

# Einsatz von Enterprise Architect

Von Dermot O'Bryan

*Alle Unterlagen © Sparx Systems 2010 PTY - Version 1.3 übersetzt von  
SparxSystems Software GmbH*

<http://www.sparxsystems.de>

## Inhaltsverzeichnis

<b>EINFÜHRUNG.....</b>	<b>3</b>
<b>BESONDERE MERKMALE BEIM EINSATZ .....</b>	<b>3</b>
<b>WAHL DES REPOSITORY .....</b>	<b>4</b>
EAP-DATEI-REPOSITORY .....	4
DBMS-REPOSITORY .....	5
<b>EINSATZSCHEMAS.....</b>	<b>5</b>
“SINGLE-SITE” .....	5
<i>Einzelne Projekte</i> .....	5
<i>Mehrere Projekte</i> .....	6
“MULTIPLE-SITE” SZENARIEN .....	8
1. <i>Zentrale EAP-Datei</i> .....	8
2. <i>Citrix oder Terminal Server-Emulation</i> .....	9
3. <i>Ein zentrales Repository-DBMS</i> .....	9
4. <i>Mehrere “Offsite”-Benutzer</i> .....	9
<b>ANHANG .....</b>	<b>10</b>
1: LEISTUNGSOPTIMIERUNG.....	10
<i>WAN-Optimierer</i> .....	11
2: REPLIKATION VS. XMI IMPORT/EXPORT.....	12
3: XMI – PACKAGEKONTROLLE UND VERSIONSKONTROLLE .....	13
3.1: <i>Packagekontrolle und Versionskontrolle</i> .....	13
3.1.1 <i>Packagekontrolle</i> .....	13
3.1.2 <i>Version Control</i> .....	15
4: DATENAUSTAUSCH ZWISCHEN EAP DATEIEN UND DBMS REPOSITORIES .....	16
5: BASELINE, DIFFERENCE & MERGE.....	17
6: REFERENCE DATA IMPORT EXPORT .....	17
7: JET 4.0 .....	20
8: SECURITY .....	20
9: REMOTE-INSTALLATION VON ENTERPRISE ARCHITECT .....	21

# Einführung

Sparx Systems Enterprise Architect ist für den Einsatz in großen Unternehmen konzipiert. Als skalierbare Modellierungsplattform bietet Enterprise Architect eine Reihe von Einsatzmöglichkeiten der Vielfalt moderner Unternehmen gerecht zu werden.

Dieses Whitepaper beschreibt die verfügbaren Optionen für den unternehmensweiten Austausch von Modellinformationen.

## Typische Enterprise Strukturen

Nachstehend sind typische Organisationsstrukturen aufgelistet, denen man bei der Bereitstellung von Modellierungs-Software innerhalb von Firmen begegnet. Jede hat ihre eigenen Bedürfnisse und Software-Konfigurationen..

- Einzelner Standort
  - Ein Gebäude mit viele Nutzer über LAN
  - Mehrere Gebäude mit Hochgeschwindigkeits-WAN
- Einzelner Standort – Mehrere Projekte
  - Mehrere Repositories
  - Ein Repository – Mehrere Projekte
- Mehrere Standorte
  - Mehrere Standorte mit einem langsamen WAN
  - Mehrere Standorte, kein spezifisches WAN
  - Ein großer Standort, sowie Head Office mit mehreren externen Kompetenzen oder Freelancer vor Ort.

Für eine einfache Übersicht über die Möglichkeiten zur Optimierung der Leistung für diese unterschiedlichen Strukturen, siehe Anhang: 1: Leistungsoptimierung

# Besondere Merkmale beim Einsatz

Enterprise Architect bietet eine Reihe von Features zur Vereinfachung der Implementierung in großen organisatorischen Strukturen. Wenn kombiniert, ermöglichen diese Funktionen eine gewisse Flexibilität, um eigene Layouts mit eingebauter Skalierbarkeit zu erstellen.

Diese Eigenschaften sind von der installierten Version (Edition) von Enterprise Architect sowie der eingesetzte Art der Datensicherung abhängig. Der Datenspeicher des Modells wird als "Repository" bezeichnet. Für weitere Informationen über die Features und Repository-Typen in den verschiedenen Enterprise Architect Editionen - siehe:

<http://www.sparxsystems.de/uml/ea-function/>

Folgende Funktionen unterstützen die unternehmensweite Bereitstellung von Enterprise Architect:

- Wahl des Repository:
  - EAP (Dateibasiertes) Repository
    - Ermöglicht schnelles Modell-Setup, offline (lokal) Arbeit und Replikation
  - DBMS (Serverbasiertes) Repository
    - Robust
    - Sicher
    - High volume usage
- Datentransfer
  - Ermöglicht Modelltransfer zwischen EAP-Dateien und DBMS-Repositories
- Package Control
  - Unterstützt gemeinsame Nutzung von Modellzweigen (Pakete)

- Versionskontrolle
  - Management der kontrollierten Pakete innerhalb des Entwicklungsteams
- Sicherheitsberechtigungen für Model Autoren
  - Ermöglicht granulares Sperren von Paketen und Elementen
- Ferninstallation
  - Automatisierte Installation von Enterprise Architect über Netzwerke

## Wahl des Repository

Die Enterprise Architect Corporate Edition speichert Modelle in einem dateibasierten Repository oder einem SQL-basierten DBMS-Repository. Bei beiden Lösungen gibt es Vor- und Nachteile.

Ein einfacher Vergleich der dateibasierten (EAP) und DBMS-Repository-Typen ist im Folgenden aufgelistet:

<b>Funktion</b>	<b>EAP</b>	<b>DBMS</b>
Replication	Ja	Nein
Nutzer	1..~10	Unbeschränkt
Non-Corruptible	Nein	Ja

### *EAP-Datei-Repository*

***EAP Dateien (Enterprise Architect Project) basieren auf der Microsoft Jet 3.5 Datenbank (MS Access '97 format .mdb<sup>1</sup>). Enterprise Architect unterstützt auch die Jet 4.0 Datenbank-Engine. Weitere Informationen über Jet 4.0 (z.B. als Unterstützung für Unicode) finden Sie im Anhang:***

7: Jet 4.0

Die Vorteile dieser Repository-Typ sind:

- α) Die Replikation des Repositories.
- β) Einfacher Dateizugriff über ein freigegebenes Netzlaufwerk.

Die Grenzen die bei diesem Repository-Typ zu betrachten sind:

- a) Gleichzeitiger Zugriff ist auf kleine Benutzergruppen beschränkt.
- b) In seltenen Fällen kann es zur Beschädigung von Daten kommen, falls ein Netzwerk / Stromausfall während der Bearbeitung auftritt.

---

<sup>1</sup> N.B.: MS Jet 3.5 wird nicht in Windows-64-Bit-Betriebssystemen unterstützt.

## *DBMS-Repository*

Mit Hilfe eines DBMS-Model-Repository werden die Grenzen des dateibasierten Repositories überwunden. Für eine größere Nutzerbasis bieten, typischerweise dedizierte, Server DBMS schnellere Reaktionszeiten als die Jet-EAP-Dateien. Darüber hinaus sind alle Netzwerk-Fehler, welche durch äußere Bedingungen verursacht wurden, durch die Fähigkeit des DBMS-Servers ein Rollback einer Transaktion durchzuführen bereits behandelt.

Enterprise Architect unterstützt die folgenden Typen von DBMS-Server-Repositories:

- MS SQL Server
- MySQL
- Oracle9i und 10g
- PostgreSQL
- MSDE
- Adaptive Server Anywhere
- Progress OpenEdge

Details über die Konfiguration DBMS-Server für Enterprise Architect Modell-Repositories finden Sie unter <http://sparxsystems.com/resources/corporate/>

## Einsatzschemas

Im folgenden Abschnitt werden Einsatzschemas skizziert, die auf den typischen Unternehmensstrukturen basieren, wie sie in der Einleitung skizziert sind.

