



x-plant Engine 2.0

Handbuch



Stand: 15.03.2010

TBFA GmbH
Birnbacher Straße 15
D-84364 Bad Birnbach

Copyright © 2010 TBFA GmbH

Alle Rechte vorbehalten. Die in diesem Handbuch enthaltenen Angaben sind ohne Gewähr und können ohne weitere Mitteilung geändert werden. Die TBFA GmbH geht hiermit keinerlei Verpflichtungen ein. Die in diesem Handbuch beschriebene Software wird auf Basis eines Lizenzvertrages geliefert. Der Lizenzvertrag befindet sich bei der Verpackung der CD, im Internet unter www.x-plant.com und wird auch durch das Installationsprogramm angezeigt. Dieses Handbuch oder Ausschnitte aus diesem Handbuch dürfen ohne schriftliche Genehmigung der TBFA GmbH nicht kopiert oder in irgendeiner anderen (z.B. digitaler) Form vervielfältigt werden.

Inhaltsverzeichnis

X-PLANT ENGINE	1
ÜBERSICHT	1
ENGINES	2
REPOSITORIES	3
ADMINISTRATION UND BETRIEB	4
Neues Repository erzeugen	4
Repository hinzufügen	6
Repository entfernen	8
Repository anpassen	8
KONFIGURATION UND ANPASSUNGEN	9
Datenmodell	9
Übersicht	9
Basisdaten	10
Objektklassen und Attribute	11
Abfragen, Ansichten und Reports	12
Sonstige Konfigurationsdateien	14
XPLANTCONFIG2.XML	14
SCALES.XML	16

x-plant Engine

Übersicht

Das gesamte Datenmanagement der x-plant Produktfamilie wird über eine zentrale Komponente abgewickelt: die x-plant Engine. Es handelt sich hierbei um einen eigenen Windows-Prozess bzw. alternativ Windows-Dienst, der für alle übrigen x-plant Module (Content, Catalog, usw.) eine einheitliche Schnittstelle für das Laden und Speichern von Daten (Katalogdaten, Projektdaten, usw.) zur Verfügung stellt.

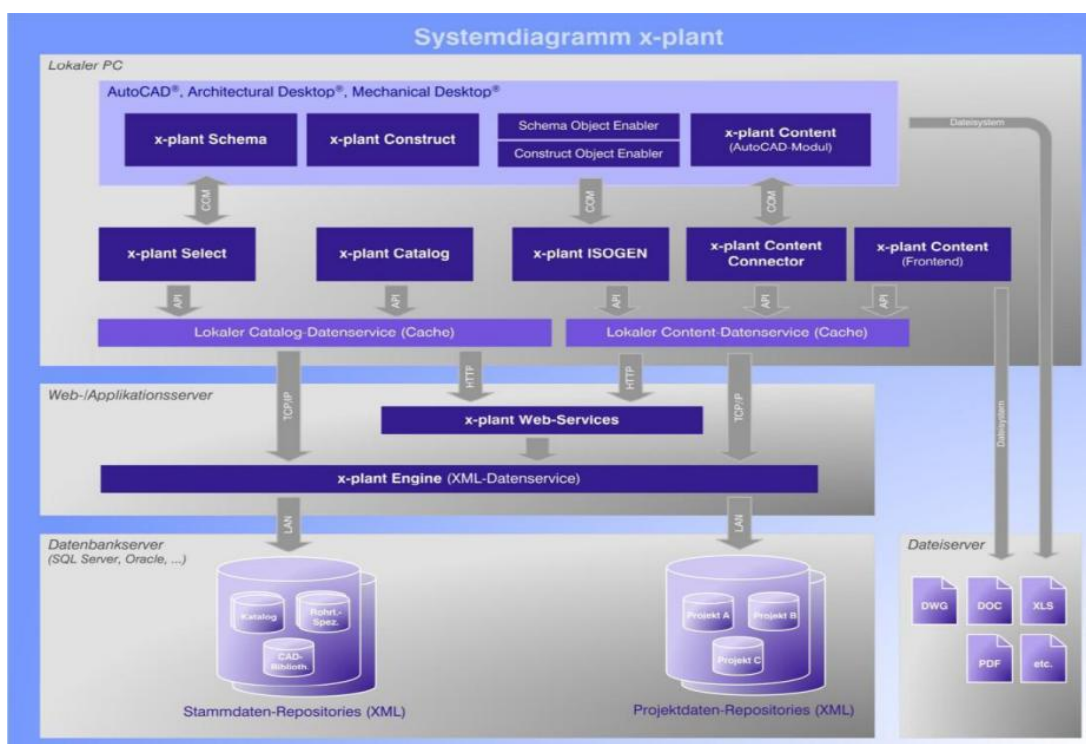


Abbildung 1 – Die x-plant Engine als zentraler XML-Datenservice

Die x-plant Engine bildet das Bindeglied zwischen den Frontend-Modulen von x-plant und den transaktionssicheren Datenspeichern bzw. SQL-Datenbanken, die unter x-plant „Repository“ genannt werden.

