을 참조하십시오.

- 파일 관리 시스템 문제로 인해 파일 다운로드 문제가 발생할 수 있습니다. FMS가 올바르게 구성되었는지 확인합니다. 자세한 내용은 Teamcenter 도움말의 *파일 관리 시스템*을(를) 참조하십시오.

데이터 집합 텍스트 파일 인코딩 오류

텍스트 파일 내용이 색인화되지 않은 경우 파일 인코딩을 확인합니다. UTF-F8 인코딩을 사용한 텍스트 파일이 올바르게 색인화되었습니다. 텍스트 파일이 다른 유형의 인코딩(예: `Shift-JIS`)을 사용하는 경우, 파일을 UTF-8로 변환합니다.

메모리 오류가 발생하는 데이터 집합 파일

비정상적으로 큰 파일을 색인화하는 경우 메모리 부족 오류가 발생하면 Tika 구문 분석기 관련 메시지를 **TcFTSIndexer 로그에서 확인**할 수 있습니다. 오류가 Tika 구문 분석기와 관련된 것으로 보이면 색인화 중에 실행하는 Tika 구문 분석기의 최대 수를 설정할 수 있습니다.

*TcFTSIndexer\conf\TcFTSIndexer_objdata.properties*에서 **objdataloadstep.maxTikaParserCount=***n*으로 설정합니다.

한 번에 구문 분석되는 데이터의 양이 줄어들도록 최대 수를 설정합니다. 메모리 부족 오류가 발생하지 않고 데이터가 구문 분석될 때까지 반복적으로 값을 줄일 수 있습니다. 기본값(값이 없음)은 무제한을 의미합니다.

사용자 정의 사용자 종료 방법과 개체 데이터를 필터링

사용자 끝내기 작업을 구현하여 특정 개체를 필터링하고 검색 색인에 추출되는 것을 방지할 수 있습니다. 사용자 끝내기 작업은 특정 자산 가치를 가진 ID 값이나 개체 등 포함할 기준을 지정할 방법을 제공합니다. 이 절차에서는 사용자 정의 방법을 구현하고 제공된 메시지에 대해 등록하고 구현에 사용자 정의 필터링을 추가하는 방법을 보여줍니다.

1. 새 라이브러리를 생성하거나 기존 라이브러리를 사용하여 사용자 정의 색인기 필터 논리를 구현합니다. 예를 들어 **libIndexerFilter**를 생성하고 이 라이브러리 이름을 **TC_customization_libraries** 환경설정에 추가합니다.

2. 라이브러리 내에서 *library-name*_**register_callbacks** 방법을 정의하고 내보내기합니다.

3. 이 방법에서는 **AWS2_filter_object_indexing_user_exit** 및 **AWS2_filter_dataset_object_indexing_user_exit** 사용자 끝내기 메시지에 대해 사용자 정의 구현을 등록합니다.

   예:

© 2024 Siemens

- **AWP0CUSTOM_skip_objects_to_index** 방법을 **AWS2_filter_object_indexing_user_exit** 메시지에 대해 등록합니다.

- **AWP0CUSTOM_skip_dataset_objects_to_index** 방법을 **AWS2_filter_dataset_object_indexing_user_exit** 메시지에 대해 등록합니다.

```
#define AWS2_filter_dataset_object_indexing_msg
   "AWS2_filter_dataset_object_indexing_user_exit"

#define AWS2_filter_object_indexing_msg
   "AWS2_filter_object_indexing_user_exit"

extern "C" DllExport int libawp0custom_register_callbacks()
{

CUSTOM_register_exit( "libawp0custom", AWS2_filter_object_indexing_msg,
   (CUSTOM_EXIT_ftn_t) AWP0CUSTOM_skip_objects_to_index );

CUSTOM_register_exit( "libawp0custom", AWS2_filter_dataset_object_indexing_msg,
   (CUSTOM_EXIT_ftn_t) AWP0CUSTOM_skip_dataset_objects_to_index );
    return 0;
}
```

4. **AWP0CUSTOM_skip_objects_to_index** 사용자 정의 방법을 구현합니다. 비 데이터세트 UID가 색인화되면 이 방법이 호출됩니다. 예:

5. **AWP0CUSTOM_skip_dataset_objects_to_index**에 대한 다른 사용자 정의 방법을 구현합니다. **AWP0CUSTOM_skip_objects_to_index**에 대해 이전 단계에 표시되는 동일한 패턴을 따릅니다. 데이터세트 UID가 색인화되면 이 방법이 호출됩니다.

6. 다음과 같은 환경설정에서 **AWP0CUSTOM_skip_dataset_objects_to_index** 방법에 대한 필터링을 활성화합니다.

   - **AWC_Search_Enable_UserExit_Datasets**에서 데이터세트 필터링을 활성화합니다.

   - **AWC_Search_Enable_UserExit_NonDatasets**에서 비 데이터세트 필터링을 활성화합니다.

   > 주:
   >
   > **AWC_Search_Enable_UserExit_NonDatasets**를 활성화하면 성능이 저하될 수 있습니다.

사용자 정의 사용자 엑시트 방법으로 외부 비즈니스 개체 반환

고급 검색은 ERP 시스템 등과 같은 대체 데이터 소스에서 개체를 반환할 수 있습니다. 외부 개체를 검색하려면 사용자 정의 사용자 엑시트 조회 방법을 구현해야 합니다. 외부 개체는 런타임 비즈니스 개체로 취급됩니다.

개체는 Active Workspace에서 고급 검색 조회를 통해 검색됩니다. 조회는 Rich Client Query Builder에서 생성됩니다.

© 2024 Siemens

사용자 정의 사용자 엑시트 조회 생성

이 절차에서는 사용자 정의 사용자 엑시트 조회를 생성하는 방법에 대해 설명합니다. 고급 검색에 대한 사용자 정의 조회가 이미 있는 경우 다음 절차로 건너뛸 수 있습니다.

1. 새 라이브러리를 생성하거나 기존 라이브러리를 사용하여 사용자 정의 외부 개체 논리를 구현합니다. 예를 들어, **libawp0custom.dll** 또는 샘플 소스 파일 *TC_ROOT\sample\examples\user_query.c*를 사용할 수 있습니다. 조회에 대한 사용자 정의 방법을 생성합니다. 그런 다음 라이브러리를 컴파일하여 *tc_bin*에 배치합니다.

2. 사용자 정의 라이브러리 방법과 일치하는 이름을 가진 조회를 생성합니다. 저장된 조회를 XML 파일로 내보낸 다음 조회 정의 XML 파일을 가져올 수 있습니다. 조회 정의 및 Query Builder에 대한 자세한 내용은 Teamcenter 도움말을 참조하십시오.

3. **tc_set_query_where_run** 명령을 실행합니다. XML 파일의 조회 이름(예: **-query=ExternalObject_exit_query**)을 지정하고 조회 플래그를 **-run=user**로 설정합니다.

4. 생성된 사용자 정의 라이브러리의 이름을 **TC_customization_libraries** 환경설정에 추가합니다.

   범위를 **사이트**로, 컨텍스트를 **Teamcenter**로, **값**을 라이브러리 이름(예: **libawp0custom**)으로 설정합니다.

개체에 대해 표시되는 내용 구성

조회를 생성하여 컴파일한 후 고급 검색 결과에서 반환된 개체에 대해 표시할 속성을 설정합니다.

1. 개체 속성을 검색 결과의 테이블 셀에 연결하는 환경설정을 생성합니다. *query-name*_**AW.CellProperties** 환경설정을 생성하고 외부 개체에 대해 표시할 속성을 지정합니다.

   사용자 엑시트 조회 이름과 동일한 조회 이름을 사용합니다. **EXTERNAL_exit_query**라는 이름의 조회인 경우 환경설정 이름은 **EXTERNAL_exit_query_AW.CellProperties**가 됩니다.

2. 서버 관리자를 다시 시작합니다.

사용자가 Active Workspace의 고급 검색 패널에서 *query-name*_**AW** 사용자 정의 조회 개체 유형을 선택합니다. 그런 다음 조회 조건을 입력하고 결과를 검색합니다.

색인기 설정 업데이트

Teamcenter Rich Client 및 Active Workspace를 패치한 후 색인기 설정을 업데이트해야 합니다. 유지보수 모드 또는 배포 센터에서 Teamcenter Environment Manager (TEM)을 사용하여 변경할 수 있습니다.

TEM에서 색인기 설정 업데이트

1. **유지보수** 패널에서 **구성 관리자**를 선택합니다.

© 2024 Siemens

2. **구성 유지보수** 패널에서 **기존 구성에서 유지보수 수행**을 선택합니다.
3. **이전 구성** 패널에서 색인기 구성을 선택합니다.
4. **구성요소 유지보수** 패널의 **색인기**에서 **색인기 설정 업데이트**를 선택합니다.
5. **Teamcenter 관리자** 패널에서 암호를 입력합니다.
6. 확인 후 **시작**을 클릭합니다.

배포 센터에서 색인기 설정 업데이트

1. **컴포넌트**에서 **색인기**를 선택합니다.
2. 모든 적용 가능한 색인기 설정을 구성합니다.
3. **컴포넌트 설정 저장**을 클릭합니다.
4. 배포 스크립트를 생성하고 영향 받는 기계에 소프트웨어를 배포합니다.

Teamcenter 이외의 데이터 색인화

외부 시스템 개체 생성

이 단계를 수행하기 전에 생성할 개체의 구조(즉, 속성 이름 및 유형)를 알고 있어야 합니다.

1. Business Modeler IDE에서 새 템플릿 프로젝트를 생성합니다. **기본** 및 **aw2**를 **종속 템플릿**으로 선택해야 합니다.

   자세한 내용은 Teamcenter 도움말의 Business Modeler IDE 템플릿 프로젝트 생성을 참조하십시오.

2. 새로 생성된 프로젝트에서 **Awp0AWCExternalSystemObject** 비즈니스 개체를 부모 개체로 사용하여 새 비즈니스 개체를 생성합니다.
3. **런타임** 속성으로 필요한 속성을 추가합니다.

   다음 속성이 지원됩니다. **Boolean(부울)**, **Date(날짜)**, **Double(이중 소수점)**, **Integer(정수)**, **String(문자열)**

4. Active Workspace에 표시할 개체 속성 내용에 대해 **Awp0SearchIsStored** 속성 상수를 **true**로 설정합니다.
5. **Awp0SearchIsIndexed** 속성 상수를 **true**로 설정하여 필드가 검색 가능한 상태임을 나타냅니다.

© 2024 Siemens

6. **Awp0AWCExternalSystemObject** 비즈니스 개체를 열고 **Awp0SearchIsIndexedExt** 비즈니스 개체 상수를 **true**로 설정하여 검색을 위해 외부 비즈니스 개체를 색인화했음을 나타냅니다.

7. 템플릿을 Teamcenter 서버에 배포합니다.

   템플릿에 대한 자세한 내용은 *Teamcenter Business Modeler IDE* 도움말에서 템플릿 배포 소개를 참조하십시오.

8. Active Workspace에서 비 Teamcenter 데이터의 형식을 지정하고 표시하는 데 사용되는 XRT 파일을 생성합니다.

   추가한 각 개체 유형에 대해서 Active Workspace에서 개체의 적절한 표시를 위해 XRT 파일을 포함한 2개의 데이터 집합을 생성해야 합니다. 하나는 **요약** 패널에 대한 것, 다른 하나는 **정보** 패널에 대한 것입니다(구문의 예시는 **Awp0ItemRevSummary** 및 **Awp0ItemRevInfoSummary**을 참조할 수 있음).

9. 다음 Teamcenter 환경설정을 추가하여 추가된 외부 시스템 개체에 이러한 XRT 파일을 링크합니다.

   - *external-system-object-name*.**CellProperties**
   - *external-system-object-name*.**INFORENDERING**
   - *external-system-object-name*.**SUMMARYRENDERING**
   - *info-XRT-dataset-name*.**INFO_REGISTEREDTO**
   - *info-XRT-dataset-name*.**SUMMARY_REGISTEREDTO**

   > 주:
   >
   > 새 환경설정에 **AWC_** 접두어를 추가하는 경우 Active Workspace에만 적용되며, 다른 Teamcenter 클라이언트에는 영향을 미치지 않습니다.

TcFtsIndexer에 데이터 색인화 추가

외부 개체 색인화를 시작하기 전에 **ObjData** 색인화가 어떻게 이루어지는지 숙지하십시오. 또한 **ObjData** 색인화를 위해 작동하는 **TcFTSIndexer** 설치가 있는지 확인하십시오. 외부 시스템에서 데이터를 가져오는 방법에 대한 자세한 내용을 확인하기 전에, **유형**, **플로** 및 **단계**를 생성하십시오. 이 플로에서는 **objdata:savedqueryFlow** 플로에 정의된 일부 단계를 재사용합니다.

1. 새 속성 파일을 *TC_ROOT***TcFTSIndexer\conf** 디렉터리에 추가하여 새 유형을 생성합니다. 예를 들어 **TcFtsIndexer_externaldata.properties** 파일을 생성합니다. (이렇게 하면 **externaldata** 유형이 생성됩니다.)

2. 플로 작업을 생성합니다(예: **extdataflow.action=extdata**).

3. *TC_ROOT***TcFTSIndexer\conf\TcFtsIndexer_objdata.properties** 파일에 있는 **objdata:savedquery** 플로에서 재사용할 단계를 복사합니다. 재사용 가능한 단계가 아래와 같이 정의됩니다. 이러한 단계에 대한 상세정보는 각 클래스와 연관된 Javadoc에 있습니다.

© 2024 Siemens

```
tcftsindexervalidatestep=com.siemens.teamcenter.ftsi.objdata.steps.TcFtsIndexerValidateStep
exportbatchsizestep=com.siemens.teamcenter.ftsi.objdata.steps.export.ExportBatchSizeStep
exportsplitstep=com.siemens.teamcenter.ftsi.core.TcFtsIndexerSplitStep
objdataloadstep=com.siemens.teamcenter.ftsi.objdata.steps.load.ObjDataLoadStep
batchexportaggregatestep=com.siemens.teamcenter.ftsi.core.TcFtsIndexerAggregateStep
postexportbatchaggregatestep=com.siemens.teamcenter.ftsi.objdata.steps.postprocess.PostBatchE
xportAggregateStep
postquerytimeaggregatestep=com.siemens.teamcenter.ftsi.objdata.steps.postprocess.PostQueryTim
eAggregateStep
```

생성할 플로의 정확한 단계 및 상세정보는 데이터 및 시스템에 따라 다르며, 그 예는 다음과 같습니다.

- 색인화할 개체의 수가 많은 시스템의 경우, **쿼리 호출 시 데이터를 ID로 반환**합니다. 이렇게 하면 시스템의 개체 수집 방식 및 처리할 데이터 청크의 크기를 더욱 폭넓게 제어할 수 있습니다.
- 기존 XML 내보내기가 이미 있는 환경의 경우, **내보내기 호출 시 데이터를 XML로 반환**합니다.
- 이와 다른 경우에는 **데이터를 Solr 입력 XML 형식으로 직접 내보내기**하는 편이 나을 수 있습니다.

조회 호출 시 외부 데이터를 외부 시스템에 ID로 반환

1. 단계를 통해 다음과 같이 플로를 정의합니다.

   ```
   extdataflow.steps=tcftsindexervalidatestep,extquerystep,
      exportbatchsizestep,exportsplitstep,extexportstep,exttransformstep,
      objdataloadstep,batchexportaggregatestep,postexportbatchaggregatestep,
      postquerytimeaggregatestep
   ```

   굵은 글꼴로 된 세 가지 단계(**extquerystep**, **extexportstep**, **exttransformstep**)는 속성 파일에서 생성하고 구현해야 합니다.

2. 외부 시스템에 연결하는 단계 클래스를 구현하고, 메시지 데이터에서 리스트로 색인화할 개체의 ID를 반환합니다. 저장된 쿼리를 실행하여 메시지 데이터에서 UID 리스트를 반환하는 샘플 코드를 보려면 *TC_ROOT***TcFTSIndexer\sample\TcFtsIndexerSavedQueryStep.java** 파일을 참조하십시오.

   출력은 ID 리스트이며, 다음 예를 참조하십시오.

   ```
   extquerystep=com.siemens.teamcenter.ftsi.externaldata.steps.query.QueryStep
   ```

3. 외부 시스템에 연결되는 내보내기 단계를 구현하고 입력 ID에 대한 XML 파일을 내보냅니다. 이 단계에서는 XML 파일에 대한 전체 경로를 메시지 데이터에 문자열로 반환해야 합니다.

   이 입력은 ID 리스트이며 출력은 외부 시스템 XML 파일에 대한 전체 경로입니다. 다음 예를 참조하십시오.

   ```
   extexportstep=com.siemens.teamcenter.ftsi.externaldata.steps.export.ExportStep
   ```


© 2024 Siemens

4. 이전 단계에서 XML 파일을 가져오는 단계를 구현하고 이를 Solr 입력 XML 파일로 변환합니다. 내보내기 단계의 출력이었던 XML 파일에 대한 전체 경로는 메시지 데이터로서 이 단계에 대한 입력으로 전송됩니다. 이 단계에서는 Solr XML 파일에 대한 전체 경로를 메시지 데이터 개체에 문자열로 반환해야 합니다. 입력은 외부 시스템 XML 파일에 대한 전체 경로를 나타내는 문자열입니다. 출력은 Solr XML 파일에 대한 전체 경로를 나타내는 문자열입니다. 다음 예를 참조하십시오.

```
exttransformstep=com.siemens.teamcenter.ftsi.externaldata.steps.transform.TransformStep
```

내보내기 호출 시 외부 데이터를 외부 시스템에 XML로 반환(조회 없음)

1. 단계를 통해 다음과 같이 플로를 정의합니다.

```
extdataflow.steps=tcftsindexervalidatestep,extexportstep,exttransformstep,
   objdataloadstep,postexportbatchaggregatestep,postquerytimeaggregatestep
```

굵은 글꼴로 된 두 가지 단계(**extquerystep**, **exttransformstep**)는 속성 파일에서 생성하고 구현해야 합니다.

2. 외부 시스템에 연결하는 내보내기 단계를 구현하고 색인화할 모든 개체에 대한 XML 파일을 내보냅니다. 이 단계에서는 XML 파일에 대한 전체 경로를 메시지 데이터 개체에 문자열로 반환해야 합니다.

출력은 외부 시스템 XML 파일에 대한 전체 경로를 나타내는 문자열입니다. 다음 예를 참조하십시오.

```
extexportstep=com.siemens.teamcenter.ftsi.externaldata.steps.export.ExportStep
```

3. 이전 단계에서 XML 파일을 가져오는 단계를 구현하고 이를 Solr 입력 XML 파일로 변환합니다. 내보내기 단계의 출력이었던 XML 파일에 대한 전체 경로는 메시지 데이터로서 이 단계에 대한 입력으로 전송됩니다. 이 단계에서는 Solr XML 파일에 대한 전체 경로를 메시지 데이터 개체에 문자열로 반환해야 합니다.