**N.B.:** Informationen über die Remote (unbeaufsichtigte) Installation über ein Firmennetzwerk finden Sie im Anhang: 9: Remote-Installation von Enterprise

### *“Single-Site”*

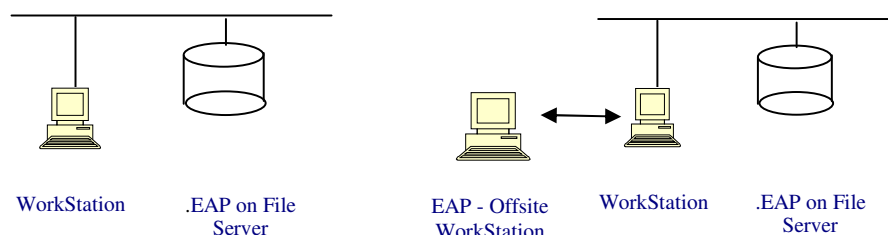
Das Szenario, das am häufigsten auftritt ist ein einziger Einsatzort mit einem einzigen gemeinsamen Repository für ein bestimmtes Projekt. In größeren Organisationen jedoch teilen sich Projekte normalerweise eine Reihe von Gemeinsamkeiten. Diese sind beispielsweise Verfahrensregeln oder allgemeine Berichte, die Corporate Standards einhalten. Unten werden diese Szenarien erläutert.

### Einzelne Projekte

#### **Mit EAP-Dateien auf einem gemeinsamen Laufwerk**

Dies ist ein einfaches Setup das für Gruppen von bis zu 10 Benutzern verwendet werden kann. Eine Repository-Datei befindet sich auf einem Fileserver, der den Nutzern über ein lokales Netzwerk (LAN) zugänglich gemacht wird. Es gibt zusätzlichen Spielraum um die Replikation von allen Repositories, die Offsite und wieder Onsite gebrachte werden, zu ermöglichen. Wie bereits erwähnt, die Einschränkungen eines dateibasierten Modell-Repositories beziehen sich ebenfalls auf diesen Ansatz.

Die folgenden Diagramme zeigen typische Konfigurationen:



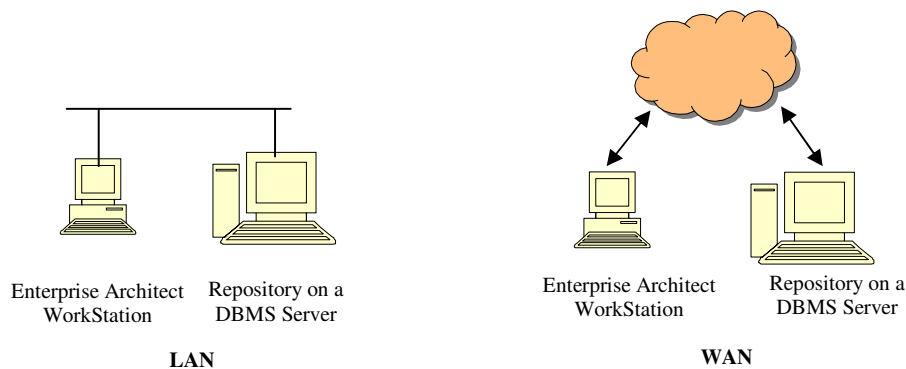
### Verwendung eines DBMS-Repositories

Das DBMS-Repository bietet ein weitaus robusteres Umfeld in Situationen, wo eine große Anzahl von Nutzern vorhanden ist.

Obwohl DBMS Repositories die mit der EAP-Struktur gelieferten Replikations-Dienstleistungen nicht anbieten, ist diese Funktionalität durch die Package Control und Versionskontroll Einrichtung in der Corporate Edition abgedeckt. Mehr:2: Replikation vs. XMI Import/Export.

Für sehr große Repositories unterstützt die DBMS-Schnittstelle eine "Lazy-Load"-Option. Wenn diese aktiviert ist, werden beim Öffnen eines Repositories nur die Daten für den sichtbaren Teil des Projektverzeichnis geladen. Dies ermöglicht schnellere Ladezeiten des Modells auf Kosten kleinerer Verzögerungen beim ersten Zugriff von Sub-Packages.

Enterprise Architect unterstützt auch eines "WAN-Optimierers" zum Anschluss an ein DBMS-Repository über ein WAN. Der Optimizer bietet Datenkompression zwischen den wichtigsten Standort-Servern und den Off-Site-Workstations. Weitere Informationen zu dieser Option finden Sie im Anhang:1: Leistungsoptimierung.



### Mehrere Projekte

Wenn mehrere Projekte gleichzeitig entwickelt werden ist es möglich, dass Kern-Code sowie Einstellungen, Berichte usw. existieren, die bei allen Projekten gleich sind. Zwei Ansätze können in solch einem Szenario hilfreich sein:

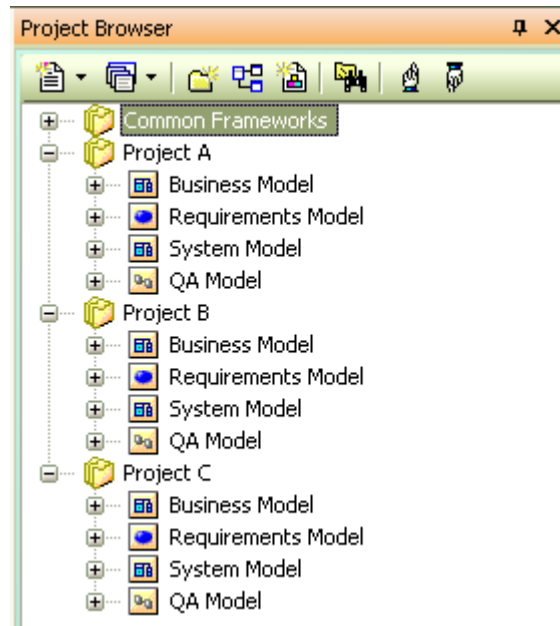
#### Ein Repository - Mehrere Projekte

Verwenden Sie ein Modell-Repository mit mehreren Projekt-Root-Knoten (siehe Abbildung unten). Jeder Root Knoten stellt ein eigenständiges Projekt dar. Gemeinsamkeiten zwischen Projekten (Frameworks, Foundation Classes etc.) können auch unter einem Wurzel-Node erfasst werden.

Mit einem einzigen Repository wird sichergestellt, dass gemeinsamer Code und Referenzen nur einmal für alle Projekte geändert werden müssen.

Darüber hinaus können in Enterprise Architects Sicherheitsfunktion verwendet werden, um den Zugriff eines Benutzers auf bestimmte Bereiche (z. B. Wurzel-Nodes oder andere Verantwortlichkeiten) zu beschränken. Die Elemente können für die Verwendung in Diagrammen abgerufen werden, etc.; Allerdings können die

Daten in diesen Elemente nur durch die Gruppe geändert werden, die auf das Paket in dem sie enthalten sind Zugriffsrechte hat.



*Project Browser-Ansicht von einem einzigen Repository mit mehreren Projekten.*

### **Mehrere Projekte - ein Repository pro Projekt**

In diesem Szenario verwendet jedes Projekt ein einziges Multi-User-Repository. Es stehen zwei Arten von Import und Export zur Verfügung: eine zum Austausch von Paketen (z.B. gemeinsame Rahmen) und die andere um zwischen den Repositories Referenzdaten (d.h. Typ-Definitionen und Berichtsvorlagen etc.) auszutauschen. Es wird empfohlen ein Master-Repository zu verwenden, um eine einzige Quelle für die gemeinsamen Projektdaten zu haben.

Für gemeinsame Pakete wie Rahmen, etc. kann XMI Import / Export, Package Control oder Versionskontrolle verwendet werden, um Arbeit zwischen den lokalen Repositories (Einzel-oder Multi-User) und dem Master-Repository auszutauschen.

Die Referenz-Daten auf lokalen Repositories (Typ-Definitionen, Makros, Sicherheit, Berichtsvorlagen, etc.) können in regelmäßigen Abständen aus dem Master-Repository über das Hauptmenü aktualisiert werden: **Tools | Export Reference Data**.

Mehr im Anhang: **6**: Reference Data Import Export

### **Großes Projekt - Branching und Merging mehrere Phasen**

In großen ("phased") Entwicklungsprojekten, kann die Darstellung der nächsten Phase ("Branch") in einer Kopie der "Base"-Repository durchgeführt werden, während das Basismodell in der Entwicklung verwendet wird. Nach einer Planungsphase können "Baselines" gegen das Branch-Modell und mit 'Baseline, Differenz und Merge' erstellt, während der Branch anschließend wieder in das Basismodell zusammengeführt werden kann.