Engines

Bei x-plant Engine handelt es sich um einen Windows-Prozess bzw. alternativ Windows-Dienst, der im Hintergrund läuft und beim ersten Start eines x-plant Modules (als Prozess) bzw. beim Hochfahren des Rechners (als Dienst) automatisch gestartet wird. Insofern wird der Anwender von x-plant nie unmittelbar mit der x-plant Engine arbeiten bzw. sie bedienen.

Die x-plant Engine bildet das Bindeglied zwischen den Frontend-Modulen von x-plant und den transaktionssicheren Datenspeichern bzw. Datenbanken, die unter x-plant „Repository“ genannt werden. Zu den übrigen x-plant Modulen stellt sie eine einheitliche Schnittstelle zur Verfügung, die die benötigten bzw. übergebenen Daten als XML-Datenpakete versendet bzw. empfängt. Auf der anderen Seite kommuniziert sie transaktionssicher mit den zur Verfügung stehenden Repositories, die sich auf demselben Rechner oder aber auch auf anderen Rechnern im lokalen Netzwerk befinden.

Repositories

Den eigentlichen Datenspeicher für x-plant bilden die Repositorys. Aus Sicht der Frontend-Module von x-plant sind Repositorys benannte Datenspeicher, die sich auf dem lokalen Rechner oder einem beliebigen anderen Rechner bzw. Server im lokalen Netzwerk befinden können. Der Datenverkehr zwischen den x-plant Modulen und den Repositorys wird von der x-plant Engine verwaltet.

Die Aufgabe eines Repositorys ist es, alle mit x-plant erstellten bzw. verwendeten Datenbestände (Kataloge, Projektdaten, usw.) transaktionssicher und performant zu speichern. Das Datenformat von x-plant ist XML, d.h. dass ein Repository ein XML-Datenspeicher ist.

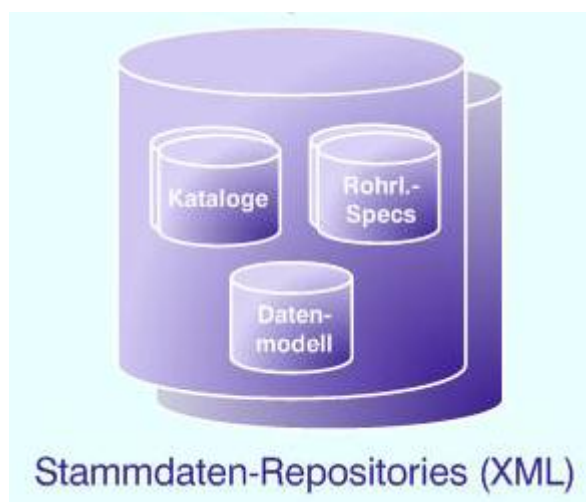


Abbildung 2 Repository-Aufbau

Ein Repository wird derzeit ausschließlich als eine Datenbank-Instanz realisiert. Zurzeit werden Microsoft SQL Server® 2005 und 2008 unterstützt. Das Entscheidende ist jedoch, dass die übrigen Module von x-plant bezüglich der Realisierung von Repositorys unabhängig bleiben. Es ist daher problemlos möglich, von einem Arbeitsplatz aus verschiedene Repositorys anzusprechen, die auf verschiedenen Microsoft SQL Server Instanzen basieren.

Auch die Datensicherung und der Datentransport kann völlig unabhängig vom verwendeten Datenbanksystem durchgeführt werden, da alle Datenbestände in den Repositorys mit speziellen Import- und Exportfunktionen in komprimierter Form (ZIP) als Datei importiert und exportiert werden können. Das verwendete Datenformat (XML) ist unabhängig vom eingesetzten Datenbanksystem.