입력은 외부 시스템 XML 파일에 대한 전체 경로인 문자열이며, 출력은 Solr XML 파일에 대한 전체 경로인 문자열입니다. 다음 예를 참조하십시오.

```
exttransformstep=com.siemens.teamcenter.ftsi.externaldata.steps.transform.TransformStep
```

외부 데이터를 외부 시스템에서 Solr 입력 XML로 반환 (조회 및 내보내기 없음)

1. 단계를 통해 다음과 같이 플로를 정의합니다.

```
extdataflow.steps=tcftsindexervalidatestep,exttransformstep,
   objdataloadstep,postexportbatchaggregatestep,postquerytimeaggregatestep
```

exttransformstep를 속성 파일에 생성하고 구현해야 합니다.

© 2024 Siemens

2. 외부 시스템에 연결할 단계를 구현하고, 색인화할 모든 개체에 대한 Solr 입력 XML 파일을 생성합니다. 이 단계에서는 Solr XML 파일에 대한 전체 경로를 메시지 데이터 개체에 문자열로 반환해야 합니다.

   출력은 Solr XML 파일에 대한 전체 경로를 나타내는 문자열입니다. 생성할 Solr XML의 기본 형식은 다음과 같습니다.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<add>
     <doc>
          <field name=”property name”>Value of Prop</field>
          ...
          <field name=”property name”>Value of Prop</field>
     </doc>
     ...
     <doc>
          ...
     </doc>
</add>
```

   각 개체가 **<doc>** 요소로 표시됩니다. 개체의 속성은 **<field>** 요소로 표시됩니다. 이름 속성은 Business Modeler IDE에 정의된 속성을 나타내며, 형식은 다음과 같습니다:**TC_0Y0_***name-of-the-object***_0Y0_propertyname**.

   다음의 요소가 필요합니다:

   - **id**

     개체의 고유 ID를 지정합니다. 개체 ID가 충돌하지 않도록 하려면 고유한 ID로 접두어를 지정합니다.

   - **TC_GROUP_FIELD_ID**

     ID를 나타냅니다. 이는 내부적으로 사용되는 값이며, **id**와 동일해야 합니다.

   - **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSysObjClassName**

     Active Workspace에서 개체 유형을 올바르게 생성하는 데 사용되는 Business Modeler IDE에 정의된 개체의 이름을 식별합니다.

   - **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSystemName**

     개체를 가져온 외부 시스템의 이름을 식별합니다.

   - **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSystemURI**

     th개체를 표시할 URL을 정의합니다.

   - **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0PropertyNameList**

© 2024 Siemens

개체 속성의 각 Solr 이름에 대한 요소를 지정합니다.

값 형식은 다음과 같습니다. **TC_0Y0_***object-name***_0Y0_***property-name*.

- **TC_PRIV_am_rule_str**

  권한 읽기 수식을 정의합니다.

  각 개체와 함께 권한 정보가 색인화됩니다. 모든 사용자가 개체를 검색할 수 있도록 허용하는 기본 보안 읽기 식 문자열은 **EAKT(EACT+)EAYT+**입니다. 상세한 권한 정보를 얻으려면 Siemens Digital Industries Software의 지원이 필요할 수 있습니다.

- **TC_0Y0_WorkspaceObject_0Y0_object_type**

  Active Workspace의 **유형** 카테고리를 사용하여 이 클래스 이름의 개체를 필터링할 수 있습니다. 이 값은 **TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0ExternalSysObjClassName** 값과 같아야 합니다.

일반적인 XML 예는 다음과 같습니다.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<add>
   <doc>
        <field name="TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0External
        SysObjClassName">The object class name defined in the BMIDE</field>
        <field name=”TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0External
        SystemName”>The name of the external system</field>
        <field name=”TC_0Y0_Awp0AWCExternalSystemObject_0Y0_awp0External
        SystemURI”><field name=”TC_0Y0_Awp0AWCExternalSystemObject_0Y0_
        awp0PropertyNameList”>TC_0Y0_A4_address_0Y0_
        a4configuration</field>
        ...
        <field name=”TC_0Y0_Awp0AWCExternalSystemObject_0Y0_
        awp0PropertyNameList”>TC_0Y0_A4_address_0Y0_ a4type_
        displayName</field>
        <field name="id">A4_FD004</field>
        <field name="TC_Group_Field_Id">A4_FD004</field>
        <field name="TC_0Y0_A4_address_0Y0_a4configuration">Feature Dictionary
        Revision</field>
       ...
        <field name="TC_0Y0_A4_address_0Y0_a4type_displayName">Address</field>
        <field name="TC_PRIV_am_rule_str">EAKT(EACT+)EAYT+”</field>
   </doc>
```

Teamcenter에 대한 SOA 방법 호출은 이 방법에서 허용되지 않습니다.

외부 데이터용 TcFTSIndexer 구성

1. *TC_ROOT***TcFTSIndexer\lib** 디렉터리의 JAR 파일을 클래스 경로에 추가하고 구현된 코드를 컴파일한 후 이를 *.jar* 파일로 패키징합니다.

© 2024 Siemens

2. 생성된 *.jar* 파일을 *TC_ROOT***TcFTSIndexer\lib** 디렉터리에 복사합니다.

3. **TcFTSIndexer***TC_ROOT***TcFTSIndexer\conf** 디렉터리에 **데이터 색인화를 추가할 때 생성된 TcFTSIndexer_*type.properties* 파일을 복사합니다**.

4. 다음 예처럼 **runTcFTSIndexer -task=***type:flow-action* 명령을 실행해 플로를 실행합니다.

```
runTcFTSIndexer -task= externaldata:extdata
```

비동기화 파일 내용 색인화 사용자 정의

비동기화 파일 내용 색인화 사용자 정의 개요

비동기화 파일 내용 색인화 응용 프로그램을 설치하면 개체 메타데이터와 별도로 데이터 집합을 색인화할 수 있습니다. 이 응용 프로그램을 구성하면 NX 및 Solid Edge CAD 파일에 있는 정보를 색인화하고 검색할 수 있습니다. 이 응용 프로그램을 추가로 사용자 정의하면 사용자 정의 추출기를 사용하여 다른 독점 데이터 집합의 내용을 색인화할 수 있습니다.

다른 독점 파일의 파일 내용 색인화

기본적으로 비동기화 파일 내용 색인화 응용 프로그램은 NX 및 Solid Edge CAD 파일을 지원합니다. 이 응용 프로그램은 사용자 정의 추출기를 사용하여 독점 파일 형식을 색인화할 수 있습니다.

전제 조건

- **비동기화 파일 내용 색인기** 응용 프로그램을 설치합니다.
- **Awp0SearchIsIndexed** 상수를 활성화하여 데이터 집합 유형 **SampleDataset**을 색인화 가능으로 표시합니다.
- **AW_Indexable_File_Extensions** 환경설정 또는 **Awp0IndexableFileTypes** 전역 상수에 확장 *.sample*을 추가합니다.
- 독점 파일 형식에서 내용을 추출하는 사용자 정의 추출기를 생성하거나 획득합니다.

  비동기화 파일 내용 색인화 프레임워크는 추출된 내용이 XML 또는 JSON 형식인 경우 모든 추출기와 호환됩니다. 매핑 프레임워크가 구문 분석 및 색인화에 사용할 수 있는 JSON 및 XML 구조의 예는 *TC_ROOT*/**TcFTSIndexer/conf/specification/***Mapping_Specification.docx*를 참조하십시오.

  - 추출된 내용이 JSON 형식인 경우 키는 속성을 나타내야 합니다. 값은 색인화할 값을 나타내야 합니다.
  - 추출된 내용이 XML 형식인 경우 XML 요소는 속성을 나타내야 합니다. 요소 값은 색인화할 값을 나타내야 합니다.

  디스패처가 설치된 시스템이 추출기에 액세스할 수 있도록 합니다.

© 2024 Siemens

사용자 정의 추출기를 사용하여 독점 파일 유형 색인화

다음 예제에서는 Teamcenter 데이터 집합 **SampleDataset**에서 명명된 참조 **sampleFile**을 색인화하는 방법을 설명합니다. 파일의 확장은 *.sample*입니다. *.sample* 파일에서 색인화 가능한 내용을 추출할 수 있는 사용자 정의 추출기를 사용할 수 있는 것으로 가정합니다.

1. *Dispatcher_Root***Module\conf***translator.xml* 구성 파일의 제공된 **tcftsindexerfilecontentnxindex** 서비스를 템플릿으로 사용하여 새 디스패처 변환기 서비스를 생성합니다.

   기존 **tcftsindexerfilecontentnxindex** 서비스 정의를 복사하고 **mysampleservice**로 이름을 변경합니다.

   원본 서비스 정의에 표시된 표준 카멜 대/소문자 구문과 일치하기 위해 XML 요소 이름을 변경합니다. **MySampleService**는 표준 카멜 대/소문자 구문을 따릅니다.

   ```
   <MySampleService provider=“SIEMENS”
        service=“mysampleservice”
        maxlimit=“3” isactive=“true”>
        <TransExecutable dir="&MODULEBASE;/Translators/TcFTSIndexer/
   bin" name="runTcFTSIndexer.bat" />
        <Options>
            <Option name="flow" string="-task=filecontent:custom"
   description="TcFtsIndexer flow to extract and index file contents
   using a custom extractor."/>
            <Option name="mode" string="-standalone"
   description="Switch to indicate that TcFtsIndexer needs to run in
   standalone mode."/>
            <Option name="inputpath" string="-i=" description="Full
   path to the input file or directory."/>
       </Options>
       <TransErrorExclStrings>
            <TransInputStreamExcl string="Error: 0"/>
       </TransErrorExclStrings>
   </MySampleService>
   ```

   서비스 정의의 미리 알림은 기본 **tcftsindexerfilecontentnxindex** 서비스와 동일하게 유지됩니다. 이 서비스는 플로 작업 입력이 **-task=filecontent:custom**으로 설정된 상태에서 **runTcFTSIndexer** 유틸리티를 호출합니다.

   isactive가 **true**로 설정되어 있어야 합니다.

2. *TC_ROOT***TcFTSIndexer\conf***DatasetGroupingConfig.json*에서 데이터 집합 및 명명된 참조 조합에 대한 새 그룹화 항목을 추가합니다. 서비스 이름이 이전 단계에서 생성한 서비스와 일치해야 합니다.

   ```
   {
       "SampleDataset": {
           "sampleFile": [
   ```


© 2024 Siemens

```
                {
                    "name": "mysampleservice",
                    "limit": 10
                }
            ]
        },
        …
    }
```

이 항목은 데이터 집합에 대한 그룹화 규칙을 생성합니다. 색인기에서 명명된 참조 **sampleFile**이 있는 Teamcenter 데이터 집합 **SampleDataset**이 검색될 때마다 **mysampleservice**에 파일 내용 색인화 요청을 라우팅합니다.

3. 색인기에 대한 매핑 규칙이 포함된 매핑 구성 파일을 생성합니다. 색인기에서는 이러한 매핑 규칙을 사용하여 추출된 파일 내용을 색인화 가능한 형식으로 변환합니다.

   다음 예제에서는 **sampleMappingConfig** 이름이 매핑 규칙에 할당되어 JSON 파일로 저장됩니다.

   자체 매핑 파일을 생성하는 방법에 대한 자세한 지침은 *TC_ROOT*/**TcFTSIndexer/conf/specification/** *Mapping_Specification.docx*을 참조하십시오.

```
{
    "schemaVersion": "1.0.0",
    "configurations": {
        " sampleMappingConfig ": {
            "configVersion": "1.0.0",
            "description": "Mapping file for SampleDatasets.",
            "sourceFormat": "json",
            "rootPath": "$.",
            "dateTimeFormat": "yyyy-MM-dd",
            "mappings": [
                …
                ]
            }
        }
}
```

4. **aw_search_config_manager**를 사용하여 매핑 구성 파일을 매핑 구성 저장소에 업로드합니다. Teamcenter 명령 프롬프트에서 다음 명령을 실행합니다.

```
aw_search_config_manager -import -dir=mapping-file-directory
```

5. *Dispatcher_Root***Module\Translators\TcFTSIndexer\conf***ExtractorConfig.json* 파일에서 데이터 집합 및 참조 이름 조합에 대한 새 항목을 추가합니다.

   실행 가능한 추출기에 대한 **extractor** 섹션을 구성합니다. 다음 예제에서 이 추출기는 *c:\extractors\custom_extractor\sample_extractor.exe*입니다.

© 2024 Siemens

configID가 이전에 생성된 매핑 구성의 식별자와 일치하도록 **transformer** 섹션을 구성합니다.

```
"SampleDataset/File": {
    "extractor": {
        "type": "system-executable",
        "path": "
c:\extractors\custom_extractor\sample_extractor.exe ",
        "inputFileArg": "",
        "outputFileArg": "",
        "optionalArg": "",
        "argValueSeparator": " "
    },
    "transformer": {
        "configID": "sampleMappingConfig"
    }
  }
```

6. **runTcFTSIndexer** 표준 **objdata** 플로(예: **색인** 또는 **동기화**)를 사용하여 **SampleDataset**를 색인화합니다.

   기본 색인기에서 데이터 집합에 대한 파일 내용 색인화 요청을 타겟 서비스에 라우팅합니다. 그런 다음 대상 서비스에서 추출기를 호출합니다. 생성한 매핑 규칙을 사용하면 내용이 Solr에 색인화됩니다.

Solr 사용자 정의

장애 조치에 대한 Solr URL 지원 구성

Solr URL에 연결 문제가 발생한 경우 **TcFTSIndexer** 및 **TcServer**가 교체 Solr URL에 액세스할 수 있습니다. 하나의 Solr URL에 장애가 발생해도 **TcFtsIndexer**가 계속 색인화할 수 있고 클라이언트가 검색하여 모든 개체를 찾을 수 있습니다.

다음과 같은 전제 조건이 있습니다.

- SolrCloud가 필수이며 장애 조치를 지원하도록 구성됩니다.
- 리더 및 Zookeeper가 여러 시스템에 구성되어 있습니다.

  SolrCloud 리더 및 Zookeeper에 대한 자세한 내용은 다음을 참조하십시오.

  https://cwiki.apache.org/confluence/display/solr/SolrCloud

장애 복구의 경우 한 시스템이 작동 중지되어도 다음이 가능해야 합니다.

- **TcFtsIndexer**는 데이터를 색인화할 수 있는 대체 Solr URL에 연결할 수 없습니다. 실패한 URL에서 색인화된 개체를 교체 URL에서 계속 액세스할 수 있어야 합니다.

© 2024 Siemens

- **tcserver**는 색인화된 모든 개체에 액세스할 수 있는 대체 URL에 연결할 수 없습니다.

TEM의 설치 패널을 통해 여러 Solr URL을 구성할 수 있습니다. 여러 Solr URL을 지원하기 위해 **AWS_FullTextSearch_Solr_URL**이 다중 값 환경설정입니다. 프로세스는 다음과 같습니다.

1. **TcFtsIndexer** 및 **TcServer**가 이 환경설정을 사용하여 Solr URL에 연결하려고 합니다.

2. URL에 대한 액세스를 세 번 시도하고, 실패한 경우 환경설정 리스트에 있는 사용 가능한 다음 Solr URL이 사용됩니다. 액세스가 성공하거나 전체 리스트가 시도될 때까지 이를 계속합니다.

3. 이 문제가 해결될 때까지 **TcFTSIndexer** 및 **tcserver**에서 실패한 URL에 대한 경고 메시지를 출력합니다.

배포 센터를 사용할 때 HTTPS에 대한 Solr 구성

기본적으로 Solr은 HTTP를 사용하지만 배포 센터에서 설치 중 또는 설치 후에 HTTPS를 사용하도록 Solr을 구성할 수 있습니다. **TcServer** 및 **TcFtsIndexer**는 자체 서명된 인증서와 CA(인증 기관) 서명된 인증서를 승인하도록 프로그래밍되어 있습니다.

> 주의:
>
> HTTPS에 대한 Solr을 구성한 경우 HTTP로 다시 변경할 수 없습니다.

절차

1. Solr에 대해 신뢰할 수 있는 CA가 서명한 인증서를 선택하거나 Java **keytool** 또는 **openssl**을 사용하여 자체 서명된 인증서를 생성합니다.

   계속하기 전에 보안 인증서를 사용할 수 있는지 확인하십시오. 키스토어에 생성된 인증서를 저장합니다.

2. Solr을 중지합니다.

3. 인증서, 예를 들어 *IdentityKeystore.jks*, *TrustKeystore.jks* 또는 *private.cer*을 *SOLR_HOME\server\etc* 디렉터리에 복사합니다.

4. 배포 센터의 색인화 엔진 컴포넌트 섹션에서 **모든 매개변수 표시** ◎를 클릭합니다.

5. SSL 구성 아래에서 **https**를 서버 프로토콜로 선택합니다.

6. 사용 가능한 필드에 인증서에 대한 **키 저장소** 및 **신뢰 저장소** 정보를 입력합니다.

7. **컴포넌트 설정 저장**을 클릭합니다.

8. 배포 스크립트를 생성하고 영향 받는 기계에 소프트웨어를 배포합니다.

© 2024 Siemens

9. **Solr를 다시 시작합니다.**

10. HTTPS를 사용하여 Solr 관리자 페이지에 로그온하여 구현을 테스트합니다.

    HTTP가 더 이상 연결되지 않도록 합니다.

11. **AWS_FullTextSearch_Solr_URL** 환경설정을 HTTPS URL, 예를 들어 **https://mysolrserver:8984/solr**를 사용하도록 업데이트합니다.

12. *TC_ROOT***TcFTSIndexer\cache\TcFTSIndexerSession.cache**를 지웁니다.

Teamcenter Environment Manager을 사용할 때 HTTPS에 대한 Solr 구성

기본적으로, Teamcenter Environment Manager(TEM)으로 설치할 때 Solr은 HTTP를 사용하지만 HTTPS를 사용하도록 Solr을 구성할 수 있습니다. **TcServer** 및 **TcFtsIndexer**는 자체 서명된 인증서와 CA(인증 기관) 서명된 인증서를 승인하도록 프로그래밍되어 있습니다.

1. Solr에 대해 신뢰할 수 있는 CA가 서명한 인증서를 선택하거나 Java **keytool** 또는 **openssl**을 사용하여 자체 서명된 인증서를 생성합니다.

   계속하기 전에 보안 인증서를 사용할 수 있는지 확인하십시오. 키스토어에 생성된 인증서를 저장합니다.