## *Weitere Informationen finden Sie im Anhang:*

5: Baseline, Difference & Merge

### ***“Multiple-Site” Szenarien***

Wenn mehrere Standorte mit einem Hochgeschwindigkeits WAN verbunden sind, kann Enterprise Architect, wie oben skizziert, als eine Standard-Single-Site-Umgebung verwendet werden. Dieser Abschnitt konzentriert sich auf Sharing-Modelle, welche über mehrere Standorte mit minimaler Netzwerkverbindung verbunden sind. Zwei gängige Szenarien sind:

- Mehrere Standorte mit langsamem WAN
- Mehrere Standorte ohne eigenem WAN

In beiden Fällen sind die einzelnen Modelle über mehrere Standorte hinweg, gemeinsam mit einem oder vielen Off-Site-Repositories verbunden, die auf dieses zentrale Repository zugreifen und in regelmäßigen Abständen aktualisiert werden.

Übliche Konfigurationen sind:

1. Eine zentrale .eap Datei mit Replikation zu und von Offsite .eap Dateien.
2. Eine zentrale .eap Datei oder ein DBMS Repository, welches Citrix oder Terminal Server Emulation auf allen Offsite Maschinen verwendet.
3. Ein zentrales DBMS Repository, wobei die externen Seiten auch ein DBMS Repository benötigen.
4. Ein zentrales DBMS Repository, wobei die Offsite Benutzer mit einer .eap Datei arbeiten.

#### **1. Zentrale EAP-Datei**

Zwei mögliche Szenarien sind:

- a) Kleiner zentraler Benutzerpool mit Offsite-Benutzern.
- b) Kleine zentrale Website-Nutzung mit mehreren kleinen Standorten.

Im ersten Fall ist die Replikation zwischen dem Haupt-Repository und den Offsite-Benutzern eine einfache Lösung.

Die Replikation ist eine nützliche Option im zweiten Szenario, wo mehrere Standorte mit definierten Bereichen, sowie Organisationseinheiten, an einem Projekt arbeiten und eine zentrale Einheit, die weitgehend bei der Zusammenstellung der Organisationseinheiten und der Erstellung von Berichten usw. beteiligt ist. Um die .eap Dateien verwenden zu können, muss die Anzahl der Benutzer in jeder Einheit innerhalb eines angemessenen Bereichs sein.

Um Replikation zu nutzen muss die zentrale Einheit eines Master-Repositories eine Reihe von Slave-Repositories einrichten. Weitere Informationen hierzu finden Sie unter:  
2: Replikation vs. XMI Import/Export

## 2. Citrix oder Terminal Server-Emulation

In Fällen in denen die Verbindungsgeschwindigkeit zwischen Standorten akzeptabel ist (20-60 ms Latenz), aber das Volumen der Datendurchsatz beschränkt ist (d.h. 64k Bandbreite), kann eine Art Terminal-Server Applikation eingesetzt werden, um Enterprise Architect auf Remote-Arbeitsplätze zu emulieren.

## 3. Ein zentrales Repository-DBMS

Replikation mit DBMS-Repositories ist nicht verfügbar. Diese Funktion ist mit der Fähigkeit ersetzt, eine Paketkontrolle oder Versionskontrolle erstellen zu können.

Paketkontrolle stellt einen Mechanismus zur "Externalisierung" oder Weitergabe von Teilen eines Enterprise Architect Modells in Form von XMI-Dateien dar. Mit Hilfe von kontrollierten Packages ist es möglich eine weit verbreitete Entwicklung von Batch-Import / Export dieser Pakete zu unterstützen. Der zentrale Speicher kann einen Batch-Upload (Import) oder eine Batch-Verteilung (Export) an ein beliebiges externes Repository durchführen.

Dies kann weiter ausgeweitet werden um Version Control zu inkludieren. Einzelheiten zum Einrichten dieser Funktion finden Sie unter: [Package Control & Version Control](#).

**Hinweis:** Für die Offsite-Übertragung von Daten ist Versionskontrolle gegenüber Batch-Steuerung zu empfehlen, es sei denn:

- a. Die Standorte haben bestimmte Sektionen, die nur von diesem Standort geändert werden (z.B. ein Standort ist der Code-Bereich, die anderen führen Tests durch und verändern Packages nur im Bezug darauf).
- b. Die Standorte befinden sich in gegenüberliegenden Zeitzonen, in denen sich möglicherweise keine Konflikte aus parallelen Änderungen ergeben.

Weitere Informationen hierzu finden Sie im Versionskontroll-Whitepaper::  
<http://www.sparxsystems.com.au/resources/whitepapers/index.html>

## 4. Mehrere "Offsite"-Benutzer

Es ist nicht ungewöhnlich einen großen Standort als Head Office zu haben, sowie mehrere externe Freelancer, die bei Kunden direkt vor Ort sind. In diesem Szenario wird der DBMS-Server für das zentrale Repository verwendet und es wird empfohlen Offsite-Benutzer mit EAP-Repositories arbeiten zu lassen. In diesem Fall kann sowohl Versionskontrolle als auch Package Control für den Datenaustausch verwendet werden.

Package Control kann dazu verwendet werden, um einen Daten Import/Export Batch-Prozess zu ermöglichen, während Versionskontrolle einen Verriegelungsmechanismus zur Kontrolle von Daten Check-out und Check-in verwendet.

# Anhang

## 1: Leistungsoptimierung

Bei der Freigabe von großen Modellen über ein Netzwerk (LAN oder WAN) sollten einige Überlegungen zur Optimierung der Leistung gemacht werden. Im Allgemeinen ist die Leistung abhängig von:

- dem verwendeten Modell-Repository-Typ (EAP oder Remote-DBMS).
  - Ob Versionskontroll-Integration zur Bearbeitung von gemeinsamen Modellen dient
  - Die Netzwerk Reaktionszeit, basierend auf:
    - Netzwerklast
    - WAN-Latenz (eine Niedrig-Latenz WAN-Verbindung hat weniger als 40-50 ms Latenz)
- Hinweis:** am Besten in Verbindung mit dem WAN Optimizer (siehe unten).

Die nachfolgenden Tabellen zeigen die Netzwerk- und Repository-Typ-Konfigurationen um hohe Leistung zu ermöglichen. Die erste Tabelle setzt sich mit Repository-Konfigurationen auseinander, die Enterprise Architects Versionskontroll Integrationsfähigkeit nicht verwenden, während die zweite Tabelle Versionskontrolle ermöglicht.



### Konfigurationen ohne Versionskontrolle-Integration

Repository-Typ	Netzwerk-Typ	Nutzer	Optimal	Kommentar
EAP	Lokal	1~10	✓	Schnellste Reaktionszeiten.
EAP	LAN	1~10	✓	Die Leistung hängt von der LAN-Last ab.
EAP	WAN	1~10	✗	Nicht empfohlen. Verwenden Sie stattdessen ein DBMS. Ansonsten wird eine geringe Latenz vom WAN erforderlich.
EAP-Replikation	Lokal	∞ *	✓	EAP-Dateien müssen an einem zentralen Punkt zusammengeführt werden. Für die Zusammenstellung von Offsite-gesammelten Daten mittels lokaler EAP-Dateien. Mehr: Mehrere "Offsite"-
DBMS	LAN	∞ *	✓	Für größere Teams geeignet. LAN Last hängt vom DBMS Typ ab. ( MySQL & SQL Server-Modelle scheinen weniger Last als Oracle zu verursachen.)
DBMS	WAN	∞ *	✓	Optimale WAN-Verbindung: <ul style="list-style-type: none"><li>➤ WAN mit niedriger Latenz</li><li>➤ WAN-Last hängt vom DBMS-Typ ab</li></ul>
DBMS Replikation	Nutzer ⇔ DBMS (LAN) DBMS ⇔ DBMS (WAN)	∞ *	✓	Lokale DBMS Leistung sollte gleich wie für LAN/DBMS sein. Benötigt DB-Replikation. <ul style="list-style-type: none"><li>➤ Oracle sowie SQL Server unterstützen DB-Replikation.</li><li>➤ Ein DBA ist erforderlich um die DB-Replikation zu konfigurieren.</li></ul>
XMI Packages (EAP oder DBMS)	LAN ⇔ LAN	∞ *	✓	Benötigt XMI Export/Import für die Synchronisation (kann automatisiert werden). Mehr: 2: Replikation vs. XMI Import/Export

\* Concurrent-User-Grenze hängt von der Kapazität der Repository Datenbank-Server ab.

## Konfigurationen mit Versionskontrolle

Bei der Verwendung von Enterprise Architect mit Versionskontrolle über ein LAN oder WAN ist die Leistung vom eingesetzten Versionskontrolle-System sowie Netzwerk-Antwortzeiten abhängig. Jedes Versionskontroll-System hat unterschiedliche Reaktionszeiten, doch es stehen eine Reihe von Optionen sowie Themen, wie unten, zur Verfügung die betrachtet werden sollen.

Repository-Typ	Versionskontrolle	Nutzer	Optimal	Kommentar
EAP - Local	LAN	1~10	✓	Schnellste Reaktionszeiten.
EAP - LAN	LAN	1~10	✓	Die Leistung hängt von der LAN-Last ab.
EAP - Replikation	LAN or WAN	1~10	✗	Nicht empfohlen. Sollten lokale EAP-Dateien zu einem zentralen Punkt gepostet und mittels Replikation zusammengeführt werden, ist die Verwendung von Versionskontrolle schwierig und muss unter Umständen manuell durchgeführt werden.
DBMS - LAN	LAN	∞ *	✓	Schnell, die beste Option für 10 + Benutzer.
EAP - LAN	WAN	∞ *	✓	Lokale EAP-Repositories - globale Updates werden mithilfe von Versionskontrolle über ein WAN gesteuert. <ol style="list-style-type: none"> <li>1. Dies ist die schnellste WAN-Option, je nach Versionskontroll-Anbieter.</li> <li>2. Eine lokale EAP kann bei einem einzelnen Benutzer- oder einer Workgroup-Ebene verwendet werden um eine optimale Leistung zu erreichen.</li> </ol>
DBMS - LAN	WAN	∞ *	✓	Mit gemeinsamen lokalen DBMS Modellen (über LAN) mit Version Control via WAN verbunden. Dies ermöglicht den Zugriff auf verteilte Repository-Daten über Netzwerk Standorte: <ol style="list-style-type: none"> <li>1. Dies bietet die beste lokale EA Leistung. WAN-Verzögerungen sind auf Versionskontroll-Updates beschränkt.</li> <li>2. Verhindert übermäßige DB-Aufrufe über langsame Verbindungen.</li> <li>3. Data-Compression, wenn auf der WAN-Verbindung verfügbar, verbessert die Übertragung von XMI Versionskontroll-Dateien.</li> </ol>
DBMS – WAN-Replikation (LAN Verbindung auf lokalem DBMS)	LAN/WAN		✗	Produktspezifische "Replication" auf ein DBMS (SQL Server oder Oracle) ist nicht empfohlen, da sich die Nutzbarkeit neben Versionskontrolle als schwierig erweisen kann. Ein professionelles Grundverständnis von DBMS-Replikation sowie den einzusetzenden Versionskontroll-Systemen sind nötig.
WAN DBMS	LAN/WAN		✗	Nicht empfohlen. Alternativen sind:  EAP – LAN   VC - WAN  DBMS – LAN   VC- WAN

\* Concurrent-User-Grenze hängt von der Kapazität der Repository Datenbank-Server ab.

## WAN-Optimierer

Durch die Reduzierung der übertragenen Datenmenge und der Anzahl der Netzwerk-Aufrufe verbessert der WAN-Optimierer die Leistung von Enterprise Architect über eine WAN-

Verbindung mit einem DBMS-Repository. Der Optimizer ist ein einfacher Service der auf jedem Server mit einer High-Speed-Anbindung an das DBMS installiert werden kann. Mehr dazu:  
[http://www.sparxsystems.com/uml\\_tool\\_guide/uml\\_model\\_management/the\\_wan\\_optimizer.html](http://www.sparxsystems.com/uml_tool_guide/uml_model_management/the_wan_optimizer.html)

## 2: Replikation vs. XMI Import/Export

Um eine verteilte Entwicklungsumgebung zu unterstützen bietet Enterprise Architect zwei Arten des Datenaustauschs:

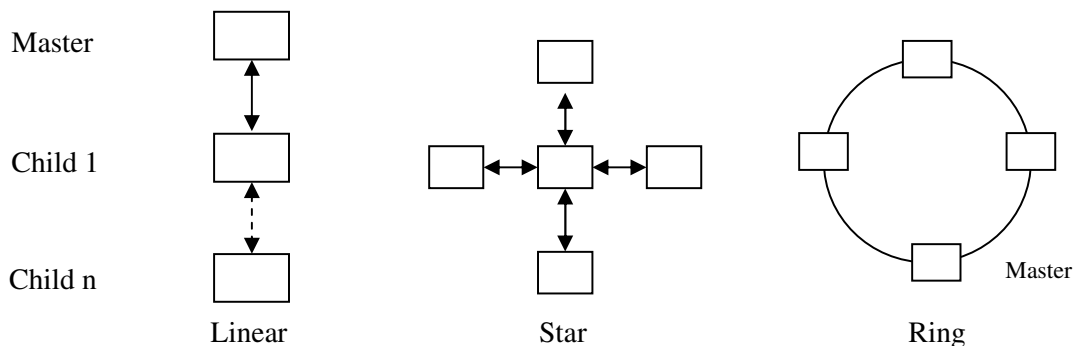
- Die Replikation von EAP-Repositories
- Der Einsatz von XMI für Package- bzw. Versionskontrolle

### Replikation

Die Replikation ist ein einfacher Weg um gültigen Austausch zwischen EAP-Repositories zu gewährleisten. In der Praxis ist ein Master-Repository in zentraler Lage aufgestellt. Repliken des Masters (Kind-Repositories) werden erstellt und Offsite für den Einsatz bei Beratern verwendet. Mithilfe dieser Repliken können dann geographisch verstreute Beratergruppen Teile des Modells aktualisieren und modifizieren. Nach der Rückkehr zum zentralen Ort werden diese Änderungen wieder in das zentrale Repository zusammengeführt.

Die Replikation kann in einer Reihe von verschiedenen Mustern aufgestellt werden (Mehr: MS Whitepapers - [Jet 3.5 White Paper](#), [Replication White Paper](#)<sup>2</sup>).

Die Replikation kann in Form einer *Linie*, *kreisförmig* oder *sternförmig* sein. Einige dieser Typen von Mustern sind in der folgenden Abbildung dargestellt.



Das typische Muster ist ein Master mit einem oder mehreren Slave-Repositories. Siehe [MSDN replication](#)<sup>3</sup> für weitere Einzelheiten über diese Muster der Replikation.

Weitere Informationen zum Einrichten der Replikation in Enterprise Architect finden Sie im Hilfethema: [Model Sharing and Team Development | Replication](#).

<sup>2</sup>

<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q164/5/53.asp&NoWebContent=1>

<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q181/3/71.ASP&NoWebContent=1>

<sup>3</sup>

[http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnaraccess/html/msdn\\_replicat.asp](http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnaraccess/html/msdn_replicat.asp)

### ***3: XMI – Packagekontrolle und Versionskontrolle***

Bei Verwendung eines DBMS (oder einer große EAP-Datei) als Repository, kann der XMI basierte Import/Export verwendet werden um einzelne oder gruppierte Packages mit Hilfe von XML zu exportieren. Diese können dann unter den Entwicklungs-Team gemeinsam genutzt werden. Dieser Ansatz hat mehrere Vorteile gegenüber der Replikation:

1. Sie können ein Modell nur aus den Teile bauen die Sie brauchen um Ihren Job zu erledigen.
2. Sie können ein vollständiges Modell bauen, falls erforderlich.
3. Sie können verschiedenen Package-Versionen eines Modells für unterschiedliche Zwecke bauen (z. B. Teile die für nur für Kunden sichtbar sind oder intern freigegeben sind etc.).
4. Nach Bedarf können Teile eines Modells zurückgeführt werden.
5. Es gibt weniger "Kollisionen" zwischen Entwicklern, wenn jeder Benutzer auf einem diskreten Package arbeitet.

Der Prozess ist noch enger steuerbar über Versionskontroll-Systeme. Weitere Informationen über das Einrichten von Bulk Import/Export finden Sie unter:3.1: Packagekontrolle und Versionskontrolle.

#### **3.1: Packagekontrolle und Versionskontrolle**

Enterprise Architect bietet zwei Methoden der Versionskontrolle:

- Packagekontrolle
- Versionskontrolle

Obwohl beide Prozesse ähnlich sind, indem sie für die Koordinierung der Verteilung der Pakete zwischen den Nutzern verwendet werden können, kann die Versionskontroll-Funktion zusätzlich ein Änderungsprotokoll von Enterprise Architect Modell-Paketen speichern und dazu frühere Versionen abrufen. Version Control erfordert jedoch die Installation und Anschluss von Software von Drittanbietern an Enterprise Architect anschließen.