2. Solr을 중지합니다.

3. 인증서, 예를 들어 *IdentityKeystore.jks*, *TrustKeystore.jks* 또는 *private.cer*을 *SOLR_HOME\server\etc* 디렉터리에 복사합니다.

4. **SOLR_HOME\server\etc***jetty-ssl.xml* 파일을 인증서 이름으로 업데이트합니다.

   ```
   <Set name="keyStorePath"><Property name="solr.jetty.keystore"
        default="./etc/IdentityKeystore.jks"/></Set>
   <Set name="TrustStorePath"><Property name="solr.jetty.truststore"
        default="./etc/TrustKeystore.jks"/></Set>
   ```

5. **SOLR_HOME\bin\solr.in.cmd**에서 다음 문장의 주석을 제거하고 해당 인증서의 전체 경로로 업데이트합니다. 예를 들면 다음과 같습니다.

   ```
   set SOLR_SSL_KEY_STORE=
      D:\SolrVersion\Leader1\solr-version\server\etc\IdentityKeystore.jks
   set SOLR_SSL_KEY_STORE_PASSWORD=secret
   set SOLR_SSL_TRUST_STORE=
      D:\SolrVersion\Leader1\solr-version\server\etc\TrustKeystore.jks
   set SOLR_SSL_TRUST_STORE_PASSWORD=secret
   set SOLR_SSL_NEED_CLIENT_AUTH=false
   ```

6. Solr을 Windows 서비스로 설치하는 경우 HTTPS에 대한 서비스를 설정하십시오.

© 2024 Siemens

a. **SOLR_HOME\server\etc***jetty-ssl.xml* 파일에서 다음 행을 찾아 주석을 추가합니다.

```
<!--
   <Call class="org.apache.solr.util.configuration.SSLConfigurationsFactory"
     name="current">
      <Get name="keyStorePassword" id="keyStorePassword"/>
      <Get name="trustStorePassword" id="trustStorePassword"/>
  </Call>
-->
```

b. KeyStorePassword 행을 찾아서 다음으로 교체합니다.

```
<Set name="KeyStorePassword">
<Env name="SOLR_SSL_KEY_STORE_PASSWORD" default="secret"/>
</Set>
```

c. TrustStorePassword 행을 찾아서 다음으로 교체합니다.

```
<Set name="TrustStorePassword">
<Env name="SOLR_SSL_TRUST_STORE_PASSWORD" default="secret"/>
</Set>
```

d. **SOLR_HOME\server\etc***jetty-https.xml*에서 Eclipse Jetty Server 연결의 배열을 다음으로 바꿉니다.

```
<Array type="org.eclipse.jetty.server.ConnectionFactory">
  <Item>
    <New class="org.eclipse.jetty.server.SslConnectionFactory">
      <Arg name="next">http/1.1</Arg>
      <Arg name="sslContextFactory"><Ref refid="sslContextFactory"/></Arg>
    </New>
  </Item>
  <Item>
    <New class="org.eclipse.jetty.server.HttpConnectionFactory">
      <Arg name="config"><Ref refid="sslHttpConfig"/></Arg>
    </New>
  </Item>
</Array>
```

e. Windows 서비스를 설치합니다.

A. 텍스트 편집기에서 **SOLR_HOME***solrWinService.bat*를 엽니다. **--module=http**를 찾아서 **--module=https**로 변경합니다. 파일을 저장합니다.

B. 관리자로 명령 윈도우를 엽니다. **solrWinService.bat -i**를 실행하여 서비스를 설치합니다.

C. 서비스를 설치한 후 Windows **서비스**를 열고 **Active Workspace 색인화 서비스**를 찾습니다. **다음 사용자로 로그온** 아래에서 시작하는 데 필요한 사용자 계정을 선택합니다.

다음 사용자로 로그온이 **로컬 시스템**으로 설정되어 있으면 로그온하는 데 사용하는 계정으로 변경한 후 서비스를 시작합니다. Solr 사용자 이름 및 암호를 사용자 계정으로 사용하지 마십시오.

© 2024 Siemens

서비스 **속성**을 열고 **로그온** 탭을 클릭하고 **계정 지정**을 선택한 후 이름과 암호를 지정합니다.

7. Solr를 다시 시작합니다.

8. 테스트를 구현합니다. HTTPS를 사용하여 Solr 관리자 페이지에 로그온합니다.

   HTTP가 더 이상 연결되지 않았는지 테스트합니다.

9. **AWS_FullTextSearch_Solr_URL** 환경설정을 HTTPS URL, 예를 들어 **https://mysolrserver:8984/solr**를 사용하도록 업데이트합니다.

10. *TC_ROOT***TcFTSIndexer\cache\TcFTSIndexerSession.cache**를 지웁니다.

2계층 환경에 대한 Solr 구성

Active Workspace를 2계층 Rich Client와 함께 사용하는 경우(예: NX와 함께 실행), 사용자가 이러한 환경에서 검색할 수 있도록 Solr을 설정해야 할 수 있습니다.

1. 2계층 환경을 위해 *tc_profilevars.bat* 파일을 엽니다.

2. **TC_INDEXING_ENGINE_PASSWORD_FILE** 환경 변수에 대해 암호 파일의 위치를 변경합니다. 예를 들면 다음과 같습니다.

   TC_INDEXING_ENGINE_PASSWORD_FILE=*path_to*_TEAMCENTER_solr_admin.pwf.

3. **TC_INDEXING_ENGINE_USER=solr_admin**을 설정합니다.

4. 2계층 **tcserver**를 다시 시작합니다.

Solr 자격 증명 업데이트

Solr 색인화 엔진에 대한 사용자 이름 및 암호를 업데이트해야 하는 경우 유지보수 모드 또는 배포 센터에서 Teamcenter Environment Manager(TEM)를 사용하여 변경할 수 있습니다. Solr 사용자 이름 및 암호는 여러 위치에서 업데이트해야 합니다. Solr 색인화 엔진, **TcFTSIndexer** 및 서버 확장에 대한 자격 증명이 일치해야 합니다.

TEM에서 Solr 자격 증명 업데이트

1. **유지보수** 패널에서 **구성 관리자**를 선택합니다.

2. **구성 유지보수** 패널에서 **기존 구성에서 유지보수 수행**을 선택합니다.

3. **이전 구성** 패널에서, Active Workspace 컴포넌트가 설치된 구성을 선택합니다.

© 2024 Siemens

4. **구성요소 유지보수** 패널의 옵션은 선택한 구성으로 설치된 항목을 기반으로 합니다.

   색인화 엔진→색인화 엔진 사용자 자격 증명 업데이트를 선택하여 Solr에 대한 자격 증명을 업데이트합니다.

   색인기→색인화 엔진 사용자 자격 증명 업데이트를 선택하여 **TcFTSIndexer**에 대한 자격 증명을 업데이트합니다.

   데이터 모델→색인화 엔진 사용자 자격 증명 업데이트를 선택하여 서버 확장 기능이 설치된 모든 Teamcenter 서버에 대해 Solr 자격 증명을 업데이트합니다.

배포 센터에서 Solr 자격 증명 업데이트

1. **컴포넌트**에서 **색인화 엔진**을 선택합니다.
2. **사용자** 및 **암호**를 포함하여 모든 적용 가능한 색인화 엔진 설정을 구성합니다.
3. **컴포넌트 설정 저장**을 클릭합니다.
4. 배포 스크립트를 생성하고 영향 받는 기계에 소프트웨어를 배포합니다.

Solr 오류 해결

Solr 로그는 TC_ROOT**\solr**-version**\server\logs***solr.log*에 있습니다.

Solr 인증

JSON 구문 분석 중에 오류가 발생하면 알 수 없는 값 유형 오류가 표시될 수 있습니다. Solr 암호가 **tcservers**에 올바르게 설정되었는지 확인하십시오.

Solr 자격 증명은 Teamcenter, Solr 색인화 엔진 및 **TcFTSIndexer**와 일치해야 합니다. **Solr 자격 증명 업데이트**.

Solr 스키마 문제

Solr 스키마가 오래된 경우 *solr.log* 파일에는 문제가 있는 필드에 대한 정보가 포함됩니다. **Teamcenter 및 Solr 스키마를 병합**했는지 확인하십시오.

```
SOLR_HOME\TcSchemaToSolrSchemaTransform.bat TC_DATA\ftsi\solr_schema_files
```

Solr 스키마를 확인할 수 있습니다.

1. Solr 관리 패널에서 Solr 스키마를 검사하고 **TC_0Y0_***으로 시작하는 Teamcenter 스키마를 찾습니다.
2. **bmide_extractor** 유틸리티를 실행하여 데이터 모델을 추출하고 데이터 모델 변경을 확인합니다.

© 2024 Siemens

3. **preferences_manager** 유틸리티를 실행하여 환경설정을 추출하고 Active Workspace 환경설정을 확인합니다.

4. Solr를 다시 시작해야 합니다.

누락된 필터 또는 기타 스키마 문제

다음을 확인하십시오.

- 데이터가 색인화되었는지 확인합니다.
- Teamcenter 스키마를 Solr 스키마와 병합한 후 Solr이 시작되었는지 확인합니다.
- 필요한 모든 데이터 모델 변경이 Business Modeler IDE에서 수행되었는지 확인합니다.
- Hot Deploy 또는 라이브 업데이트를 사용하여 데이터 모델 변경을 Teamcenter에 배포한 경우 **TC_ROOT** 및 **TC_DATA**가 정의된 Teamcenter 명령 윈도우에서 **bmide_modeltool** 유틸리티가 실행되었는지 확인합니다. 예를 들면 다음과 같습니다.

```
bmide_modeltool.bat -u=username -p=password -g=dba -tool=all
   -mode=upgrade -target_dir="TC_DATA"
```

Solr 메모리 문제

색인화 프로세스를 이해하고 색인화할 하드웨어를 고려해야 합니다.

Solr 메모리 문제가 발생한 경우 Java 힙 크기를 늘릴 수 있습니다. 메모리 사용량 해석에 대한 자세한 내용은 Solr 문서의 **JVM 설정** 섹션을 참조하십시오.

https://solr.apache.org

초기 전략으로, 문서가 1천만 개 미만인 색인의 경우 힙 크기를 10-16GB로 설정할 수 있습니다. 1천만 개 이상의 문서에 대한 색인의 경우 힙 크기를 컴퓨터 메모리의 50%로 늘일 수 있습니다. 예를 들어, 80GB의 메모리가 있는 경우 힙 크기를 40GB로 설정해야 합니다.

메모리 사용량을 정기적으로 모니터링하고 필요한 경우 힙 크기를 조정합니다. 예를 들어, 힙 크기를 40GB로 설정했지만 메모리 사용량이 20GB인 경우 힙 크기를 25GB로 설정하여 실제 사용량에 25%를 더한 값으로 고려합니다.

스크립트를 엽니다.

SOLR_HOME\bin\bin\solr.in.sh 또는 cmd

다음과 같이 Java 메모리 값을 설정합니다.

SOLR_JAVA_MEM="-Xms*size* -Xmx*size*"

© 2024 Siemens

-Xms는 초기 Java 힙 크기를 지정합니다. **-Xmx**는 최대 힙 크기를 지정합니다.

size 값을 **m** 또는 **M**을 사용하여 메가바이트로 지정하거나 **g** 또는 **G**를 사용하여 기가바이트로 지정합니다.

> 예제:
>
> SOLR_JAVA_MEM="-Xms8g -Xmx10g"

AdoptOpenJDK가 포함된 Windows 서비스로 Solr을 실행하는 경우 메모리 문제가 발생할 수 있습니다. Solr Windows 서비스의 경우, 힙 크기가 설정되어 있지 않으며 기본값은 256MB입니다. **-Xmx**를 사용하여 최대 힙 크기를 늘릴 수 있습니다.

스크립트를 엽니다.

SOLR_HOME\bin**solrWinService.bat**

다음으로 시작되는 행을 찾습니다.

windowsService.exe %PARAM1% -service=%SERVICENAME%

-Xmx*n***g** 사양을 행 끝에 추가합니다(예: 8GB의 경우 **-Xmx8g**). 파일을 저장하고 서비스를 다시 시작합니다.

> 주:
>
> Oracle OpenJDK는 이 문제의 영향을 받지 않습니다.

Solr Java 버전 문제

Solaris에서 시작하는 동안 Solr에 대한 Java 버전 문제가 발생할 수 있습니다. **runSolr.sh**가 다음과 같은 오류를 반환하는 경우 시작 스크립트에서 참조된 버전을 업데이트할 수 있습니다.

awk: /version/ {print $2}을(를) 열 수 없음
현재 Java 버전이 너무 오래되어 이 Solr 버전을 실행할 수 없습니다.

SOLR_HOME\bin\solr.sh을 열고 **JAVA_VER_NUM** 행을 찾습니다. 다음과 같이 나타납니다.

```
JAVA_VER_NUM=$(echo $JAVA_VER | head -1 |
  awk -F "" '/version/ {print $2}' | sed -e's/^1\.//' | sed -e's/[._-].*$//')
```

값을 Java 버전 번호로 교체합니다.

JAVA_VER_NUM=*Java 버전*

주요 Java 버전 번호(예: **8**)를 입력해야 합니다. 버전 문자열(예: **1.8.0**)을 입력하지 마십시오.

© 2024 Siemens

SolrCloud

schema.xml 파일을 외부 Zookeeper로 업로드했을 때 SolrCloud가 시작되지 않는 경우 다음 오류가 나타날 수 있습니다.

```
org.apache.solr.cloud.SolrZkServer  ZooKeeper Server ERROR java.io.IOException:
    Unreasonable length = 1070263 message
```

부적절한 길이로 인해 *schema.xml* 파일의 Zookeeper에서 허용되는 최대 크기까지 참조됩니다. 기본적으로 최대 크기는 1MB입니다.

*SOLR_HOME***server\solr\collection1\conf***schema.xml**이 **jute.maxbuffer** 시스템 속성에서 허용하는 제한보다 큰 경우 오류가 발생합니다.

문제를 해결하려면 Zookeeper 데이터 한도를 늘립니다. 다음 예제에서는 2MB로 설정합니다.

1. Zookeeper를 중단하고 *zoo_data* 캐시를 삭제합니다.

2. 한도를 변경합니다. 이 예제에서 **-Djute.maxbuffer=2097152**는 한도를 2MB로 설정합니다.

```
call %JAVA% -Djute.maxbuffer=2097152 "-Dzookeeper.log.dir=%ZOO_LOG_DIR%"
   "-Dzookeeper.root.logger=%ZOO_LOG4J_PROP%" -cp "%CLASSPATH%" %ZOOMAIN% "%ZOOCFG%"
%*
```

3. 스크립트에서 **jute.maxbuffer***ZOOKEEPER_HOME**bin**zkServer.cmd* 값을 변경합니다.

SolrCloud 구성

SolrCloud 구성 개요

SolrCloud 구성은 샤딩을 사용하는 복제본 및 분산 색인을 사용하여 데이터의 고가용성을 지원합니다. 이러한 구성을 독립적으로 또는 조합하여 사용할 수 있습니다. 이러한 용어는 Solr 클라우드 구성 옵션을 설명하는 데 도움이 됩니다.

노드	Solr의 단일 인스턴스입니다.
클러스터 (Cluster)	노드 그룹입니다.
컬렉션	단일 검색 색인입니다.
샤드	단일 컬렉션의 논리 섹션입니다.
복제본 (Replica)	샤드의 물리적 인스턴스입니다.
리더	좌표 요청으로 지정된 복제본입니다.
ZooKeeper	Solr 구성, 로드 밸런싱 및 장애 복구 동작을 처리합니다.

© 2024 Siemens

독립형 Solr 구성

기본적으로 독립형 Solr은 단일 복제본이 있는 단일 샤드로 단일 노드에서 구성됩니다. 이는 SolrCloud 구성이 아니므로 ZooKeeper는 이 구성의 일부가 아닙니다.

클라우드 구성 옵션 이해

SolrCloud는 서버 간에 색인화된 내용을 관리하기 위한 분포 환경을 제공합니다. 모든 SolrCloud 구성은 ZooKeeper를 사용합니다. 색인화 전략과 검색 목표에 가장 적합한 방법을 선택할 수 있습니다.

> 주:
>
> Smart Discovery는 단일 샤드와 하나 이상의 복제본이 있는 SolrCloud 구성만 지원합니다.

SolrCloud 구성	기반 기술	설명	혜택
고가용성	하나 이상의 복제본이 있는 단일 샤드.	조회는 리드 복제에 의해 관리됩니다.	사용 가능성을 유지하고 장애 복구 중복을 제공합니다.
분포 색인	여러 샤드(각 샤드는 자체 복제본이 있음).	조회는 모든 복제본에 동시에 전송됩니다.	특히 큰 색인의 경우 성능을 향상시킵니다.
조합	여러 샤드(각 샤드는 자체 복제본 집합이 있음).	각 샤드의 복제본은 다른 샤드의 복제본과 그룹화되어 그룹을 형성합니다. 조회는 복제본의 리드 그룹에 전송됩니다.	사용 가능성, 성능 및 장애 복구 중복을 제공합니다.

고가용성 구성

이 구성은 물리적 서버 또는 네트워크의 문제로 인해 노드의 연결이 끊긴 경우 장애 조치를 관리하도록 설계되었습니다. 샤드당 복제 수를 연장할 수 있습니다. 최선의 방법으로 별도의 시스템에 복제본을 설정합니다. 리더 위치는 처음 시작된 복제본에 할당됩니다.

> 예제:
>
> **replica1**(리더)은 **server1**에, **replica2**는 **server2**에 있습니다. 색인은 **replica1**과 **replica2**에서 복제됩니다.
>
> **server1**이 오작동하고 연결이 끊긴 경우 검색 요청이 ZooKeeper에 의해 자동으로 **replica2**로 라우팅됩니다. **server2**의 **replica2**는 리더로 사용됩니다. 새 복제본 리더는 해당 연결이 중지되거나 손실되지 않는 한 리더를 유지합니다.

분포 색인 구성

이 구성은 샤드 간에 색인을 배포하여 성능을 향상시키기 위해 설계되었습니다. 고가용성 구성과 비슷하지만 각 샤드에는 자체 복제본이 있습니다. 검색 요청은 각 복제본에 동시에 전송되어 색인의 모든 섹션을 조회합니다. 결과가 단일 응답으로 조합됩니다.

© 2024 Siemens

예제:

shard1에 대한 replica1은 **server1**에 있으며, **shard2**에 대한 **replica2**는 **server2**에 있습니다. 색인은 샤드1과 샤드2 사이에서 분할됩니다. 요청이 **replica1** 및 **replica2** 모두에 전송됩니다.

server1이 오작동하고 연결이 끊긴 경우 검색 요청이 **server2**의 **replica2**로 자동 라우팅됩니다. 요청은 **replica2**에 의해서만 처리됩니다. **shard1**의 색인 부분을 사용할 수 없습니다.