##### **3.1.1 Packagekontrolle**

Packagekontrolle kann für das importieren oder exportieren eines Batchs von Packages von einem Repository in ein anderes eingesetzt werden (z.B. eine Niederlassung zu einem zentralen Repository). Um dies einzurichten, ist es notwendig die Packages zu definieren, die im Batch enthalten sein sollen. Diese nennen sich 'Controlled Packages'.

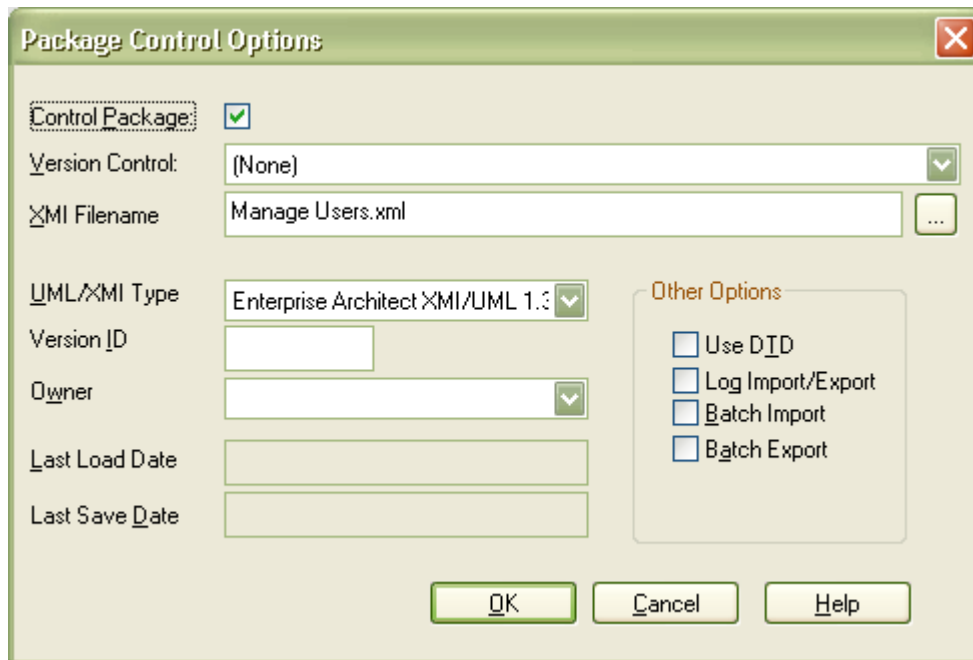
Die dafür geeignete Vorgehensweise ist:

- a) Kontrollierte Packages konfigurieren
- b) Batch Import/Export anwenden

Um die Control Packages konfigurieren zu können:

  Ein Package auswählen und zum Batch Export anhängen.

  Vom Hauptmenu auswählen: **Project | Version Control | Configure Current Package.**  
Danach kommt der folgende Dialog:



☐ Die **Control Package** Option abhaken.

☐ **OK** drücken.

☐ Wiederholen Sie diesen Vorgang für die Pakete, die als Batch exportiert werden sollen.

Danach erhalten Sie Zugriff, um Batch-Import/Export wie folgt zu verwenden:

1. Vom **Project | Import/Export** Untermenü, **Batch XMI Export** auswählen.



2. Im *Batch-Export*-Dialog: überprüfen Sie die Packages, die in diesem Export-Run enthalten sein sollen.
3. Drucken Sie auf *Save Settings*, um diese Konfiguration als Standard zu speichern.
4. *Run Export* drücken.

### 3.1.2 Version Control

Enterprise Architect unterstützt Versionskontrolle von Packages und deren Subpackages in einem zentralen Versionskontroll-Repository, das von einem versionskontrollierten Drittanbieter-Anwendung gepflegt wird. Es bietet zwei entscheidende Vorteile:

- Die Koordination der Package-Verteilung zwischen den Nutzern.
- Speichern einer Historie der Änderungen von Enterprise Architect-Packages, einschließlich die Möglichkeit, frühere Versionen abzurufen.

Es kann mit jeder beliebigen Versionskontroll-Software eingerichtet werden, die mit den folgenden Normen kompatibel ist:

- SCC standards
- CVS
- Subversion
- Microsoft TFS

Das Teilen eines Modells mit Versionskontrolle bringt einige Vorteile, abhängig von der Umwelt und der Art der benötigte Kontrolle - wie etwa Unterstützung für verschiedene Versionen und Versionierungen.

Grundsätzlich gibt es vier Möglichkeiten, um Versionskontrolle einzusetzen:

Zweck	Beschreibung
Shared-Modelle	Benutzer teilen sich eine zentrale EAP-Datei oder SQL-Datenbank. Diese Konfiguration ermöglicht es Benutzern von anderen Nutzern Packages zu sehen, ohne sie ausdrücklich abrufen zu müssen. Version regelt den Zugriff auf Packages und protokolliert Package-Revision.
Duplikat-Modelle	Eine EAP-Datei wird von einem einzigen Benutzer erstellt, der sie für Versionskontrolle konfiguriert. Die Datei wird dann an andere Nutzer, die mit ihren eigenen Benutzernamen einloggen, verteilt. Packages von anderen Benutzer werden mit dem Get-Package Befehl abgerufen.
Shared-Packages	Einzelne Benutzer erstellen ihre eigenen EAP-Dateien, teilen aber auch ein oder mehrere Packages durch den Anschluss an das gleiche SCC-Projekt und mit dem Get-Package Befehl.
Standard-Packages	Ein Unternehmen kann einen Standard-Satz von Packages haben, die weitläufig verteilt werden (schreibgeschützt). Einzelne Benutzer verwenden den Get-Package-Befehl um eine Verbindung zum Haupt-Package zu schaffen.

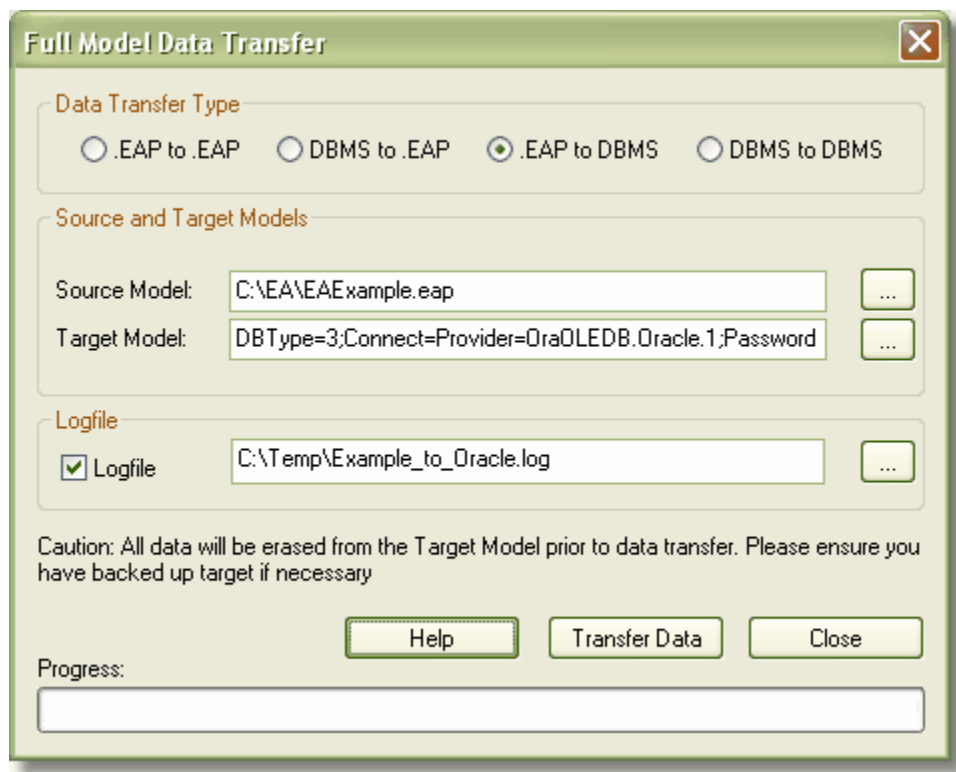
Weitere Informationen zum Einrichten der Versionskontroll-Systeme in Enterprise Architect finden Sie in der Hilfe-Datei unter: [Model Management | Version Control](#). Das folgende Whitepaper enthält weitere Einzelheiten über die Verwendung von Version Control: <http://www.sparxsystems.com.au/resources/whitepapers/index.html>

#### 4: Datenaustausch zwischen EAP Dateien und DBMS Repositories

Notwendig für die Einrichtung Offsite-Repositories ist die Fähigkeit eine vollständige Datenübertragung zwischen Repositories zu realisieren. Dies wird in Enterprise Architect durch die Datentransfer-Funktion, die für den Datenaustausch sorgt, unterstützt:

- EAP ↔ EAP
- DBMS ↔ EAP
- DBMS ↔ DBMS

Die Datenübertragungs-Option kann über das Hauptmenü aufgerufen werden unter: **Tools | Data Management | Data Transfer**. Dies öffnet den *Full Model Data Transfer* -Dialog.



Dieser Funktion wird häufig während einer Ersteinrichtung eines DBMS-Repositories verwendet.

## ***5: Baseline, Difference & Merge***

Eine Baseline ermöglicht es Ihnen eine "Momentaufnahme" eines Teils des Modells zu erstellen. Sie können Baselines verwenden um zwei Schnappschüsse von einem bestimmten Teil des Projekts zu vergleichen, die Unterschiede zwischen ihnen zu erfassen und diese entweder rückgängig machen oder ausgewählte Änderungen bzw. alle Änderungen übernehmen.

Ein Standard-Verfahren, bei dem diese verwendet werden kann, ist bei der Erstellung periodischer Baselines und das Vergleichen der aktuellen Modell-Daten gegen eine Grundlinie, wonach Updates erlaubt oder zurückgesetzt werden können.

Diese Funktionalität unterstützt auch "Branching und Merging" zwischen den Modellen. Dies umfasst das Kopieren eines "Base"-Modells als "Branch". Dann, nach einer Designphase, ist die Erstellung einer Baseline gegen das Branch-Modell möglich und mit "Baseline, Difference und Merge" wird die Branch wieder mit dem Base Modell zusammengeführt.

Kopie eines Basis-Modells – siehe Appendix 4: [Interchanging Data between Repositories](#)

Erfahren Sie mehr über Baseline, Difference and Merging unter:

[http://www.sparxsystems.com/uml\\_tool\\_guide/uml\\_model\\_management/baselinesanddifferences.html](http://www.sparxsystems.com/uml_tool_guide/uml_model_management/baselinesanddifferences.html)

## ***6: Reference Data Import Export***

Es gibt zwei Möglichkeiten mit Projekten umzugehen die gemeinsame Daten verwenden:

- a) Behalten Sie ein einzelnes Repository mit Projekt Roots für jede Anwendung, sowie Pakete für alle gemeinsamen Funktionen (d.h. Foundation Classes).
- b) Repositories für jedes Projekt getrennt halten, jedoch gemeinsame Referenz-Daten regelmäßig aktualisieren.

Die Referenz Import-/Export-Funktion unterstützt die zweite Option. Typischerweise wird ein Master-Repository immer auf dem neuesten Stand gehalten und eine Auswahl der Referenzdaten regelmäßig auf andere Repositories übertragen.

Gemeinsam genutzte Teile eines Repositories sind, unter anderem:

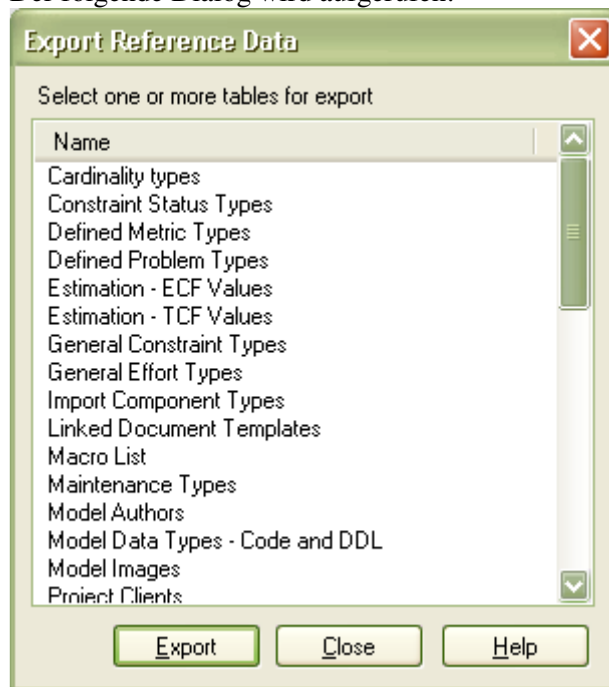
- Glossarien
- Typbeschreibungen (Status-Typen etc.)
- Ressourcen, Clients, etc.
- RTF- und HTML-Vorlagen
- Sicherheitsfunktionen

- Zusätzliche, stereotype Profile

Wenn Referenzdaten exportiert werden, schreibt Enterprise Architect sie auf eine eigene XML-Datei. Diese enthält Tabelleninformationen, Filterinformationen, Zeilen sowie Spalten.

Um Daten zu exportieren, befolgen Sie bitte folgende Schritte:

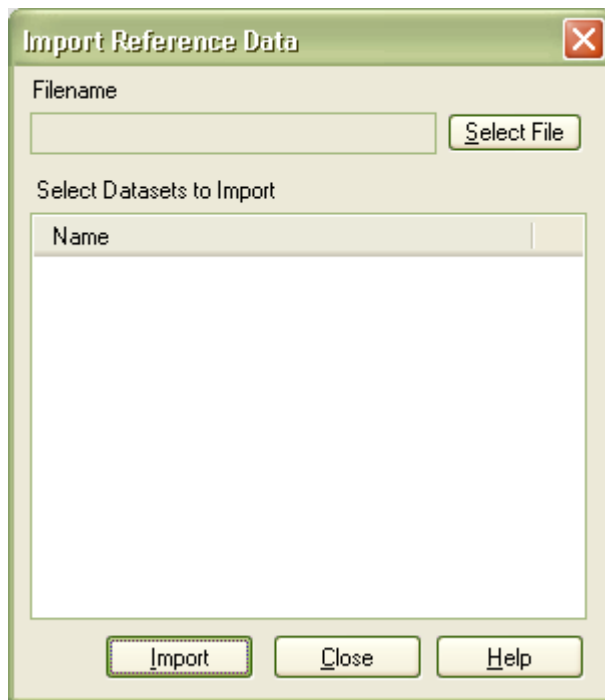
- 1) Im *Tools*-Menü, *Export Reference Data* auswählen.  
Der folgende Dialog wird aufgerufen:



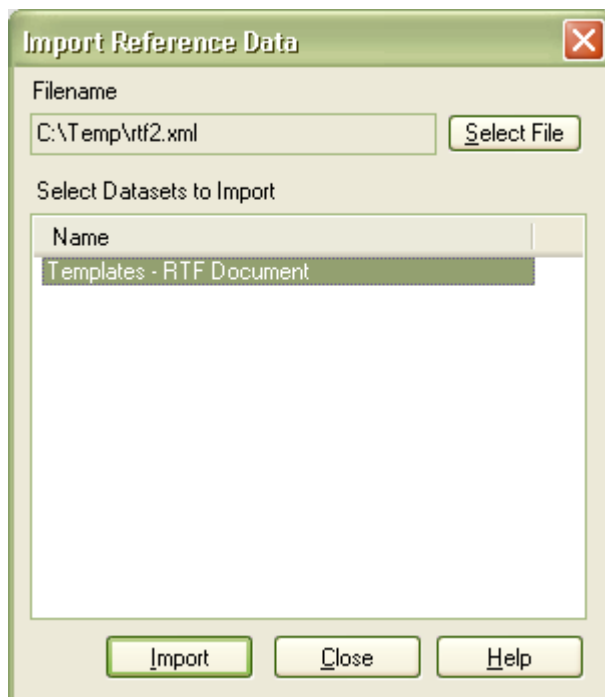
- 2) Wählen Sie die Tabelle(n) die Sie exportieren wollen – Sie können eine oder mehrere Datengruppen für eine einzige Datei auswählen.
- 3) Drücken Sie auf *Export*.
- 4) Sobald aufgefordert, geben Sie einen gültigen Dateinamen mit .XML Dateinamenerweiterung an.
- 5) Die Daten werden in der Datei exportiert.