조합 구성

이 구성은 단일 구성에서 고가용성 및 분포 색인 접근법을 결합합니다. 색인은 여러 샤드에 분산되고, 각 샤드는 여러 복제본에 복제됩니다. 각 복제본을 별도의 서버에 배치하도록 선택하거나 복제본을 그룹으로 결합할 수 있습니다. 여기에는 서버의 각 그룹이 각 샤드에 대해 하나의 복제본을 포함합니다.

각 샤드에 대한 복제본 수에는 제한이 없으므로 여러 복제 그룹을 구성할 수 있습니다.

예제:

shard1에 대한 **replica1** 및 **shard2**에 대한 **replica1**은 server1에서 함께 그룹화됩니다. **shard1**에 대한 **replica2** 및 **shard2**에 대한 **replica2**는 **server2**에서 함께 그룹화됩니다. **server1**이 오작동하고 연결이 끊긴 경우 두 샤드에 대한 모든 검색 요청이 **server2**의 복제본에 의해 수행됩니다. **server2**의 복제본 그룹이 리더가 됩니다.

ZooKeeper 설정

SolrCloud는 로드 밸런서와 Solr 구성 유지를 위해 ZooKeeper가 필요합니다. SolrCloud 환경의 경우 외부 ZooKeeper 앙상블을 설정합니다. 장애 복구 시나리오에 대한 최상의 지원을 제공하려면 대부분 또는 모든 복제 서버에 ZooKeeper를 설치합니다. 적절한 장애 조치를 취하려면 최소 3개의 ZooKeeper가 필요합니다. 추가 확장 시 항상 홀수 개의 노드를 사용해야 합니다. 각 ZooKeeper는 독립 응용 프로그램으로 실행됩니다. 그러나 Solr 시작 스크립트는 사용 가능한 모든 ZooKeepers를 사용합니다.

절차

1. 지원되는 ZooKeeper 버전을 다운로드하고 압축을 풉니다.

2. *ZOOKEEPER_HOME*에 *data* 디렉터리를 생성합니다.

3. *ZOOKEEPER_HOME***conf**\에서 *zoo_sample.cfg*의 이름을 **zoo.cfg**로 바꿉니다.

4. *zoo.cfg*를 열고 **dataDir** 속성을 찾습니다. *data* 디렉터리를 가리키도록 **dataDir** 경로를 수정합니다. *data* 디렉터리는 SolrCloud에 대한 모든 구성 파일을 저장합니다.

   Windows 시스템에서 이중 역슬래시를 사용합니다.

   dataDir=C:\\workdir\\apps\\zookeeper-*version***\\zookeeper-***version***\\data**

© 2024 Siemens

ZooKeeper 앙상블을 생성하는 경우 계속 합니다. 단일 ZooKeeper를 생성하는 경우 **zkServer.cmd를 업데이트**하려면 건너뜁니다.

5. 실행되는 각 ZooKeeper 인스턴스에 대해 하나씩 파일의 맨 아래에 문을 추가합니다. 이 예제에서는 두 개의 서버를 정의합니다.

   서버.*ID*=realdealsolr:2888:3888

   서버.*ID*=realdealdisp:2888:3888

   *서버 ID*에서 ZooKeeper 서버를 식별합니다. 포트 **2888** 및 **3888**은 리더 선택 항목에 대한 기본 통신 포트입니다.

   파일을 저장하고 닫습니다.

6. *myid*라는 이름의 파일을 생성하고 ZooKeeper 서버의 ID를 입력합니다. ID는 *server.ID*를 사용하여 **zoo.cfg**에 정의되어 있습니다(예: ID가 **234**인 **myserver.234**).

   각 ZooKeeper 서버는 자체 *myid* 파일이 필요합니다. 이 파일을 **\data** 폴더에 저장합니다.

7. *ZOOKEEPER_HOME***bin\zkServer.cmd** 스크립트를 엽니다. 다음과 같이 **"-Djute.maxbuffer=5097152"**를 추가합니다.

   ```
   call %JAVA% "-Djute.maxbuffer=5097152" "-Dzookeeper.log.dir=%ZOO_LOG_DIR%"
      "-Dzookeeper.root.logger=%ZOO_LOG4J_PROP%" -cp "%CLASSPATH%"
      %ZOOMAIN% "%ZOOCFG%" %*
   ```

   파일을 저장하고 닫습니다.

8. ZooKeeper를 시작하려면 **zkServer.cmd**를 실행합니다.

9. 각 ZooKeeper 설치에 대해 이 단계를 반복합니다. *myid* 서버 ID를 제외하고 모든 ZooKeeper 서버가 동일해야 합니다.

configset 및 컬렉션 생성

구성에 필요한 샤드 및 복제본의 수를 결정합니다. 다음 샘플 절차는 고가용성 접근과 분포 색인 접근 방식을 결합하고 교체 Solr 컬렉션과 두 개의 샤드를 생성합니다. 데이터는 다시 색인화해야 합니다.

절차

1. 현재 *SOLR_HOME*> **컬렉션**(**collection1**, **collection2**, **admin**, **ProductScopeCollection**)을 임시 디렉터리에 복사합니다.

2. Solr 서버에 기존 Solr 설치의 복사본을 만들어 앞으로 *Leader1_SOLR_HOME*이라고 합니다.

© 2024 Siemens

3. *Leader1_SOLR_HOME\server\etc\webdefault.xml*을 엽니다. **<security-constraint>** 및 **<login-config>**에 대한 문을 설명합니다.

   이 작업은 기본 인증을 비활성화합니다. SolrCloud에서 기본 인증을 사용하려면 **기본 인증 설정**을 참조하십시오.

4. *Leader1_SOLR_HOME\server\solr*의 모든 컬렉션을 삭제합니다.

5. *Leader1_SOLR_HOME***\runSolr.bat**를 열고 다음을 찾습니다.

   call %SCRIPT_DIR%**/bin/solr start -f**

   다음으로 교체합니다.

   %SCRIPT_DIR%**/bin/solr start -f -cloud -s ..\server\solr -p 8983 -z** *hostname*:*port*

   -z 매개변수는 ZooKeeper 기계 및 포트를 지정합니다. 둘 이상의 ZooKeeper가 있는 경우 쉼표로 구분된 리스트(예: **-z server1:2181,server2:2181,server3:2181**)에서 지정합니다.

6. *Leader1_SOLR_HOME* 인스턴스를 시작합니다. Solr이 클라우드 모드에서 시작되었고 기존 컬렉션이 없는지 확인합니다.

7. *Leader1_SOLR_HOME*에 대한 컬렉션을 생성합니다. 다른 명령 프롬프트를 열고 임시 디렉터리에 복사한 각 컬렉션에 대해 다음 명령을 실행합니다.

   solr create_collection -c *collection_name* **-d** *Temp_path******collection_name* **-n** *collection_name* **-shards** *n* **-p 8983 -replicationFactor** *n*

   -shards는 구성에 대한 샤드의 수이고, **replicationFactor**는 구성에 대한 샤드당 복제본의 수입니다. 예를 들면 다음과 같습니다.

   solr create_collection

   -c collection1

   -d c:\SolrTest\collection1

   -n collection1 -shards 2

   -p 8983 -replicationFactor 2

8. 계속하기 전에 **HTTPS를 구성**하여 SSL을 사용하도록 Solr을 구성할 수 있습니다. *Leader1_SOLR_HOME* 의 후속 복제본에 SSL 구성이 포함되도록 *Leader1_SOLR_HOME*에서 이 절차를 수행합니다.

© 2024 Siemens

9. Solr을 중지합니다. 복제본을 호스팅할 모든 서버에 전체 *Leader1_SOLR_HOME* 인스턴스를 복사합니다.

   앞으로 *Leader1_SOLR_HOME*는 원본 인스턴스를, **Solr_Cloud_Home**은 해당 복사본을 나타냅니다.

10. 서버 인프라스트럭처 계획 및 복제본 그룹화 전략을 기반으로 어떤 서버에서 어떤 샤드 및 복제본을 호스팅할 것인지 결정합니다.

    디렉터리는 샤드 번호 및 제어되는 복제 번호로 레이블이 지정됩니다. *Leader1_SOLR_HOME* 및 각 **Solr_Cloud_Home** 서버에서 해당 서버를 제어하려는 *\server\solr*에 있는 디렉터리를 결정합니다.

    각 서버 인스턴스에 고유한 디렉터리가 있고 다른 모든 항목에 중복 항목이 없는지 확인합니다. 중복은 이미 다른 서버에 할당된 디렉토리입니다. 그런 다음 각 서버에서 중복 디렉터리를 삭제합니다.

11. 먼저 인스턴스 *Leader1_SOLR_HOME*을 시작하고 리더가 할당된 순서대로 각 **Solr_Cloud_Home** 인스턴스를 시작합니다.

기본 인증 설정

SolrCloud 구성에 대한 기본 인증을 설정해야 합니다.

절차

1. **https://curl.haxx.se/download.html**에서 **cURL**을 다운로드하고 디렉터리에 압축을 풉니다. **Path** 환경 변수를 *Curl\bin* 디렉터리의 경로로 업데이트합니다.

2. *Leader1_SOLR_HOME\ server\scripts\cloud-scripts*에서 *security.json* 파일을 생성합니다. Solr 관리자에 대해 설정된 암호화된 암호에 대한 **사용자 이름**을 제공합니다.

   ```
   {
   "authentication":{
     "blockUnknown": true,
     "class":"solr.BasicAuthPlugin",
     "credentials":{"username":"..."}
   }
   }
   ```

   이 예제에서는 표준 Solr 암호 **SolrRocks**에 대해 암호화된 암호를 사용합니다.

3. *Leader1_SOLR_HOME\bin*에서 다음 명령을 실행합니다.

   solr zk cp file:*Leader1_SOLR_HOME***\ server\scripts\cloud-scripts\security.json zk:/security.json -z localhost:2181**

4. SolrCloud를 시작합니다.

© 2024 Siemens

5. **cURL**을 실행합니다.

   curl --user solr_admin:SolrRocks http://localhost:8983/solr/admin/authentication -H 'Content-type:application/json' -d "{\"set-user\":{\"solr_admin\":\"*NewPassword*\"}}"

   이 경우 **SolrRocks** 를 *security.json*에 필요한 원래 암호로 지정합니다. 그런 다음 **NewPassword**에서 암호를 변경합니다.

6. Solr 관리자 페이지(**http://hostname:port/solr/**)로 이동하고 방금 설정한 사용자 이름과 암호를 사용하여 로그온합니다.

© 2024 Siemens

2. 검색 구성

검색 기능 설정

검색 제안 구성

전역 검색에서 검색 조건을 입력하면서 제안된 검색 용어를 구성할 수 있습니다. 기본적으로 0.5% 이상으로 색인화된 데이터로 구성된 키워드가 검색 제안으로 표시됩니다. 예를 들어, 1000으로 색인화된 아이템이 있는 경우 제안으로 사용하려면 키워드가 5회 이상 있어야 합니다. 키워드를 검색 제안으로 표시하기 위한 임계값을 구성하거나 제안을 끄도록 구성할 수 있습니다.

키워드 제안 외에 사용자는 입력한 검색 기준에 일치하는 제안되는 구문을 구성할 수 있습니다. 제안 구문은 8단어로 제한되며 검색 기준과 일치하는 상위 5개의 제안 구문이 제안 구문 리스트에 표시됩니다. 임계값 구성은 검색 구문 제안에 적용되지 않습니다.

검색 제안 임계값 구성

색인화된 데이터의 5% 이상을 구성하는 키워드가 검색 제안으로 표시되도록 제안에 대한 기본 구성이 설정됩니다. 예를 들어, 1000으로 색인화된 아이템이 있는 경우 제안으로 사용하려면 키워드가 5회 이상 있어야 합니다. 키워드를 검색 제안으로 제시하기 위한 임계값을 구성할 수 있습니다.

절차

1. Solr 설치 디렉터리에서 **solrconfig.xml** 파일을 찾아서 엽니다.

   TC_ROOT\solr-*version*\server\solr\collection1\conf\solrconfig.xml

2. **solrconfig.xml** 파일에서 다음을 찾으십시오.

   ```
   <float name=threshold”>0.005</float>
   ```

3. 추가적인 검색 제안을 표시하려면 값을 줄이십시오(예: 0.3%가 되도록 **0.003**으로 변경). 더 적은 검색 제안을 표시하려면 값을 높이십시오(예: 0.8%가 되도록 **0.008**로 변경).

4. 파일을 저장합니다.

5. **Solr를 다시 시작합니다**.

검색 제안 끄기

색인화된 데이터의 5% 이상을 구성하는 키워드가 검색 제안으로 표시되도록 제안에 대한 기본 구성이 설정됩니다. 검색 제안을 해제할 수 있습니다.

© 2024 Siemens

절차

1. Solr **solrconfig.xml** 파일을 엽니다.

2. **solrconfig.xml** 파일에서 **tcftssuggest** 구성 섹션을 찾습니다.

```
<requestHandler
class="org.apache.solr.handler.component.SearchHandler"
   name="/tcftssuggest">
    <!-- Request handler for Teamcenter search suggestions -->
    <lst name="defaults">
         <str name="spellcheck">true</str>
         <str name="spellcheck.dictionary">tcftssuggest</str>
         <str name="spellcheck.onlyMorePopular">true</str>
         <str name="spellcheck.count">5</str>
         <str name="spellcheck.collate">true</str>
      </lst>
      <arr name="components">
         <str>tcftssuggest</str>
      </arr>
  </requestHandler>
```

3. 다음과 같이 라인에 주석을 작성합니다.

```
<str>tcftssuggest</str>
```

   파일을 저장하고 닫습니다.

4. **Solr를 다시 시작합니다**.

검색 구문 제안 구성

전역 검색 시, 사용자는 입력한 검색 기준에 일치하는 제안되는 구문을 구성할 수 있습니다. 제안 구문은 8단어로 제한되며 검색 기준과 일치하는 상위 5개의 제안 구문이 제안 구문 리스트에 표시됩니다. 임계값 구성은 검색 구문 제안에 적용되지 않습니다.

절차

1. 이러한 환경설정을 설정하여 검색 구문 제안을 활성화합니다.

   - **AW_search_suggestion_handler**를 **문구**로 설정합니다. 기본값은 **spellcheck**입니다.

   - **AW_IndexedProperties_for_Phrased_Search** 환경설정에 속성 리스트를 추가합니다.

     이름 및 설명 속성 **WorkspaceObject.object_name**과 **WorkspaceObject.object_desc**는 기본적으로 검색 구문으로 구성됩니다.

© 2024 Siemens

2. Teamcenter 명령 프롬프트에서 **tc_solr_schema_gen** 도구를 사용하여 **bmide_modeltool.bat**을 실행합니다.

```
bmide_modeltool -u=user -p=password -g=dba -tool=tc_solr_schema_gen
   -mode=upgrade -target_dir="%TC_DATA%"
```

3. **생성된 스키마와 Solr 스키마를 병합합니다**.

4. **Solr를 다시 시작합니다**.

5. 이미 색인화된 데이터에 구문 제안을 적용하려면 **runTcFTSIndexer** 유틸리티의 **objdata:index** 타스크를 사용하여 모든 데이터를 다시 색인화합니다.

   그렇지 않은 경우 구문 제안이 새 데이터 또는 업데이트된 데이터에 대해 증분적으로 동기화됩니다 (**objdata:sync**).

이러한 제안 구문은 액세스 권한에 상관 없이 모든 사용자가 읽을 수 있습니다.

기본 검색 연산자 설정

AWS_Default_Query_Operator 환경설정을 사용하여 여러 용어를 검색하는 데 적용할 기본 검색 작업을 설정할 수 있습니다. 기본적으로 검색 상자에 여러 개의 용어를 입력하는 경우, 적용되는 기본 연산자는 **AND**입니다.

따라서 **HDD 0500**을 검색하면 **HDD** 및 **0500**가 모두 포함된 결과를 반환합니다 (예: **HDD 0500** 또는 **HDD 050055**).

그러나 **AWS_Default_Query_Operator** 환경설정을 **OR**로 설정하는 경우는 동일한 검색에서 **HDD** 또는 **0500** 중 하나를 반환합니다(예: **HDD 123** 또는 **ABC 0500**).

검색에 자동으로 와일드 카드 추가

더 많은 결과를 내기 위해 기본적으로 **AWC_search_automatic_wildcard** 환경설정으로 * 와일드카드를 검색 단어의 끝에 추가합니다. 기본적으로 **1**으로 설정되며, 별표(*)가 접미사로 추가됩니다.

검색어를 제한하거나 확장하려면 환경설정을 변경하면 됩니다.

0은 검색 기준을 변경하지 않고 유지합니다.

2는 별표 와일드가드를 접두사로 추가합니다.

3은 별표 와일드카드를 접두사 및 접미사로 추가합니다.

다음의 경우에는 환경설정이 무시됩니다.

- 사용자가 큰 따옴표 안에 검색어를 입력하는 경우 (예: **"파트"**).

© 2024 Siemens

- 사용자가 검색을 저장한 경우. 사용자가 저장된 검색을 실행하면 재실행 시 환경설정 값이 적용됩니다.

사용자가 검색 텍스트에 별표를 명시적으로 입력하면 와일드카드 동작이 결합됩니다. 예를 들어, 환경설정 값이 **2**(접두어로 와일드카드 사용)이고 사용자가 **파트***를 입력하면 검색어는 ***파트***가 됩니다.

일반 단어 검색

검색에 *the*, *and*, *a*와 같은 일반적인 단어가 포함되면 검색에서 무시될 수도 있습니다. 정지 단어라고 알려진 이러한 단어는 기본적으로 검색 색인의 전체 크기를 줄이기 위해 색인화되지 않습니다. 사용자가 이러한 정지 단어의 일부 또는 전체를 검색할 수 있도록 허용할 수 있습니다. 무시 단어 리스트는 Solr 문서에 포함되어 있습니다.

https://wiki.apache.org/solr/

AnalyzersTokenizersTokenFilters를 검색하고 **solr.StopFilterFactory**를 찾습니다.

따옴표로 묶인 정지 단어가 포함된 검색 구문으로 조회를 실행하면 다음이 발생합니다.

- 인용 부호로 묶인 검색 구문의 키워드 사이에 정지 단어가 있으면 조회에서 와일드 카드로 바뀝니다.
- 정지 단어가 인용 부호 안에 있는 검색 구문의 시작 또는 끝 부분에서 발생하면 무시됩니다.