Um Daten in ein Repository zu importieren:

- 1) Wählen Sie aus der Hauptmenü: *Tools | Import Reference Data*. Folgender Dialog erscheint:



- 2) **Select File** drücken um einen Datei-Explorer zu öffnen. Wählen Sie eine exportierte Referenz-XML-Datei aus.



- 3) **Import** drücken um die Daten zu importieren.

## 7: Jet 4.0

Falls eine Jet 4.0-Datenbank benötigt wird (z.B. um Unicode zu unterstützen), muss die EAP-Datei mittels MS Access in Jet 4.0-Format konvertiert werden und die Jet 4.0-Option in Enterprise Architect aktiviert sein. Die genauen Schritte dazu sind Folgende:

### Dateikonvertierung:

- a) MS Access 2000+ verwenden um die EAP-Datei von Jet 3.5 zu konvertieren:
  - a. Öffnen Sie die EAP-Datei in MS Access
  - b. Eine Nachricht, "You can't make changes to the database objects in the database" wird angezeigt. – Mit **OK** bestätigen
  - c. Wählen Sie aus dem MS Access-Hauptmenü:  
**Tools | Database Utilities | Convert Database | To Access 2000 Format**
  - d. Danach erscheint die Eingabeaufforderung:
    - i. Wählen Sie den Speicherort und den Dateinamen mit "eap" Erweiterung aus: **repository.eap**
    - ii. Wählen Sie: **Save type as:** auf: "all files (\*.\*)"
    - iii. **OK** drücken

Die neue Datei sollte jetzt im Jet 4.0-Format sein.

### Enterprise Architect Einstellung:

Die Jet 4.0 Option muss in Enterprise Architect aktiviert sein. Wählen Sie hierzu vom Hauptmenü: **Tools | Options | General | [ ] Use Jet 4.0**. Zum aktivieren ist ein Neustart erforderlich.

Das Jet 4.0-Repository kann jetzt in Enterprise Architect geöffnet werden.

**Hinweis:** Es gibt eine Jet 4.0 Basismodell-Datei: EABase.eap. Diese ist unter <http://www.sparxsystems.com.au/resources/> verfügbar oder direkt bei: [http://www.sparxsystems.com.au/bin/EABase\\_JET4.zip](http://www.sparxsystems.com.au/bin/EABase_JET4.zip)

8:

Die Corporate-Version von Enterprise Architect verfügt über frei definierbare Sicherheitseinstellungen – diese ermöglichen es den Nutzerzugriff auf die Modell-Update Funktionen einzuschränken. Elemente können benutzer- oder gruppenspezifisch gesperrt werden - ein Passwort wird zum Login benötigt.

Sicherheit in Enterprise Architect ist nicht konzipiert um unauthorisierten Zugriff zu verhindern; vielmehr dient sie als ein Hilfsmittel um Design- und Entwicklungskooperationen zu verbessern indem es paralleles Editieren verhindert und ungewollte „Mehrauthorenschaft“ bei Modelländerungen durch nichtauthorisierte Autoren zu vermeidet.

Enterprise Architect stellt zwei Sicherheitsrichtlinien zur Verfügung:



#### Standard Sicherheitsmodell

Alle Elemente und Diagramme sind allgemein zugänglich wobei Nutzer, abhängig von ihren zugeteilten Rechten, je nach Bedarf jedes Element oder jede Elementgruppe auf Nutzer- oder Gruppenebene sperren können.



#### Rigoroses Sicherheitsmodell

Diese Vorgehensweise setzt voraus dass alles gesperrt ist, bis es durch Anwendung eines User Locks ausgecheckt wird. Somit ist ein Enterprise Architect Modell schreibgeschützt bis ein Nutzer sein Editor-Lock auf ein oder mehrere Elemente anwendet.

Sicherheit ermöglicht es Elemente vor Änderungen zu schützen, jedoch haben Benutzer die Möglichkeit diese als verlinkte Elemente in Diagrammen zu setzen, falls Sie die nötigen Schreibrechte haben. Diese Elemente werden in den Diagrammen angezeigt und sind editierbar. Beispielsweise ist die Definition eines Servers durch den Architekten gesperrt, bleibt aber darstellbar in einem Diagramm welches vom Deployment-Manager erzeugt wurde.

Mehr Informationen zu Sicherheits Features von Enterprise Architect findet man in der Enterprise Architect Hilfe:

[Model management](#) | [Team Development](#) | [User Security](#)

## ***9: Remote-Installation von Enterprise Architect***

Wenn Enterprise Architect über ein Netzwerk mit Workstations eingesetzt wird, gibt es einige Anwendungen und Methoden die implementiert werden können. Der Einsatz kann durch Windows Server Tools oder Implementierungssoftware wie Microsoft SMS durchgeführt werden. Im Folgenden wird die Methode zur Erstellung einer MSI-Datei für diese Art der Remote-Installation kurz erläutert:

### **Überblick**


Das Setup von Enterprise Architect beinhaltet einen eingebetteten Windows MSI Installer. Beim Ausführen der .exe Datei wird die .msi Datei entpackt und in einem temporären Verzeichnis gespeichert. Danach kann sie für die Remote-Installation von Enterprise Architect benutzt werden.


Manche Softwarelösungen extrahieren standardmäßig die MSI Datei. Nachstehend finden Sie die manuelle Methode:

### **Methode**


Die MSI für Ihre Remote-Installation wird wie folgt extrahiert:


 ① Doppelklick auf Enterprise Architects setup.exe (d.h. [EA80\\_Reg.exe](#)).

 ① Das Installations-Setup durchführen bis der "Willkommen"-Dialog erscheint.

 ① Gehen Sie mit dem Dateixplorer zu [%Temp%](#) oder [C:\Documents and Settings\Administrator\Local Settings\Temp](#).

Hinweis: es kann sein, dass {administrator} als aktuelle Nutzer gesetzt ist – falls Sie einen weiteren Administrator-Benutzername haben, dann benutzen Sie das Verzeichnis welcher nach diesem benannt ist.

 ① Finden Sie die MSI-Datei – dies ist eine zufällig benannte Datei mit der Erweiterung „.msi“. (Am besten sortiert man nach Datum um die aktuellsten Einträge zu sehen).

 ① Diese MSI-Datei zu einem anderen Ort kopieren.

 ① Diese Datei können Sie für die Installation durch Windows Server oder SMS, etc., verwenden.

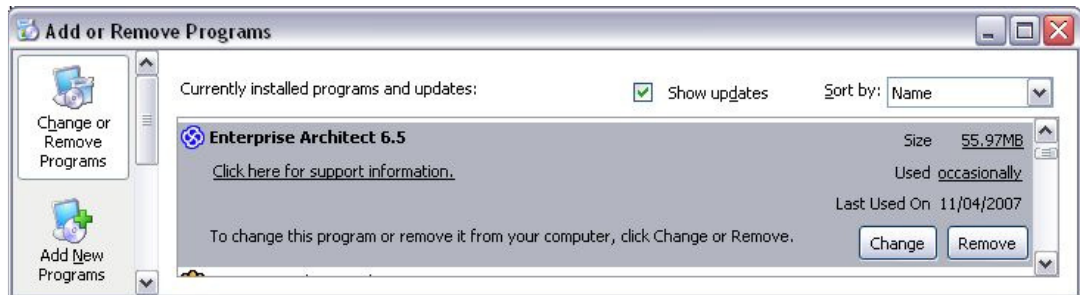
Hinweis: Standardmäßig installiert Enterprise Architect für den "Derzeitigen Benutzer". Um für "Alle Benutzer" zu installieren bitte angeben:

```
msiexec /i c:\setupfull.msi /q allusers=2
```

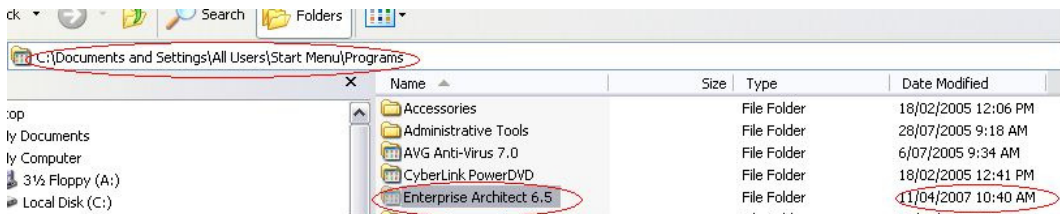
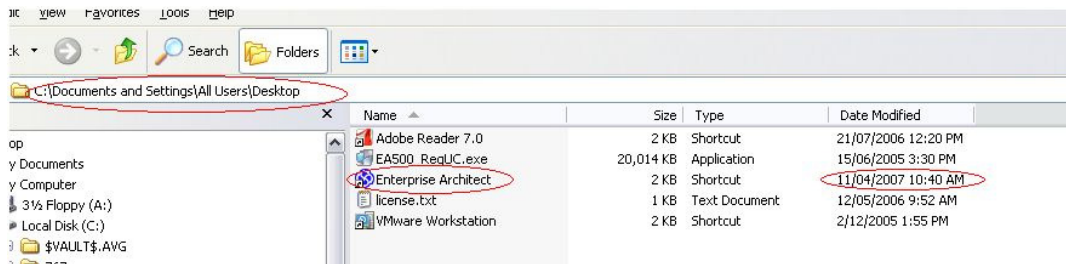
## Installation Nachprüfen:

Bitte beachten Sie, dass im Folgenden Beispiel Enterprise Architect für "All Users" installiert wurde.