정지 단어가 포함된 검색에서 정확한 일치 항목이 필요한 경우 Solr *정지 단어* 기능을 비활성화하거나 Solr 정지 단어 리스트를 편집해야 합니다.

> 주의:
>
> 정지 단어 기능을 비활성화하면 검색 색인이 굉장히 커지고 검색 성능이 저하될 수 있습니다.
>
> 정지 단어를 변경한 경우 다시 색인화해야 합니다.

일반적인 단어에 대해 검색 구성

정지 단어가 포함된 검색에서 정확한 일치 항목이 필요한 경우 Solr *정지 단어* 기능을 비활성화하거나 Solr 정지 단어 리스트를 편집해야 합니다. 정지 단어 기능을 비활성화하면 검색 색인이 굉장히 커지고 검색 성능이 저하될 수 있습니다. 정지 단어를 변경한 경우 다시 색인화해야 합니다.

절차

1. **schema.xml** 파일(예: *TC_ROOT***solr-***version****server\solr\collection1\conf\schema.xml**)로 이동합니다.

2. (선택 사항) 검색에 모든 정지 단어를 포함합니다.

   *stopwords_language.txt*가 포함된 행을 검색하고 주석을 추가하십시오.

© 2024 Siemens

예를 들면 다음과 같습니다.

```
<!--filter class="solr.StopFilterFactory" ignoreCase="true" words="lang/
stopwords_en.txt"/-->
```

3. (선택 사항) 정지 단어 리스트를 편집합니다.

   a. *stopwords_language.txt* 파일을 열고 리스트를 편집합니다.

   b. 파일을 저장하고 닫습니다.

4. **Solr를 다시 시작합니다**.

5. **runTcFTSIndexer** 유틸리티를 사용하여 다시 색인화.

검색에 동의어 추가

Solr를 사용하면 동의어를 검색할 수 있습니다. 예를 들어, **MB**를 검색하는 경우 **megabyte**에 대해서도 검색 결과가 반환됩니다. 동의어의 Solr 사용에 대한 정보는 Solr 설명서에서 찾을 수 있습니다.

https://wiki.apache.org/solr/

AnalyzersTokenizersTokenFilters를 검색하고 **solr.SynonymFilterFactory**를 찾습니다.

동의어 파일에 단어를 추가하여 추가적인 동의어 검색을 활성화할 수 있습니다.

1. Solr **synonyms.txt** 파일(예: *TC_ROOT***solr-***version****version\solr\collection1\conf\synonyms.txt**)로 이동합니다.

2. 한 줄에 다음과 같이 쉼표로 구분하여 동의어를 추가합니다.

   ```
   GB,gig,gigabyte,gigabytes
   ```

3. Solr를 재시작합니다.

규칙 및 사전 필터 정의

리비전 규칙 구성

전역 검색 사용자에 대한 리스트를 채우는 환경설정 세트를 사용하여 리비전 규칙 리스트를 정의합니다. 사용 가능한 리비전 규칙은 Teamcenter에서 관리됩니다. 리비전 규칙 정보에 대해서는 Teamcenter 도움말을 참조하십시오.

© 2024 Siemens

사용자 권한 수준에 맞는 버전 제공

사용자 검색에 리비전 규칙이 포함되면 사용자가 검색 조건 및 리비전 규칙 모두에 일치하는 아이템에 액세스할 수 없는 경우가 있습니다. 이 경우에 사용자가 개체 최신 버전을 보고자 하는 경우가 발생할 수 있습니다. 사용자 액세스 권한을 충족하는 기존 버전을 제공하도록 전역 검색을 구성할 수 있습니다.

예를 들어, 엔지니어는 두 리비전이 있는 개체 최신 리비전을 보고자 할 수 있습니다. 첫 번째 리비전은 릴리스되어 모든 사용자가 읽을 수 있지만, 두 번째는 작업 진행 중이므로 설계자만 사용할 수 있습니다. 엔지니어가 설계자 그룹에 없는 경우 리비전 규칙이 액세스 규칙 전에 적용되므로 검색을 수행해도 개체를 반환하지 못합니다.

하지만 먼저 검색 데이터에 액세스 규칙을 적용한 후 결과의 하위 집합에 리비전 규칙을 적용하도록 검색을 구성할 수 있습니다. 이 프로세스에서는 환경설정 및 임계값을 설정해야 합니다. 읽기 액세스 권한 후 리비전 규칙을 적용할 때를 결정하려면 다음 지침을 사용합니다. 임계값은 모든 검색 포스트프로세스 전에 적용됩니다.

- 검색 결과 수가 임계값보다 더 많으면 검색 조건과 일치하는 모든 아이템의 모든 허용 리비전이 검색 결과로 반환됩니다. 결과 위의 검색 결과 이동 경로는 각 아이템의 여러 리비전이 검색 결과에 있음을 나타내는 메시지를 표시합니다.

- 검색 결과 수가 임계값보다 작거나 같으면 검색에서 먼저 사용자의 액세스 권한을 적용한 다음, **AW_Search_check_read_access_for_RevRule**에 의해 구성된 리비전 규칙을 적용합니다. 따라서 검색을 수행하면 사용자에게 허용된 이전 리비전을 찾아서 아이템의 해당 리비전을 반환합니다.

임계값 아래에서 적용할 리비전 규칙을 지정할 수 있습니다. 적용할 리비전 규칙을 결정하여 **AW_Search_check_read_access_for_RevRule** 환경설정에서 지정합니다(기본적으로 비어 있음).

리비전 규칙을 지정할 때 지정된 리비전 규칙을 적용하기 전에 액세스 규칙을 적용하는 검색 동작을 활성화할 수도 있습니다. 검색에서 이 방법을 활성화하면 이미 재색인화하지 않은 경우 Active Workspace의 현재 버전을 사용하여 데이터를 다시 색인화합니다.

> 주의:
>
> 리비전 규칙 검색 동작은 SolrCloud의 샤딩 기능에서 지원되지 않습니다. 이 검색 동작은 SolrCloud에서 분할에 대한 복제 기능에 영향을 미치지 않습니다.

지원되는 리비전 규칙 항목 및 절

지원되는 리비전 규칙 항목 및 리비전 규칙 절을 검색에 적용할 수 있습니다. 리비전 규칙에 지원되는 절과 지원되지 않는 절이 모두 포함되어 있으면 지원되는 절이 적용되고 지원되지 않는 절은 무시됩니다.

리비전 규칙 항목	리비전 규칙 절
Working	**Working( )** **Working(Owning User = Current)** **Working(Owning User = <user_id>)**

© 2024 Siemens

리비전 규칙 항목	리비전 규칙 절
	Working(Owning Group = Current)
	Working(Owning Group =<group_name>)
	Working(Owning User = Current, Owning Group = Current)
	Working(Owning User = Current, Owning Group = <group_name>)
	Working(Owning User = <user_id>, Owning Group = Current)
	Working(Owning User = <user_id>, Owning Group = <group_name>)
Status	**Has Status(Any Release Status, Configured Using Released Date)**
	Has Status(<status_name>, Configured Using Released Date)
	Has Status(<status_name>, Configured Using Effective Date)
	이 리비전 규칙을 사용하려면 **적용일자/단위 날짜**가 구현되었는지 확인합니다.
Latest	**Latest (Creation Date)**
	Latest (Alphanumeric Rev ID)
	Latest (Numeric Rev ID)
	Latest (Alpha+Numeric Rev ID)

Active Workspace 및 Teamcenter Rich Client의 검색 결과 수가 일치하지 않습니다.

Active Workspace에서는 현재 리비전 규칙에 일치하는 리비전만 반환되므로 개수가 Teamcenter Rich Client에서 반환된 개수와 일치하지 않을 수 있습니다. 모든 리비전을 보려면 다음을 수행하십시오.

- Active Workspace에서, 리비전 규칙을 **Precise만**으로 설정합니다. 그러면 개체 데이터에 대한 모든 리비전이 반환됩니다.
- 모든 실패한 개체 및 보류 중인 개체가 색인화되었는지 확인합니다. 실패한 개체 및 보류 중인 개체를 결정하려면 **runTcFTSIndexer** 유틸리티를 사용합니다.

  ```
  runTcFTSIndexer -task=objdata:show n uid_file_dir
  ```

  인수 *n*에 대해 **1**을 지정하여 보류 중인 개체를 가져오거나 **3**을 지정하여 실패한 개체를 가져옵니다.

검색 시 리비전 규칙 전에 액세스 규칙 적용

AW_Search_check_read_access_for_RevRule 환경설정에서 리비전 규칙을 지정할 때 지정된 리비전 규칙을 적용하기 전에 액세스 규칙을 적용하는 검색 동작을 활성화할 수도 있습니다. 검색에서 이 방법을 활성화하면 이미 재색인화하지 않은 경우 Teamcenter web client의 현재 버전을 사용하여 데이터를 다시 색인화합니다.

© 2024 Siemens

절차

적용할 동작을 결정하는 결과 수에 대한 임계값을 설정합니다. 그런 다음 개체를 다시 색인화합니다.

1. *TC_ROOT\solr-버전\server\solr\collection1\conf\solrconfig.xml*에서 임계값을 구성합니다.

   ```
   <int name="RevsionRuleFilter.Threshold">20000</int>
   ```

   임계값은 기본적으로 **10000**으로 설정됩니다.

2. 진행 중인 동기화 흐름을 중지하고 Solr을 중지합니다.

3. *TC_ROOT\solr-버전\server\solr\collection1\conf\schema.xml*의 백업 사본을 만듭니다.

4. **schema.xml**의 **field** 항목을 업데이트하여 **docValues="true"**를 설정합니다.

   ```
   <field ... name="TC_RevisionAnchor" ... docValues="true" ... type="string"/>
   <field ... name="TC_PRIV_am_rule_str" ... docValues="true" ... type="string"/>
   <field ... name="TC_PRIV_matchingRevisionSelectors" ... docValues="true" ...
   type="string"/>
   ```

5. Solr를 다시 시작합니다.

6. 색인화를 시작합니다.

   ```
   runTcFTSIndexer -task=objdata:index
   ```

   이전에 Teamcenter web client의 동일한 버전을 사용하여 개체 데이터를 색인화한 경우 이 단계를 건너뛸 수 있습니다.

7. 색인화가 완료된 후 동기화를 시작합니다.

   ```
   runTcFTSIndexer -task=objdata:sync -interval=300
   ```

리비전 규칙에 대한 적용일자 구현

아이템 리비전 또는 기타 유사한 개체에 대한 전역 검색에 적용일자를 적용하는 리비전 규칙을 지정할 수 있습니다. 적용일자는 아이템 리비전의 복합 속성에 정의됩니다. 적용일자가 없으면 기본값은 **오늘**입니다.

절차

1. Business Modeler IDE에서 적용일자/단위를 저장할 복합 속성을 생성합니다. 색인화 가능으로 표시해야 합니다.

© 2024 Siemens

2. Business Modeler IDE에서 전역 상수 **Awp0EffectivityDatesPropertyName**에 대한 값을 사용자가 생성한 복합 속성 이름으로 설정합니다.

3. *TC_ROOT***TcFTSIndexer****conf***TcFTSIndexer_objdata.properties* 파일에서 **objdatasaxtransformstep.EffectivityDatePropertyName**에 대한 값을 사용자가 생성한 복합 속성 이름으로 설정합니다.

4. 아이템에 대한 리비전 규칙을 생성 또는 편집하고 **날짜** 항목을 정의합니다. 규칙 조건이 적용되는 날짜를 지정합니다. 전역 검색 사용자가 적용일자/단위를 선택하는 것을 도우려면 규칙 이름에 적용일자/단위를 포함시킵니다.

5. **bmide_modeltool** 유틸리티를 사용하여 Solr 스키마 파일을 재생성하고 변경 사항을 병합합니다. Solr를 다시 시작합니다.

6. 복합 속성에 대해 업데이트하려면 다시 색인화합니다.

   단일 아이템 리비전에 대해 여러 적용일자/단위가 활성화된 경우(예: 여러 상태로 인해) 최신 아이템만 색인화됩니다.

7. 리비전 규칙이 **AWC_Rev_Rule_List** 환경설정에 지정되어 전역 검색에 사용 가능하도록 해야 합니다.

검색 사전 필터 정의

검색 사전 필터는 전역 검색 상자 옆에 표시되며 사용자가 전역 검색을 수행할 때 데이터를 필터링하는 데 사용되는 속성입니다. 검색 결과를 필터링하는 데 사용할 수 있는 구속조건 리스트를 보려면 사전 필터를 클릭하십시오. 기본 전역 검색 사전 필터는 개체 카테고리를 기반으로 사전 필터링을 적용하는 **모든 카테고리**입니다.

다음 예에서는 **모든 소유자** 및 **모든 카테고리**가 검색 사전 필터입니다. 사전 필터는 단일 필터 유형만 지정할 수 있습니다. **모든 소유자** 검색 필터는 **AWS_SearchPreFilter_Property1** 환경설정을 다음과 같은 값으로 설정했을 때 사용할 수 있습니다.

```
POM_application_object.owning_user
user-name ( user-id )
```

*사용자-이름*을 지정할 때 폼을 사용합니다.

```
LastName, FirstName ( username )
```

예, `Karia, Pamela ( pkaria )`

© 2024 Siemens

> 주:
>
> 예시 내 공백이 필요합니다. 성 다음에 공백과 쉼표를 넣고 이름을 배치한 후, 괄호 안에 사용자 이름을 넣습니다.

사용자가 검색에 사전 필터를 적용하면 선택 항목이 **AWS_SearchPreFilter_Property1_SelectedValue** 및 **AWS_SearchPreFilter_Property2_SelectedValue** 환경설정에 저장됩니다. 사용자가 로그오프한 후 다시 로그인하면 이러한 환경설정에서 사전 필터를 가져와 클라이언트에 표시합니다.

다음과 같이 환경설정에서 키워드를 사용할 수 있습니다.

- 날짜 속성의 경우 **$TODAY**, **$THIS_WEEK** 및 **$THIS_MONTH**를 사용합니다.
- 값을 *user-name*(*user-ID*)으로 저장하는 속성에 대해 **$ME**를 사용합니다.
- 그룹 이름을 저장하는 속성의 경우 **$MY_GROUP**을 사용합니다.

  이러한 키워드는 검색을 수행할 때 올바른 값으로 대체됩니다.

다음 예에서는 **최종 수정일** 사전 필터를 지정합니다. 기본 옵션은 **$TODAY**, **$THIS_WEEK** 및 **$THIS_MONTH**입니다.

```
POM_application_object.last_mod_date
```

다음 예에서는 **그룹 ID** 사전 필터에 대한 옵션으로 **엔지니어링** 및 **관리자** 그룹을 지정합니다.

```
POM_application_object.owning_group
Engineering
Admins
```

카테고리 사전 필터 사용자 정의

모든 카테고리 사전 필터 리스트의 값은 관리자가 정의한 대로 관련 비즈니스 개체별로 그룹화됩니다. Teamcenter Business Modeler IDE을 사용하여 카테고리를 사용자 정의할 수 있습니다.

1. Business Modeler IDE에서 사이트의 비즈니스 템플릿을 엽니다.
2. 카테고리에 개체를 할당하려면 개체의 **Awp0BusinessObjectCategories** 비즈니스 개체 상수 값을 카테고리 이름으로 설정합니다. 카테고리는 하나 이상의 개체를 이 상수로 지정할 때 생성됩니다.

   둘 이상의 카테고리에 개체를 할당하려면 각 카테고리 이름을 쉼표(공백으로 구분 안 함)로 구분하여 나열합니다. 예를 들어, 개체의 **Awp0BusinessObjectCategories** 값을 **Documents,Files**로 설정하면 **문서**와 **파일** 카테고리에 할당됩니다.

   카테고리에서 개체를 제거하려면 해당 카테고리 이름을 개체의 **Awp0BusinessObjectCategories** 값에서 제거합니다.

© 2024 Siemens

개체의 모든 자식 개체는 동일한 카테고리로 자동 할당됩니다. **Awp0BusinessObjectCategories** 값을 업데이트하여 자식에 대해 재정의할 수 있습니다.

다음 개체는 기본적으로 다음 카테고리에 할당됩니다.

파일	문서	파트
MSExcelX MSPowerPointX MSWordX PDF 텍스트 JPEG	DocumentRevision RequirementSpec 리비전	DesignRevision PartRevision

3. 카테고리도 추가하는 경우 카테고리 사전 필터 환경설정에 카테고리 이름을 추가합니다. 예를 들어, 환경설정 **AWS_SearchPreFilter_Property2** 를 사용해 **모든 카테고리**를 검색 사전필터로 정의하려면 환경설정에 정의된 카테고리 리스트에 카테고리 이름을 추가 합니다.

4. 사용자 정의가 완료되면 사이트에 템플릿을 배포합니다.

5. **bmide_modeltool.bat** 유틸리티를 실행합니다.

```
bmide_modeltool.bat -u=username -p=password -g=dba -tool=all -mode=upgrade
-target_dir="TC_DATA"
```

새 템플릿을 배포하거나 **bmide_modeltool.bat** 유틸리티를 실행하지 않고 카테고리를 정의할 수 있는 다른 방법이 있습니다. Siemens Digital Industries Software에서는 사이트를 업데이트하기 위해 이 방법을 사용하는 경우 각별히 주의할 것을 권장합니다. 이 절차를 수행하기 전에 설정을 백업해 둡니다.

이 방법을 사용하여 분류된 개체의 자식은 이 방법을 사용하여 정의된 카테고리에 자동으로 속하지 않습니다. 해당 카테고리를 변경하려면 각 자식 개체를 수동으로 업데이트해야 합니다. 사용자가 변경 내용을 보려면 세션을 다시 시작해야 합니다.

1. Teamcenter Rich Client를 엽니다.

2. **편집→옵션**을 선택하고 **AW_FullTextSearch_TypeCategories** 환경설정을 찾습니다. **값** 필드를 모든 카테고리의 전체 값으로 설정합니다. 예를 들어, **파일**, **문서**, **파트** 및 **변경** 카테고리의 기본값은 다음과 같이 설정할 수 있습니다.

```
Files:MSExcelX,MSPowerPointX,MSWordX,PDF,Text,JPEG
Documents:DocumentRevision,RequirementSpec Revision
Parts:DesignRevision,PartRevision
Changes:ChangeItemRevision
```


© 2024 Siemens

경고:

이 방법으로 카테고리를 변경하면 예기치 않은 결과가 발생할 수 있습니다. 개체의 내부 이름은 암호화되며 **값** 필드 내용에 대해서는 맞춤법 또는 기타 오류 검사가 수행되지 않습니다.

카테고리 이름은 로컬화되어 있지 않습니다.

사전 필터를 사용하여 검색 결과를 생성하는 경우 성능 향상

AWS_Search_disable_exclude_prefilter 환경설정을 사용하여 사전 필터 카테고리에서 선택한 사전 필터에 대한 결과만 생성할 수 있습니다. 선택한 사전 필터는 **필터** 패널에서 사전 필터 카테고리에 대해 표시되는 유일한 필터입니다.

사용자의 사전 필터 선택에 관계없이, **필터** 패널에서 검색 결과와 일치하는 사전 필터 카테고리의 모든 파셋을 생성하려면 환경설정을 **false**(기본값)로 설정합니다. 반환되는 결과의 수가 많으면(예: * 와일드 카드 입력) 시간이 더 걸릴 수 있지만 사용자에게 더 많은 유연성을 제공합니다.

필터 패널의 파셋을 생성하기 전에 사전 필터 선택 항목을 검색 결과에 적용하려면 환경설정을 **true**로 설정합니다. 이렇게 하면 결과의 수가 많을 때 성능이 향상되지만 카테고리에 대해 선택된 사전 필터로 범위가 제한됩니다.

예제:

사용자는 **유형** 사전 필터 리스트에서 **파트 리비전**을 선택하고 검색 조회를 입력합니다.

AWS_Search_disable_exclude_prefilter가 **true**로 설정된 경우, 검색은 **유형** 카테고리의 **파트 리비전** 사전 필터와 일치하는 결과만 반환합니다.

AWS_Search_disable_exclude_prefilter가 **false**로 설정된 경우, 검색은 **유형** 카테고리의 모든 사전 필터와 일치하는 모든 결과를 반환합니다.

사용자 정의 유형 사전 필터 유지

전역 검색 **유형** 사전 필터를 사용자 정의했거나 이전 버전의 Teamcenter web client에 포함된 **유형** 사전 필터를 선호하는 경우 web client를 업데이트할 때 해당 사전 필터를 유지할 수 있습니다.

절차

1. Teamcenter **preferences_manager** 유틸리티를 실행하여 **AWS_SearchPreFilter_Property2** 환경설정을 내보냅니다. 예를 들면 다음과 같습니다.

```
preferences_manager -u=user -p=password -mode=export
   -preferences=AWS_SearchPreFilter_Property2 -out_file=c:\temp\prefilter.xml
```

2. Teamcenter web client 업그레이드를 수행합니다.

3. Teamcenter **preferences_manager** 유틸리티를 실행하여 **AWS_SearchPreFilter_Property2** 환경설정을 가져옵니다. 예를 들면 다음과 같습니다.

© 2024 Siemens

```
preferences_manager -u=user -p=password -mode=import
   -preferences=AWS_SearchPreFilter_Property2 -out_file=c:\temp\prefilter.xml
-action=OVERRIDE
```

개체 집합에 대한 검색 사전 필터 구성

컨텍스트 검색에 적용하도록 개체 집합에 사전 필터를 구성할 수 있습니다. 사용자가 개체 집합에 추가할 경우, 구성된 사전 필터가 구성된 속성 및 해당 값을 사용하여 검색 결과를 필터링합니다. 또한, 사용자는 사전 필터를 쉽게 제거하고 다른 필터를 계속 사용할 수 있습니다.

사전 필터 **searchFilter** 매개변수는 **추가** 개체 패널을 시작하는 명령에 대해 설정된 매개변수에서 검색됩니다.

개체 집합 XRT의 명령에서 `parameter name="searchFilter"`를 지정합니다. 다음 형식을 사용하여 **searchFilter** 매개변수 값을 두 번째 개체의 값과 속성의 조합으로 지정합니다.

유형.속성1=값1 연산자 유형.속성2=값2

유형 및 *속성*은 대소문자를 구분하는 내부 이름입니다.

연산자:

AND는 둘 모두 사전 필터를 사용하는 키워드입니다.

TO는 날짜 및 숫자 범위에 대한 시작 및 끝 값을 연결하는 키워드입니다.

사전 필터 사이에 선행 및 후행 공백이 허용됩니다.

예제:

```
<objectSet>
  <command id="AddNew">
   <parameter name="searchFilter"
    value="WorkspaceObject.object_type=Fnd0LogicalBlockRevision
    AND POM_application_object.owning_user=Zhu, Ray ( zhur )"/>
  </command>
</objectSet>
```

지원되는 필터 유형은 **문자열**, **날짜**, **날짜 범위**, **숫자** 및 **숫자 범위**입니다.

문자열 예시	POM_application_object.owning_user = Engineer,Ed (ed)
날짜 예시	POM_application_object.last_mod_date_0Z0_year= 2016 POM_application_object.last_mod_date_0Z0_year_month= September 2016

© 2024 Siemens

	POM_application_object.last_mod_date_0Z0_year_month_day= Sunday - Sep 25, 2016
날짜 범위 예시	POM_application_object.last_mod_date= 2016-08-07T20:00:00-04:00 TO 2017-08-09T19:59:59-04:00 POM_application_object.last_mod_date= 2016-08-07 TO 2017-08-09 POM_application_object.last_mod_date= * TO 2017-08-09 POM_application_object.last_mod_date= 2016-08-07 TO *
숫자 예시	WorkspaceObject.s2clAverageRatingFmSy=2
숫자 범위 예시	WorkspaceObject.s2clAverageRatingFmSy=2 TO * WorkspaceObject.s2clAverageRatingFmSy=* TO 100

검색 일치 지정

검색 결과 부스팅

사용자에 대한 검색 결과 관련성은 각 결과에 대한 점수를 계산하여 결정됩니다. 먼저 초기 점수가 생성됩니다. 그런 다음 특정 조건에 따라 각 점수를 높일 수 있습니다. 채점이 완료되면 다음을 볼 수 있습니다.

- 정확히 일치하는 속성이 부분적으로 일치하는 것보다 우선합니다.
- 일치하는 속성은 파일 내용의 정확한 일치보다 우선적으로 적용될 수 있습니다.
- 특정 개체 데이터 속성에 대해 구성된 부스팅.

다음을 기반으로 검색 결과 부스팅을 변경할 수 있습니다.

- 비즈니스 속성
- 소유권(사용자, 그룹 및 프로젝트를 기반으로 함)
- 최종 수정 날짜 또는 생성 날짜
- 단어 근접성

일부 일치하는 결과가 리스트에서 더 높은 점수가 매겨질 수 있는데, 검색 조건과 일치하는 모든 결과가 반환됩니다. 한 개체의 초기 점수가 다른 개체의 높아진 점수를 초과할 수도 있습니다. 따라서 구성을 테스트하여 검색 결과에 대한 부스팅 전략의 효과를 확인해야 합니다.

© 2024 Siemens

속성을 기반으로 부스팅 구성

AWS_Preferred_Attributes 환경설정에 지정된 속성을 기반으로 부스팅을 구성합니다. 부스팅을 위해 고려할 개체 속성을 지정합니다. 검색 조건이 개체 속성과 일치하는 경우, 일치하는 속성이 있는 반환된 개체가 검색 결과 리스트에서 더 높은 위치에 표시될 수 있습니다. 다음 형식에 따라 비즈니스 개체 및 속성을 지정하십시오.

business-object-type.property-name

쉼표로 구분된 여러 속성을 지정합니다. 기본적으로 아이템 ID가 부스팅에 고려됩니다.

ItemRevision.awp0_Item_item_id

LOW, **MEDIUM** 또는 **HIGH** 부스팅 우선순위 값 중 하나를 추가할 수 있습니다. 속성에 대해 부스팅 우선순위를 지정하지 않으면 기본값은 **MEDIUM**입니다. 예:

ItemRevision.awp0_Item_item_id:MEDIUM

이 환경설정에 의해 지정된 비즈니스 개체 및 속성은 **Awp0SearchIsIndexed** 속성 상수를 사용하여 색인화 가능으로 표시되어야 합니다.

소유권을 기반으로 부스킹 구성

AWS_Preferred_Attributes 환경설정을 사용하여 소유권을 기반으로 부스팅을 구성합니다. 검색 결과 리스트에서 부스팅에 고려할 사용자, 그룹 또는 프로젝트를 설정합니다.

LOW, **MEDIUM** 또는 **HIGH** 부스팅 우선순위 값 중 하나를 속성에 할당할 수 있습니다. 속성에 대해 부스팅 우선순위를 지정하지 않으면 기본값은 **MEDIUM**입니다. 소유권 일치 항목은 결과 리스트에서 더 높은 위치에 표시될 수 있습니다.

현재 또는 최신 사용자

owning_user 또는 **last_modified_user**에 대해 유효한 사용자 이름 또는 변수 **$ME**(기본값)를 지정합니다. 여러 사용자를 할당하려면 동일한 속성에 대해 여러 항목을 생성하고 각각에 대해 사용자를 지정합니다.

POM_application_object.owning_user:LOW:*user-name*

POM_application_object.last_modified_user:HIGH:$ME

현재 사용자의 그룹

user_group에 대해 유효한 그룹 이름 또는 변수 **$MY_GROUP**(기본값)을 지정합니다. 여러 그룹을 할당하려면 동일한 속성에 대해 여러 항목을 생성하고 각각에 대해 그룹을 지정합니다.

POM_application_object.owning_group:HIGH:*user-group*

POM_application_object.owning_group:HIGH:$MY_GROUP

현재 사용자의 현재 프로젝트

user_project에 대해 유효한 프로젝트 이름 또는 변수 **$MY_PROJECT**(기본값)를 지정합니다. 여러 프로젝트를 할당하려면 동일한 속성에 대해 여러 항목을 생성하고 각각에 대해 프로젝트를 지정합니다.

© 2024 Siemens

WorkspaceObject.project_list:LOW:*user-project*

WorkspaceObject.project_list:LOW:$MY_PROJECT

날짜를 기반으로 부스킹 구성

AWC_Search_Results_Recency_WeightAge 환경설정을 사용하여 날짜를 기반으로 결과 부스팅을 구성합니다. 가장 최근에 생성되거나 수정된 개체를 부스팅하려면 숫자 값을 점진적으로 높게 설정합니다. 최근 생성 날짜 또는 수정 날짜와 일치하는 항목이 결과에서 더 높은 위치에 표시될 수 있습니다. 값을 **0**(날짜 기준 부스트 끄기)에서 **6**(최근 날짜 기준 부스트 최대화)으로 설정합니다.

근접도를 기반으로 부스킹 구성

AWC_Search_Results_Word_Proximity 환경설정을 사용하여 검색어의 근접도에 따라 결과 부스팅을 구성합니다. 개별 검색어의 최대 근접도에 대한 숫자 값을 설정합니다. 근접도 설정 내에서 발생하는 검색어와 일치하는 항목이 결과에서 더 높은 위치에 표시될 수 있습니다. 값을 **0**(근접도에 따라 부스팅 끄기) ~ **100**까지 설정합니다. 기본값은 **10**입니다.

예를 들어, 값이 **5**인 경우 용어가 5개 이상의 단어로 구분되지 않는 결과가 더 높은 위치에 표시될 수 있습니다.

결과 임계값 구성

굉장히 많은 결과를 반환하는 검색어를 입력할 수 있으며, 이로 인해 대기 시간이 길어질 수 있습니다. 결과가 상세 리스트 한계를 초과하는 경우에는 검색 조건을 세분화하여 더 적은 결과를 반환할 수 있습니다. **AWC_Search_Threshold_Value** 환경설정에서 최대 결과 수를 지정하면 반환되는 결과의 수를 제한할 수 있습니다.

결과 수가 한계를 초과하면 결과는 표시되지 않습니다. 결과가 한계를 초과한다는 메시지가 사용자에게 표시됩니다. 부스팅, 보안 액세스, 리비전 규칙 검사 또는 차트 등의 다른 검색 작업은 결과에 적용되지 않습니다. 검색 조건을 변경하거나 필터를 선택하여 검색 범위를 좁힐 수 있습니다.

비활성화하려면 임계값 한계에 해당하는 정수 또는 **0**(기본값)을 지정합니다.

저장된 검색을 실행하거나 필터 패널에서 필터를 적용하는 경우에는 임계값 한계가 결과에 적용되지 않습니다. 기본적으로 **카테고리** 및 **유형** 필터가 항상 표시됩니다.

데이터 집합에 대해 부모 비즈니스 개체 반환

전역 검색과 일치하는 데이터 집합 내용과 관련된 부모 비즈니스 개체를 확인해야 할 수 있습니다. 검색 조건이 비즈니스 개체에 첨부된 데이터 집합 파일 내용과 일치하는 경우 해당 비즈니스 개체가 검색 결과에 반환되도록 구성할 수 있습니다. 비즈니스 개체가 색인화되면 연관 데이터 집합의 파일 내용이 색인화됩니다. 검색이 파일 내용과 일치하면 데이터 집합 대신 비즈니스 개체가 반환됩니다. 이 동작은 특정 유형 수준 또는 전체 유형 계층에 적용할 수 있는 비즈니스 개체 상수에 의해 구성됩니다.

© 2024 Siemens

파일 내용은 UTF-8 언어일 수 있습니다. 필터 패널에는 사용자 필터링을 위한 비즈니스 개체 및 데이터 집합의 공통 속성을 표시할 수 있습니다.

- 비즈니스 개체 또는 그 하위 유형은 하나 이상의 데이터 집합과 동일한 관계 유형을 가져야 합니다. 데이터 집합에는 하나 이상의 파일을 포함할 수 있습니다.

- **Awp0SearchDatasetIndexingBehavior** 상수는 데이터 집합을 비즈니스 개체로 인라인으로 색인화할지 정의합니다.

- **Awp0DatasetTypeToBeIndexedInline** 상수는 비즈니스 개체로 인라인으로 색인화할 데이터 집합 유형을 식별합니다. 데이터 집합이 색인화 가능한 것으로 표시되면 독립형 데이터 집합이 해당 파일 내용과 함께 색인화됩니다. 인라인 색인화 방법만 사용하려면 데이터 집합 유형을 색인화 가능으로 표시하지 마십시오.

  예를 들어, **Awp0DatasetTypeToBeIndexedInline**을 사용하여 유형에 대해 지정된 첨부 관계가 있는 PDF 문서를 반환하려면 다음을 수행합니다.

  INHERIT:TC_Attaches:PDF

  PDF로 데이터 집합을 색인화할 때는 **TC_Attaches** 관계 유형에서 참조되는 1차 비즈니스 개체 UID가 색인화됩니다. 검색 결과와 PDF의 내용이 일치하면 해당 PDF 데이터 집합과 함께 **TC_Attaches** 관계 유형에서 참조되는 부모 비즈니스 개체가 반환됩니다.

  동일한 개체에 대해 **INHERIT** 및 **NO_INHERIT** 규칙을 모두 지정할 수 있습니다.

  INHERIT:*:MSWORDX, NO_INHERIT:IMAN_specification:PDF

  NO_INHERIT:TC_Attaches:PDF, INHERIT:IMAN_specification:HTML~MSWordX

  INHERIT 규칙은 정의된 유형 및 해당 하위 유형에 적용됩니다. **NO_INHERIT** 규칙은 정의된 유형에만 적용됩니다.

  데이터 집합에 대해 인라인 색인화를 선택한 경우 많은 유형 간에 공유되는 참조 데이터 집합을 색인화하지 않도록 할 수 있습니다. 예를 들어, 특정 재료를 설명하는 PDF는 이를 사용하는 1000개의 파트에 첨부될 수 있습니다. 효율성과 성능을 향상시키려면 해당 유형을 가장 잘 나타내는 데이터 집합 및 관계만 포함하도록 색인을 세분화합니다.

- 데이터 집합 파일 내용을 색인화하려면 **AWS_FullTextSearch_Index_Dataset_File_Content** 환경설정을 **on**으로 설정해야 합니다.

- 속성 파일에서 **objdataloadstep.indexStandaloneDatasets** 속성을 구성하여 부모 비즈니스 개체 외에도 검색 결과의 데이터 집합을 반환합니다.

- **AW_FTSIndexer_skip_modifications_via_relations**는 **TcFTSIndexer**가 동기화 색인화 플로 동안 관계 조회를 실행하여 유형에 대한 수정 사항을 찾을 수 있는지 여부를 제어합니다. 예를 들어 데이터 집합 개체가 **DocumentRevision** 유형 또는 해당 하위 유형에 첨부되어 있으며 이미 색인화된 경우, 동기화

© 2024 Siemens

색인화 플로를 실행하기 전에 해당 환경설정을 **false**로 설정합니다. 기본값은 **true**이며, 조회를 건너뜁니다.

고려할 요소:

- 데이터 집합 파일 내용의 인라인 색인화를 사용하려면 전체 데이터 집합 파일 세트를 다시 색인화해야 합니다.
- 사용자 권한의 유효성을 각 개체에 대해 검사합니다. 사용자 권한은 데이터 집합을 필터링할 수 있지만 여전히 문서 리비전을 반환합니다.
- 부모 비즈니스 개체 자체에서 또는 그 일부로 색인화하지 않을 내용이 있는 데이터 집합의 경우, 사용자 정의 필터를 생성하여 필터링할 수 있습니다.

예제

이들 예시에서는 사용자의 검색 결과에서 데이터 집합의 가시성을 구성하는 방법을 보여줍니다.

먼저 인라인 색인화에 대해 색인화가 구성되었는지 확인해야 합니다.

- **Awp0SearchDatasetIndexingBehavior** 상수를 **Inline**으로 설정합니다.
- 색인화할 파일 유형에 대해 **Awp0DatasetTypeToBeIndexedInline** 상수를 구성합니다.
- **AWS_FullTextSearch_Index_Dataset_File_Content** 환경설정을 **true**로 설정합니다.

사용자의 검색 결과에서 부모 비즈니스 개체와 첨부된 데이터 집합을 모두 반환하려면 **objdataloadstep.indexStandaloneDatasets** 속성을 **true**로 설정하십시오. 데이터 집합은 독립형 모드로 색인화됩니다.

사용자의 검색 결과에 부모 비즈니스 개체만 반환하려면 **objdataloadstep.indexStandaloneDatasets** 속성을 **false**로 설정하십시오. 데이터 집합이 인라인으로 색인화됩니다. 독립형 데이터 집합 내용 색인화는 원하는 데이터 집합 유형에 대한 **Awp0SearchIsIndexed** 상수를 변경하여 비활성화할 수도 있습니다. **Awp0SearchIsIndexed** 상수가 변경되지 않고 **objdataloadstep.indexStandaloneDatasets** 속성이 **false**로 설정되어 있는 경우 독립형 데이터 집합 내용 색인화가 비활성화됩니다.

전역 검색을 위해 반환할 개체 유형 구성

특정 개체 유형을 포함하지 않도록 전역 검색 결과를 제한할 수 있습니다. 유형은 **AWC_Global_Search_Suppress_Types_From_Results** 환경설정에서 지정해야 합니다. 검색은 검색 결과에서 지정된 개체 유형 및 해당 필터 속성을 생략합니다.