- 1) Überprüfen Sie nach der Installation in **Windows | Add/Remove Programs** ob Enterprise Architect erfolgreich installiert wurde.



2. Überprüfen Sie das "All Users"-Profil für Desktop und die Start Menüpunkte um sicherzustellen, dass Enterprise Architect erfolgreich für alle Benutzer installiert wurde.



## “Remote Floating License” Installation für Enterprise Architect 8 und spätere Versionen

Während der automatischen Installation von Enterprise Architect 8 (und spätere Versionen), können Registry-Einträge benutzerspezifisch gesetzt werden, welche Zugriff auf einen „Floating License Key“ beim Start von Enterprise Architect bieten. Die Registry-Einstellungen in den dateibasierten und servicebasierten Keystoren weichen voneinander ab, wie unten beschrieben:

1. Beispiel-Registry-Einstellungen für den **File Based** Keystore:

```
[HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]
"SKT"=hex:00
"SharedKeyFolder"=" Y:\Dev\Licenses"
"AutoCheckoutEx"=hex:1a,00,00,00
```

2. Beispiel-Registry-Einstellungen für den **Service Based** Keystore:

```
[HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]
"SKT"=hex:01
"SSKSAddress"="ssks://pathToKeystoreService"
"SSKSPassword"="service password (encrypted)"
"AutoCheckoutEx"=hex:1a,00,00,00
```

## Wichtige Begriffserklärungen

**SKT**  
(SharedKeystoreType) Definiert den Keystore-Typ von dem Shared Keys erhalten werden. Erlaubte Werte sind 0x00 (File-Based Keystore) oder 0x01 (Service-Based Keystore).

**SharedKeyFolder** Dieser Wert soll zum abgebildeten Verzeichnis- oder Netzwerkpfad zeigen wo die **sskeys.dat**-Datei enthalten ist. Diese Einstellung wird nur verwendet falls der SKT-Schlüssel einen Wert von 0x00 (File-Based Keystore).

Das obige Beispiel zeigt auf ein Verzeichnis in einem Netzwerklaufwerk: **Y:\dev\licenses**.

### Hinweis:

- Achten Sie auf den doppelten Backslash im Registry-Eintrag.
- Ein vollständiger UNC-Pfad wird empfohlen. Zum Beispiel: **"\\DevelopmentServer\EA Licenses"**.
- Enterprise Architect Benutzer müssen sowohl Lese- als auch Schreibrechte besitzen, um auf dieser Datei zugreifen zu können.
- Wenn Sie auf eine Schlüsseldatei auf einem **Novell-Server** zugreifen wollen, achten Sie bitte bei der Pfadangabe auf Groß/Kleinschreibung.

**SSKSAddress** Die ssks-Adresse zum Shared Keystore Service Endpunkt. Diese Einstellung wird nur verwendet, fall der SKT-Key einen Wert von 0x01 (service based keystore) hat.

**SSKSPassword** Sollte ein Kennwort für den Shared Keystore Service erforderlich, kann dieses in den Wert eingegeben werden. Beachten Sie, dass dieser Wert verschlüsselt ist und nicht als reines Text eingegeben werden kann. Diese Einstellung wird nur verwendet, wenn der SKT-Key einen Wert von 0x01 (Service-Based Keystore) hat.

**AutoCheckoutEx** Zeigt welche Produktschlüssel Enterprise Architect beim Start automatisch beziehen soll. Jeder Key wird aus 4 Bytes konstruiert:  
  
1a 00 00 00  
  
wobei Bytes 1-2 der Lizenzcode sind (1a00) und Bytes 3-4 sind die Lizenztyp Flag (0000). Unten finden Sie eine Auflistung von zulässigen Werte:

### Lizenzcodes für AutoCheckoutEx:

License	Code
Enterprise Architect Corporate	0200
Enterprise Architect Ultimate	1a00
Enterprise Architect Business & Software Engineering	1800

Enterprise Architect Systems Engineering	1900
MDG Integration for Visual Studio	0a00
MDG Integration for Eclipse	1400
MDG Integration for TcSE	2300
MDG Link for Visual Studio	0300
MDG Link for Eclipse	0800
MDG Link for Doors	0e00
MDG Technology for SysML	1000
MDG Technology for DDS	1200
MDG Technology for Zachman Framework	1600
MDG Technology for TOGAF	1d00
MDG Technology for DoDAF-MODAF	1b00
RaQuest	0c00

### Lizenztypen für AutoCheckoutEx:

- Full License: 0000
- Academic License: 0100

**Beispiel:** Angenommen alle Benutzer bekommen sowohl eine Corporate-Lizenz als auch eine Visual-Studio-Integrationslizenz, dann wäre der Registry Schlüsselwert:

"AutoCheckoutSharedKeyArray"=hex:02,00,00,00,0a,00,00,00

### Remote-Floating-Lizenz Installation für Enterprise Architect 7.5 und frühere Versionen

Folgende Registry-Einträge können benutzerspezifisch gesetzt worden indem der Zugriff auf einen „Floating License Key“ beim Start von Enterprise Architect erteilt wird:

1. SharedKeyFolder
2. AutoCheckoutSharedKeyArray

Unten ist ein Registry Export von Key-Werte, die gesetzt werden müssen:

```
[HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]
"SharedKeyFolder"="Y:\Dev\Licenses"
"AutoCheckoutSharedKeyArray"=hex:02
```

### SharedKeyFolder

Der Wert des **SharedKeyFolder** sollte auf den abgebildeten Verzeichnispfad, oder Netzwerkpfad, zeigen wo die **sskeys.dat**-Datei enthalten ist.

Das obige Beispiel zeigt auf ein Verzeichnis in einem Netzwerklaufwerk:

Y:\dev\licenses.

#### Hinweis:

- Achten Sie auf den doppelten Backslash im Registry-Eintrag.
- Ein vollständiger UNC-Pfad wird empfohlen. Zum Beispiel: "\\DevelopmentServer\EA Licenses".
- Enterprise Architect Benutzer müssen sowohl Lese- als auch Schreibrechte besitzen, um auf dieses Datei zugreifen zu können.
- Wenn Sie auf eine Key-Datei auf einem **Novell-Server** zugreifen wollen, bitte achten Sie bei der Pfadangabe auf Groß/Kleinschreibung.

## AutoCheckoutSharedKeyArray

Der Wert von `AutoCheckoutSharedKeyArray` zeigt welche Lizenztypen Enterprise Architect beim Start automatisch beziehen soll. Dies sind die zulässigen Werte:

Lizenz	Code
Enterprise Architect Corporate	02
Enterprise Architect Ultimate	1a
Enterprise Architect Business & Software Engineering	18
Enterprise Architect Systems Engineering	19
MDG Integration for Visual Studio	0a
MDG Integration for Eclipse	14
MDG Integration for TcSE	23
MDG Link for Visual Studio	03
MDG Link for Eclipse	08
MDG Link for Doors	0e
MDG Technology for SysML	10
MDG Technology for DDS	12
MDG Technology for Zachman Framework	16
MDG Technology for TOGAF	1d
MDG Technology for DoDAF-MODAF	1b
RaQuest	0c

**Beispiel:** Angenommen alle Benutzer bekommen sowohl eine Corporate-Lizenz als auch eine Visual-Studio-Integrationslizenz, dann wäre der Registrierungsschlüsselwert:

"AutoCheckoutSharedKeyArray"=`hex:02,0a`