환경설정에서 표시하지 않으려는 개체 유형에 대한 내부 이름 하나 이상을 지정합니다. 유형 계층은 지정된 개체 유형에 적용됩니다. 예를 들어, 환경설정 값 중 하나가 **데이터 집합**인 경우, 데이터 집합 유형(예: PDF, Microsoft Word 및 텍스트), 해당 자식 유형 및 필터 속성이 검색 결과에서 표시되지 않습니다.

© 2024 Siemens

검색 결과 구성

필터 패널 동작 구성

검색 **필터** 패널에는 전역 검색 결과의 아이템과 관련된 속성 및 값이 표시됩니다. 필터 카테고리 및 값을 선택하여 결과를 좁힐 수 있습니다. 관리자는 사용자가 필터를 쉽게 찾아 적용할 수 있도록 필터 동작을 구성할 수 있습니다. 사용자는 **검색 설정** 패널을 사용하여 이러한 환경설정 일부를 직접 변경할 수 있습니다.

타이프 어헤드 동작 구성

AW_DisableTypeAheadFacetSearch 환경설정을 사용하면 사용자가 🔍를 클릭하면 필터 검색을 시작할지 또는 사용자가 검색어를 입력할 때 필터 결과를 반환할지 구성할 수 있습니다. 기본값 **false**는 사용자가 입력하면 결과를 표시합니다.

AW_TypeAheadFacetSearch_Delay 환경설정을 사용하면 사용자가 검색 필드에 입력을 시작할 때부터 결과가 표시되기 시작할 때까지의 시간을 제어할 수 있습니다. 밀리초 단위로 양의 정수를 지정합니다. 기본값은 **500**입니다.

일치하는 필터 값에 대한 와일드카드 동작 구성

AWC_search_filter_wildcard 환경설정을 사용하여 검색 동작에서 와일드카드 사용을 구성합니다.

0	와일드카드 문자는 적용되지 않습니다.
1	와일드카드는 검색 조건에 접미사로 적용됩니다.
2	와일드카드는 검색 조건에 접두사로 적용됩니다.
3	와일드카드는 검색 조건에 대한 접두사와 접미사로 모두 적용됩니다.

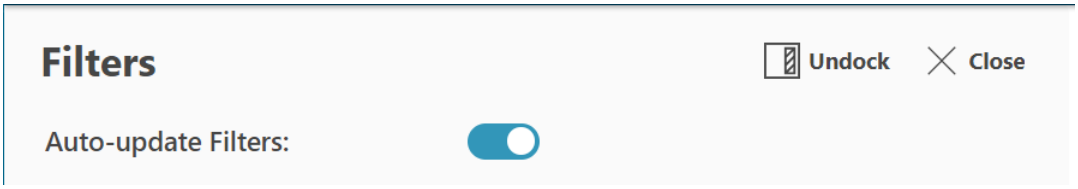
환경설정이 설정되지 않거나 0, 1, 2, 3이 아닌 값이 있는 경우 **AWC_search_automatic_wildcard 환경설정**이 적용됩니다.

벌크 필터 옵션 구성

검색 설정에서 **AWC_Search_Show_Auto_Update_Filters** 환경설정을 사용하여 **필터 자동 업데이트 옵션 표시**를 구성할 수 있습니다. 구성되면 사용자가 **검색 설정**에서 **필터 자동 업데이트 표시 옵션**을 선택하여 **필터** 패널에 **필터 자동 업데이트**를 표시할 수 있습니다.

필터 패널에서 **필터 자동 업데이트**가 선택된 경우 필터를 선택하면 검색 결과가 새로 고쳐집니다. **필터 자동 업데이트**가 선택되어 있지 않은 경우 사용자는 여러 필터를 선택하고 **필터**를 클릭하여 검색 결과를 새로 고칠 수 있습니다.

© 2024 Siemens



할당되지 않은 값 표시 구성

AWC_dynamicPriority_hideUnassignedValueFacetCategories 환경설정을 사용하여 **필터** 패널에서 할당되지 않은 값이 있는 필터 카테고리 표시를 토글합니다. 미할당 값만 포함된 필터를 숨기려면 환경설정을 **true**로 설정합니다. 미할당 필터 카테고리를 표시하려면 환경설정을 **false**로 설정합니다.

확장된 필터를 제한하는 것은 미할당 필터 숨기기보다 우선합니다. 할당되지 않은 값을 숨기고 **AWC_Limited_Filter_Categories_Enabled** 환경설정을 활성화하면 일부 필터에 여전히 할당되지 않은 값이 표시될 수 있습니다.

필터 그룹 구성

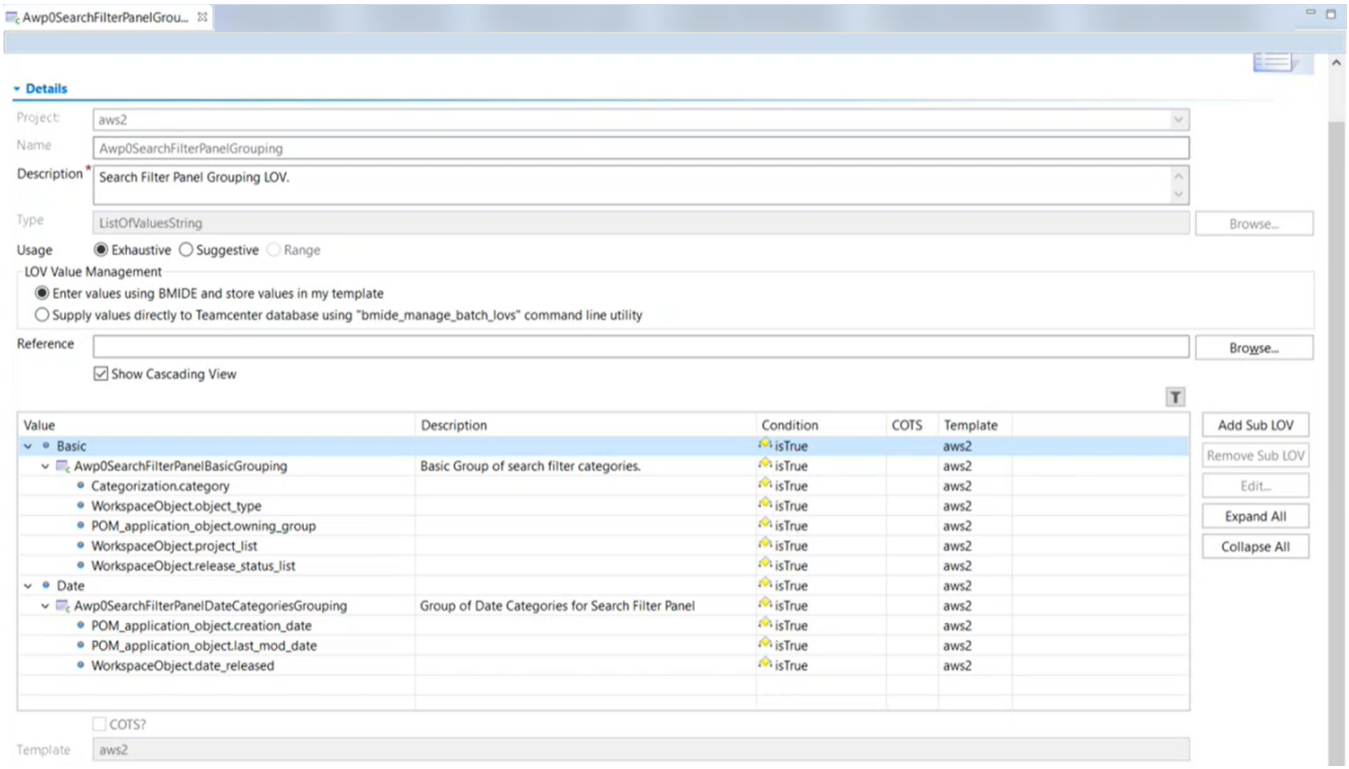
필터 그룹은 기본적으로 구성됩니다. 각 사용자는 검색 설정을 통해 필터를 리스트에 표시할지 그룹에 표시할지 선택할 수 있습니다. **AWC_Search_Filter_Categories_Display_Mode** 환경설정을 사용하여 필터 그룹을 해제할 수 있습니다.

기본 그룹은 **기본**, **날짜** 및 **기타**입니다. 기본 또는 날짜로 분류되지 않은 필터는 **기타** 필터 그룹에 포함됩니다.

자체 필터 그룹을 사용자 정의할 수 있습니다. Business Modeler IDE에서 **Awp0SearchFilterPanelGrouping** LOV(값 리스트) 템플릿을 사용하여 고유한 그룹으로 새 LOV를 생성합니다.

새 LOV 이름으로 **AWC_Search_Filter_Panel_Groups_Lov_Name** 환경설정을 구성합니다.

© 2024 Siemens



확장된 필터 그룹 세트 지정

검색 결과가 반환될 때 자동으로 확장할 필터 그룹을 구성할 수 있습니다. 확장된 그룹 수를 줄이면 긴 리스트를 더 쉽게 보거나 더 일반적으로 사용되는 필터 그룹을 더 쉽게 볼 수 있습니다.

기본적으로 **AWC_Search_Filter_Panel_Expand_Selected_Groups** 환경설정이 활성화되어 있습니다. 이 옵션을 활성화하면 자동으로 확장되는 그룹이 제한됩니다.

자동으로 확장할 하나 이상의 그룹 이름을 **AWC_Search_Filter_Panel_Groups_Expanded** 환경설정에 추가합니다.

각 사용자는 **검색 설정** 패널을 통해 확장할 추가 그룹을 선택할 수도 있습니다.

표시되는 필터 그룹의 최대 수 지정

필터 그룹의 긴 리스트를 쉽게 볼 수 있도록 한 번 표시되는 필터 그룹 수를 제한할 수 있습니다. 더 많은 필터 그룹을 사용할 수 있는 경우 리스트 맨 아래에 있는 **모두 표시**를 클릭하여 필터 그룹을 확장할 수 있습니다.

필터 그룹 표시 방법을 지정하려면 **AWC_Search_Filter_Panel_Number_Of_Groups_Shown** 환경설정을 사용합니다.

각 그룹에 표시되는 필터 수를 제한할 수 있습니다. 필터 카테고리 수가 적을수록 필터 값의 긴 리스트를 쉽게 볼 수 있습니다. 더 많은 필터를 사용할 수 있는 경우 리스트 맨 아래에 있는 **모두 표시**를 클릭하여 필터 그룹을 확장할 수 있습니다.

© 2024 Siemens

필터 그룹 표시 방법을 지정하려면 **AWC_Search_Filter_Panel_Number_Of_Categories_Shown_Inside_Each_Group** 환경설정을 사용합니다.

표시된 필터 구성

사용자에 대해 자동으로 확장되는 필터 리스트를 지정하고, 각 필터 카테고리에 대해 표시할 값의 수를 설정하고, 환경설정을 사용하여 필터 카테고리의 우선순위를 지정할 수 있습니다.

확장된 필터 세트 지정

검색 결과가 반환될 때 자동으로 확장되는 필터 리스트를 구성할 수 있습니다. 확장된 필터 수를 줄이면 긴 리스트를 더 쉽게 보거나 더 일반적으로 사용되는 확장된 필터 세트를 더 쉽게 볼 수 있습니다.

기본적으로 **AWC_Limited_Filter_Categories_Enabled** 환경설정이 활성화되어 있습니다. 이 옵션을 활성화하면 자동으로 확장되는 카테고리가 제한됩니다. **카테고리** 및 **유형** 필터 카테고리는 추가 구성이 없는 사용자에 대해 항상 확장됩니다.

각 사용자는 **검색 설정** 패널을 통해 확장할 추가 카테고리를 선택할 수도 있습니다. 사용자가 **확장할 필터** 드롭다운 목록에서 사용할 수 없는 카테고리를 확장하려는 경우 카테고리 이름을 **AWC_Master_List_Limited_Filter_Categories_Expanded** 환경설정에 추가합니다.

카테고리 및 **유형** 필터 카테고리를 사용자를 위해 자동으로 확장하려면 추가 카테고리 이름을 **AWC_Limited_Filter_Categories_Expanded** 환경설정에 추가합니다. 각 행에 값을 하나씩 사용하여 자동으로 확장할 속성을 하나 이상 지정합니다.

예제:

AWC_Limited_Filter_Categories_Expanded=POM_application_object.owning_user,
POM_application_object.last_mod_user

카테고리 및 **유형** 필터 카테고리는 **AWC_Limited_Filter_Categories_Expanded** 환경설정에 나열되지 않을 경우에도 항상 확장됩니다.

또한 사이트 환경설정 설정은 사용자의 **검색 설정** 패널을 구성합니다.

다음 예제에서는 **유형** ,**카테고리** 및 **최종 수정자** 카테고리가 사용자에 대해 자동으로 확장됩니다.

환경설정	구성
AWC_Limited_Filter_Categories_Enabled	활성화되어 있습니다(기본값). 사용자는 자동으로 선택한 **검색 설정** 패널의 **확장** 아래에 **선택한 필터** 및 **모든 필터** 옵션을 볼 수 있습니다.
AWC_Limited_Filter_Categories_Expanded	**POM_application_object.last_mod_user**를 값 리스트에 추가합니다.

© 2024 Siemens

환경설정	구성
	사용자는 **선택한 필터** 드롭다운 리스트에서 자동으로 확장할 선택된 **최종 수정자**를 볼 수 있습니다.
AWC_Master_List_Limited_Filter_Categories_Expanded	**POM_application_object.last_mod_user**가 값 리스트에 없는 경우 이를 추가합니다. 사용자는 **선택한 필터** 드롭다운 리스트에 나열된 모든 카테고리를 볼 수 있습니다. 사용자는 추가 카테고리를 확장하도록 선택할 수 있습니다.

표시되는 필터 값의 최대 수 지정

지정된 각 카테고리에 대해 표시되는 필터 값의 수를 제한할 수 있습니다. 값이 적으면 상세 필터 리스트를 쉽게 볼 수 있습니다. 더 많은 필터 값을 사용할 수 있는 경우, 리스트 맨 아래에서 **더 보기**를 클릭하여 확장할 수 있습니다.

필터 값 표시 방법을 지정하려면 **AWC_Category_Filter_Show_Count** 환경설정을 사용합니다.

ALL	**필터** 패널에 검색 결과와 관련된 모든 필터 및 값을 표시합니다.
양의 정수	지정된 수의 필터를 반환합니다. 나머지 필터는 필터 값 목록에서 **더 보기**를 클릭하여 볼 수 있습니다. 또한 이 설정은 전체 리스트가 표시되는 **더 보기**를 클릭할 때 표시되는 추가 필터의 수를 제어합니다. 반환된 카테고리의 순서는 Business Modeler IDE의 각 속성에 설정된 우선순위를 기반으로 합니다.
Hidden	**필터** 패널에서 필터 및 해당 값을 숨깁니다.

예제:

AWC_Category_Filter_Show_Count=POM_application_object.owning_user=20,
POM_application_object.owning_user=hidden

모든 필터의 기본값은 **50**입니다. 잘못 지정된 필터는 무시되며, 이런 경우 기본값이 적용됩니다.

필터 카테고리 표시 우선순위 지정

다음 환경설정을 사용하여 우선순위를 설정하면 필터 카테고리 표시 방법을 변경할 수 있습니다.

- **AWC_dynamicPriority_facetCategories**

  동적 우선순위가 필요한지 여부를 결정하는 데 사용되는 필터 카테고리를 지정합니다. 유효한 속성 값은 Business Modeler IDE의 **Awp0SearchCanFilter** 속성 상수에 정의되어 있으며 *business-*

© 2024 Siemens

object.property 형식을 사용합니다. 이 환경설정의 값은 **Mdl0ModelElement.mdl0model_object** 또는 **Classification.1000** 등과 같이 필터에 표시할 수 있는 비즈니스 개체의 속성일 수 있습니다.

기본값은 **Lbr0LibraryElement.lbr0Ancestors**입니다.

이 환경설정은 **AWC_dynamicPriority_threshold** 환경설정과 함께 작동합니다. 예를 들어, 지정된 검색 조건에 대한 검색 결과로 1000개의 개체가 반환되고 **AWC_dynamicPriority_facetCategories** 환경설정이 기본값으로 설정된 경우를 생각해 보십시오. 이 카테고리에서 반환된 개체 수가 600이고 **AWC_dynamicPriority_threshold** 환경설정이 50%로 설정되어 있으면 필터 카테고리의 동적 우선순위 지정이 활성화됩니다.

이 환경설정의 값이 **WorkspaceObject.object** 유형(가장 일반적인 유형)으로 설정된 경우 결과가 항상 100%이므로 필터 카테고리의 동적 우선순위가 항상 활성화됩니다.

- **AWC_dynamicPriority_hideSingleValueFacetCategories**

  true로 설정하면 필터 패널에서 모든 단일 값 필터 카테고리의 표시를 숨깁니다. **false**로 설정하면 모든 단일 값 필터 카테고리를 표시합니다.

  이 환경설정은 할당되지 않은 단일 값 파셋 카테고리에 적용되지 않습니다. 또한 **AWC_dynamicPriority_hideUnassignedValueFacetCategories** 환경설정에 의해 제어됩니다.

- **AWC_dynamicPriority_preferredNumberOfFacets**

  동적 우선순위에 사용할 수 있는 필터의 기본 수를 지정합니다. 기본값은 **4**입니다.

- **AWC_dynamicPriority_threshold**

  필터 카테고리 우선순위를 다시 지정하도록 트리거하는, 검색으로 반환되는 라이브러리 개체 비율을 지정합니다. 유효한 값은 1과 100입니다. 기본값은 **50**입니다.

  따라서 검색 결과의 50% 이상이 라이브러리 요소인 경우, 필터 카테고리의 우선순위를 다시 지정합니다.

주의 시작 요일 지정

속성에 날짜 상자가 표시되면 사용자가 시작 및 끝 날짜를 입력할 수 있습니다. 필터가 날짜를 표시하면 날짜가 날짜 범위에 따라 증분으로 그룹화됩니다. **주 시작 요일** 사이트 환경설정에서 주의 시작 요일을 지정합니다.

저장된 검색 구성

저장된 검색을 공유하는 경우 사용자에게 제공할 액세스 권한을 제어할 수 있습니다. 공유 검색 유형에 클래스를 사용하는 액세스 규칙을 설정합니다. 권장 권한 설정만 소유자의 사용자 그룹과 공유됩니다.

다음과 같이 조건, 값 및 액세스 규칙 이름을 설정합니다.

© 2024 Siemens

예제:

전역 검색 클래스 **Awp0FullTextSavedSearch**의 경우:

Has Class (Awp0FullTextSavedSearch)
Has Attribute(Awp0FullTextSavedSearch:awp0is_global_shared=1)
SharedSavedSearchACL

권장 사항과 같이 권한을 설정합니다.

소유 사용자: grant READ, WRITE, EXPORT
소유 그룹: grant READ, deny WRITE, EXPORT
기타: deny READ, WRITE, EXPORT

고급 검색 클래스 **SavedSearch**의 경우:

Has Class (SavedSearch)
Has Attribute(SavedSearch:shared=1)
SharedSavedSearchACL

권장 사항과 같이 권한을 설정합니다.

소유 사용자: grant READ, WRITE, EXPORT
소유 그룹: grant READ, deny WRITE, EXPORT
기타: deny READ, WRITE, EXPORT

규칙 구조의 위치가 규칙 평가 방법을 결정하기 때문에 규칙의 위치를 확인합니다.

열 정렬 구성

검색 결과에서 관련성 순서 또는 특정 테이블 열에 의해 결정된 순서에 따라 전역 검색 결과를 불지의 여부를 구성할 수 있습니다.

AWC_SearchUseUIConfigDefinedDefaultSortInplaceOfRelevance 환경설정을 추가하고 설정을 지정합니다.

- **true**

  import_uiconfig 유틸리티를 사용하여 구성한 지정된 테이블 열에 따라 검색 결과를 표시합니다.

- **false**

  검색 문자열이 개체와 일치하는 횟수, 개체 생성 날짜 또는 마지막 수정 날짜 및 기타 요소에 의해 결정된 연관성 순서에 따라 검색 결과를 표시합니다.

© 2024 Siemens

내보내기 구성

검색 결과 내보내기를 수행하는 경우 **표시된 대로**를 선택할 수 있으며, 이 옵션은 검색 결과에서 선택한 행 또는 모든 행을 Microsoft Excel로 내보냅니다. **모든 결과**를 선택하면 구성된 최대 수까지 모든 행을 내보냅니다.

AW_Search_Results_Export_Max_Rows 환경설정을 사용하여 최대 행 수를 설정합니다. 기본값은 **1000**입니다.

검색 관련 로그 찾기

Active Workspace 검색 오퍼레이션에서는 서버를 호출하여 결과를 반환합니다. 서비스 지향 아키텍처(SOA) 또는 Solr 오류의 경우는 Active Workspace에서 오류에 대한 정보가 제한됩니다. 다음 로그에서 정보를 살펴보십시오.

디버깅을 완료한 후에 로깅 수준 및 변수를 이전 설정으로 되돌리고 문제를 해결하는 시스템을 다시 시작하십시오.

tcserver 로그

- **tcserver syslog** 파일을 확인합니다. 서버측 오류에 대한 자세한 내용은 사용자 및 SOA 호출과 연관된 **syslog** 파일을 찾으십시오.

  사용자 이름을 검색하여 올바른 **syslog** 파일을 찾아서 적용 가능한 **syslog** 파일 리스트의 범위를 좁힌 후 해당 사용자와 연결된 로그에서 문자열 **performSearch**를 검색합니다.

  syslog 파일은 *Temp* 디렉터리에 있습니다. 해당 위치에 파일이 없으면 **TC_KEEP_SYSTEM_LOG**가 true로 설정되어 있는지 확인하십시오.

- 디버그 로깅의 경우 다음 파일을 엽니다.

  ```
  TC_LOGGER_CONFIGURATION\logger.properties
  ```

  다음에 대한 로깅 수준을 **DEBUG**로 변경합니다.

  ```
  logging.rootLogger
  logging.logger.Teamcenter
  logging.logger.Teamcenter.Soa.Communication
  ```

- **저널** 파일을 검사하여 **tcserver**의 낮은 수준 해석을 가져옵니다. 다음 환경 변수를 설정하여 호출된 모든 방법을 추적할 수 있습니다.

  ```
  TC_JOURNAL=FULL
  TC_JOURNALLING=ON
  TC_JOURNAL_LINE_LIMIT=0
  ```


© 2024 Siemens

Solr 로그

Solr 오류가 발생한 경우 *TC_ROOT\solr-version\server\logs\solr.log* 파일을 확인하십시오.

웹 브라우저 콘솔 로깅

검색을 수행하면 간단한 설명과 함께 오류를 확인하는 메시지가 표시될 수 있습니다. 서버 오류의 경우 웹 브라우저 콘솔에 자세한 내용과 함께 log 문이 제공됩니다. 웹 브라우저 콘솔 윈도우는 일반적으로 웹 브라우저 개발자 도구를 통해 사용할 수 있습니다.

잘못된 검색 개수 검토

*solr-version\server\solr\collection1\conf\solrconfig.xml*에서 **debugoptions**의 값을 변경합니다.

```
<str name="debugoptions">2</str>
```

콘솔에서 Solr을 다시 시작하고 검토를 위해 모든 로깅을 캡처합니다.

또한 Fiddler 또는 Firebug를 사용하여 서버에서 오는 결과를 검사할 수 있습니다.

검색 성능 향상

로깅을 생성해 검색 성능 문제를 해석할 수 있습니다. 한 가지 방법을 사용하여 성능 문제를 분리하거나 정보에 기반한 최적의 해석을 위해 여러 방법을 결합할 수 있습니다.

성능을 위해 Syslog 디버깅

*logger.properties*에 대한 로그 수준을 **DEBUG**로 설정합니다. 로깅은 검색 호출의 계층을 제공합니다. **fnd0performSearchBase**를 검색하여 검색 호출의 시작 및 끝 지점을 찾습니다.

성능 저널 로깅

.pjl 파일을 확인합니다. 환경 변수를 설정하여 성능 피드백을 가져올 수 있습니다.

```
TC_JOURNAL_PERFORMANCE_ONLY=true
```

풀 관리자 및 **tcserver**를 다시 시작합니다.

이들 파일은 매우 커질 수 있으므로 조사 중인 사용 사례만 실행하십시오. 저널 로깅은 해석하기 어려울 수 있는 대량의 데이터를 생성합니다. **TC_JOURNAL_PERFORMANCE_ONLY**를 재설정하여 정상 오퍼레이션을 복원하십시오.

© 2024 Siemens

SQL 성능 로그

환경 변수를 사용하여 느린 SQL 조회에 대한 디버그 로깅을 설정할 수 있습니다. 로깅으로 인해 **tcserver** 성능이 저하될 수 있습니다.

1. *tc_profilevars* 파일을 엽니다.
2. 1초보다 긴 SQL 문을 로깅하려면 **TC_SLOW_SQL=1**로 설정합니다.
3. **syslog** 파일에 모든 SQL 문 및 해당 타이밍을 전송하려면 **TC_SQL_DEBUG=PT**를 설정합니다.
4. 풀 관리자 및 **tcserver**를 다시 시작합니다.
5. **syslog** 파일에서 **DEBUG** 메시지를 확인합니다.

네트워크 호출에 사용할 HTTP 보관 (HAR) 파일

네트워크 트랜잭션의 웹 브라우저 성능을 로깅하기 위한 HTTP 보관 파일 (HAR)을 생성합니다. HAR 파일은 Active Workspace에서 수행한 네트워크 호출을 조사하는 데 도움을 줄 수 있습니다.

HAR 파일 생성은 웹 브라우저마다 다를 수 있지만 필수 프로세스는 동일합니다. 웹 브라우저 개발자 도구를 사용하여 **네트워크** 메뉴 또는 탭을 찾고 네트워크 트래픽을 캡처하십시오. 조사 중인 브라우저 요청만 실행하십시오. HAR 파일을 저장하는 작업을 선택해야 합니다. (예를 들어, Google Chrome에서 로깅된 항목 위로 마우스를 이동하면 컨텍스트 메뉴에서 **Save as HAR with content** (내용과 함께 HAR로 저장)을 선택할 수 있습니다.

다른 검색 유형 구성

고급 검색 구성

고급 검색은 특정 개체 속성을 검색하는 미리 정의된 조회를 사용자에게 제공할 수 있는 내장형 기능입니다. 사용자가 미리 정의된 조회에 포함되지 않은 특정 개체 속성을 정기적으로 검색하는 경우, 추가 사용자 정의 조회를 생성할 수 있습니다. 사용자는 검색 결과를 리스트 또는 테이블로 볼 수 있습니다. 테이블의 경우 사용자는 특정 속성에 대한 열 정렬 기준을 설정할 수 있습니다.

가시성 제어

고급 검색은 사용자에게 표시하거나 숨길 수 있습니다. **AW_Advanced_Search_Visibility** 환경설정을 설정하여 다음 아이템을 표시하거나 숨길 수 있습니다. 이 환경설정의 기본값은 **true**로, 사용자가 고급 검색에 액세스할 수 있도록 합니다.

- **검색** 페이지 머리글 영역의 **고급** 페이지.
- 전역 **검색** 상자의 **고급 검색**.

© 2024 Siemens

- 홈 페이지의 **고급 검색** 타일.

조회 생성

Rich Client에서 Query Builder를 사용하여 고급 검색 조회를 생성할 수 있습니다.

사용 가능한 조회 제한

AWC_Search_Saved_Queries_List 환경설정을 구성하여 고급 검색에서 사용 가능한 조회 리스트를 저장된 조회의 정의된 리스트로 제한할 수 있습니다.

QRYColumnsShownPref 환경설정을 구성하여 고급 검색에서 기본 쿼리 리스트를 설정할 수 있습니다.

QUERY_MAX_THRESHOLD 환경설정을 구성하여 고급 검색 조회의 최대 결과 수를 정의할 수 있습니다. 고급 검색 조회에서 이 환경설정에 설정된 최대값보다 더 많은 결과를 생성하는 경우 검색이 실행되지 않고 사용자에게 더 구체적인 조건을 제공하라는 메시지가 표시됩니다.

고급 검색 결과에 대한 표시 개수 설정

WSOM_find_set_search_limit 환경설정을 구성하여 고급 검색 결과의 표시 개수를 설정할 수 있습니다. 이 환경설정의 기본값은 1000입니다. 예를 들어, **WSOM_find_set_search_limit**를 1000으로 설정하면 전체 결과가 1000 이하일 때 정확한 검색 결과 수가 표시됩니다. 1000개가 넘는 결과가 반환되면 표시 개수는 **1000+**입니다.

AW_Display_Not_Readable_Count 환경설정을 구성하여 사용자가 액세스할 수 없는 결과(읽을 수 없는 내용이라고도 함)에 대한 정보를 제공하도록 표시 개수를 설정할 수 있습니다. 이 환경설정의 기본값은 **false**입니다. 이 환경설정을 구성하는 경우, 읽을 수 없는 결과가 반환되면 잠금 아이콘이 표시됩니다. 이 아이콘 위에 마우스를 두면 사용할 수 없는 결과 수가 표시됩니다.

고급 검색 조건을 조합하기 위해 구분 기호 사용

사용자가 여러 검색 조건을 결합하여 결과를 동일한 테이블에 표시하려면 구분 기호를 사용할 수 있습니다. 그러나 **AWC_WSOM_find_list_separator** 환경설정으로 특정 구분 기호를 활성화하면 해당 구분 기호가 시스템 데이터에도 있는 경우 검색 결과에 영향을 미칠 수 있습니다. 이 환경설정의 기본값은 **SEMICOLON**, **NEWLINE** 및 **TAB**입니다.

고급 검색 결과에서 열 정렬

고급 검색에서 반환된 테이블에는 조회 정의에 지정된 **검색 유형**과 일치하는 개체가 포함되어 있습니다. 기본 열에는 반환된 개체의 특정 속성 및 관련 개체의 특정 속성이 표시됩니다.

© 2024 Siemens

검색 결과 테이블의 열 정렬은 속성 유형에 따라 결정됩니다. 다음 테이블에서는 지원되는 정렬 옵션과 지원되지 않는 정렬 옵션에 대해 설명하고 있습니다.

지원되는	로컬 조회의 경우 사용자는 다음을 사용하여 정렬할 수 있습니다. • **int**, **double**, 및 **string** 등의 영구 속성 특성. 예를 들어, **object_name** 또는 **revision_number**를 사용하여 정렬할 수 있습니다. • AW_Advanced_Search_Reference_Sort_property 환경설정에서 구성한 **TypedReference** 속성. 예를 들어, **owning_user.user_id**를 사용하여 정렬할 수 있습니다. • 영구 속성 특성을 참조하는 수준이 하나만 있는 복합 속성. 예를 들어, **items_tag.item_id**에는 정렬할 수 있지만 **items_tag.owning_user.user_id**에는 정렬할 수 없습니다. 후자는 하나 이상의 수준 참조를 가지고 있기 때문입니다.
지원되지 않음	원격 조회를 사용하여 정렬하거나 사용자 정의 코드를 사용하는 조회로 정렬하는 것은 지원되지 않습니다. 사용자는 다음을 사용하여 정렬할 수 없습니다. • 배열 • **Runtime** 속성 • **UntypedReference** 속성 • 하나 이상의 수준 참조가 있는 **Compound** 속성 또는 **reference**, **runtime**, **compound**, **external**, **relation** 또는 배열 속성인 소스를 나타냅니다. • **Name-Value** 속성을 포함한 **Table** 속성.

고급 검색 결과에 대한 열 정렬 조건 정의

다음 상황에서 **AW_Advanced_Search_Reference_Sort_property** 환경설정을 **TypedReference** 속성으로 정렬하도록 구성합니다. 로컬 조회의 경우 유효한 환경설정 값은 단일(배열이 아님) **TypedReference** 속성이 되어야 합니다. 사용자 정의 코드를 사용한 원격 조회 또는 조회에 대한 정렬은 지원되지 않습니다.

예를 들어, 참조 속성이자 특성인 **owning_user.user_id**를 정렬할 수 있지만 후자는 복합 참조이므로 **owning_user.owning_site**가 아닙니다.

- 관련 개체의 열에 대한 정렬 기준을 변경하려고 합니다:

© 2024 Siemens

환경설정에서 기존 값을 구성합니다.

예를 들어 **소유자** 열을 **user_name** 속성 대신 **user_id**를 기준으로 정렬하려면 기존 **owning_user.user_name** 값을 **owning_user.user_id**로 변경합니다.

- 관련 개체의 속성을 표시하는 열을 추가하려고 합니다:

  typedReferenceProperty.attribute 폼을 사용하여 환경설정 값에 새 속성을 추가합니다.

  예를 들어, 소유 그룹 이름을 표시하는 열을 추가하는 경우 **owning_group.name**을 값 리스트에 추가합니다.

형상 검색 구성

형상 검색 기능을 Teamcenter Environment Manager (TEM) 또는 배포 센터의 Windows 또는 Linux에서 설치하는 경우 다음 기본 설정을 생성하고 구성해야 합니다.

GeolusServer

형상 검색과 Geolus 간의 통신에 사용할 URL을 정의합니다.

형상 검색 기능이 성공적으로 설치된 후 다음 환경설정을 사용하여 이를 구성할 수 있습니다. 이는 설치 프로세스 중에 **AWC_StartupPreferences** 환경설정에 자동으로 추가됩니다.

SS1_DASS_enable

형상 검색을 활성화하고 비활성화합니다.

SS1_DASS_shape_default

형상 검색에 대한 기본 형상 유사성을 지정합니다.

SS1_DASS_size_default_max

크기 필터를 적용할 때 사용자가 지정할 수 있는 기본 상한 범위 한계를 지정합니다.

SS1_DASS_size_default_min

크기 필터를 적용할 때 사용자가 지정할 수 있는 기본 하한 범위 한계를 지정합니다.

SS1_DASS_size_lower_limit

크기 필터를 적용할 때 사용자가 지정할 수 있는 최소 하한 범위 한계를 지정합니다.

SS1_DASS_size_upper_limit

크기 필터를 적용할 때 사용자가 지정할 수 있는 최대 상한 범위 한계를 지정합니다.

© 2024 Siemens

팁:

사용자에 대해 사용할 수 있는 다른 유형의 검색에 대한 자세한 내용은 Teamcenter 도움말에서 *검색 가이드맵 관리*를 참조하십시오.

GeolusServer 환경설정 설정

Active Workspace의 **형상 검색** 기능에서는 Geolus 형상 검색 엔진을 사용합니다. Active Workspace 및 Geolus 간의 통신을 활성화하려면 **GeolusServer** 환경설정을 생성해야 합니다.

Active Workspace의 **환경설정 관리** 또는 Rich Client의 **조직**을 사용하여 다음과 같은 속성을 가진 **GeolusServer** 환경설정을 생성합니다.

속성	수행할 작업
이름	**GeolusServer**를 입력합니다.
카테고리	**DecisionApps.ShapeSearch.Preferences**를 입력합니다.
Description	해당 환경설정에 대해 유용한 설명을 입력합니다.
Value	다음 형식으로 Geolus 서버 URL을 입력합니다. *protocol*://*gServer*:*gPort*/*gContext* 여기서: • *프로토콜*은 **http** 또는 **https**일 수 있습니다. • *gServer*는 Geolus 서버가 실행되는 시스템 이름 또는 시스템의 IP 주소입니다. 연결해야 하는 모든 Teamcenter 클라이언트에서 액세스할 수 있어야 합니다. • *gPort*는 서버에서 http 또는 https 요청을 처리하는 데 사용하는 포트 번호입니다. • *gContext*는 Geolus 서버의 컨텍스트 루트입니다.

여러 사이트에 대한 검색 구성

다중 사이트 공동 작업은 여러 사이트 간에 개체 공유를 활성화합니다. 이러한 개체 디렉터리 서비스(ODS) 개체들이 게시되며 여러 사이트에 걸쳐 복제됩니다. 게시된 개체들을 Active Workspace 사용자에 대해 색인화할 수 있습니다. **결과** 탭의 **필터** 패널에서는 로컬 결과 또는 원격 결과를 보여줍니다. 게시된 개체는 마스터에 대한 업데이트를 추적하고 공유된 데이터를 다른 사이트에서 업데이트해야 하는지 여부를 결정합니다.

1. **다중 사이트 색인화를 활성화**하는 설정 및 색인화를 수행해야 합니다.

2. **AWC_Search_DisplayODSContent** 환경 설정을 **true**로 설정하여 사용자의 다중 사이트 검색을 활성화합니다. **AWC_Search_DisplayODSContent**는 **필터** 패널에 **검색 결과** 카테고리를 표시할 수 있도록 해줍니다.

© 2024 Siemens

사용할 수 있는 게시 레코드가 없는 경우에는 로컬 결과만 반환됩니다. 필터링을 위해 **원격**을 선택하면 개체가 표시되지 않습니다.

© 2024 Siemens