Jedes Repository enthält sowohl die x-plant Metadaten (Objektklassen und deren Attribute) als auch die eigentlichen Nettodaten (Objekte mit Inhalten). Die Metadaten können von Repository zu Repository durchaus unterschiedlich sein, um z.B. unterschiedlichen Anforderungen an verfügbaren Objektklassen und Attributen in mehreren Projekten gerecht zu werden. Auch ein Datenaustausch zwischen Repositorys mit unterschiedlichen Daten- bzw. Objektmodellen ist möglich.

Administration und Betrieb

Neues Repository erzeugen

Hinweis:

Die Notwendigkeit für die Erzeugung eines neuen Repositories ist bei kleineren Installationen unwahrscheinlich. Nach Möglichkeit sollte dies nur von sachkundigem IT-Personal durchgeführt werden.

ACHTUNG:

Wir empfehlen, unseren Support zu kontaktieren, bevor Sie in Eigenregie neue Repositories anlegen. Wir sind Ihnen dabei gern behilflich.

Ein neues Repository anzulegen bedeutet zunächst einmal, dass eine neue Datenbank erzeugt und für die Verwendung als x-plant Repository eingerichtet werden muss.

Für den Fall eines lokalen Repositories bei Verwendung vom SQL Server Express (2005 oder 2008) existiert dazu eine fertige Batchdatei. Diese erzeugt und konfiguriert die beiden Standard-Repositories „Kataloge“ (xpCatalogs2) und „Projekte“ (xpProjects2). Diese Batchdatei befindet sich im Verzeichnis

<DataBase>\metadata2\repository\rdbms\sqlserver2k\scripts
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)

und heißt

CreateStandardRepositorys.bat

Die Batchdatei verwendet folgende Steuerdateien mit den nötigen SQL-Kommandos, um die Datenbanken anzulegen und zu konfigurieren (bitte Reihenfolge beachten):

Dateiname	Funktion
0_create_login_user2.sql	Erzeugt den SQL Server Benutzer xplantuser2 für den SQL Server (wird anschließend von den x-plant Datenbanken verwendet und damit vorausgesetzt)
1_create_db_xpCatalogs2.sql bzw. 1_create_db_xpProjects2.sql	Erzeugt die Datenbank xpCatalogs2 für das Repository „Kataloge“ bzw. xpProjects2 für das Repository „Projekte“
2_DataModel_2.sql	Erzeugt das Basis-Datenmodell für ein x-plant Repository in einer SQL Server Datenbank (Tabellen, usw.)
3_StoredProc_2.sql	Erzeugt alle notwendigen Stored Procedures für ein x-plant Repository in einer SQL Server Datenbank
4_DatamodelUpdate_CATALOGS.sql bzw. 4_DatamodelUpdate_PROJECTS.sql	Erzeugt weitere Tabellen in den SQL Server Datenbanken für x-plant mit den aktuellen und spezifischen Datenmodellen. Es wird hier

	unterschieden zwischen Datenbanken für Katalog- bzw. Stammdaten (CATALOGS) und Datenbanken für Projektdaten (PROJECTS).
--	---

Um ein neues Repository zu erzeugen und zu konfigurieren, können z.B. Kopien der Batchdatei `CreateStandardRepositorys.bat` angelegt werden, in denen die entsprechenden Anpassungen (Hostname, Login, Passwort und Datenbankname) editiert werden. Durch das anschließende Ausführen der neuen Batchdatei wird die neue Datenbank bzw. das neue Repository physikalisch angelegt und konfiguriert.

Alternativ und bedeutend komfortabler können neue x-plant konforme Datenbanken mit dem *SQL Server Management Studio* angelegt und eingerichtet werden, das mit der x-plant DVD kostenfrei ausgeliefert wird und auch einzeln installiert werden kann. Hier müssen die o.g. SQL Scriptdateien geladen und in der gezeigten Reihenfolge ausgeführt werden.

Repository hinzufügen

Jede x-plant Engine kann beliebig viele Repositories verwalten. Die erreichbaren Repositories werden in einer zentralen Steuerdatei mit XML-Format konfiguriert:

```
<DataBase>\config2\RepositoryIndex2.xml
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)
```

Diese Datei enthält den sog. „Repository-Index“. Hier ist jedes Repository verzeichnet, das von dieser Engine aus erreichbar sein soll. Ein Eintrag für ein Repository besitzt den folgenden Aufbau:

```
<Repository>
  <GUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</GUID>
  <Name>Kataloge</Name>
  <DataProvider>System.Data.SqlClient</DataProvider>
  <Type>MASTERDATA</Type>
  <ConnectionParameter>
    <RepositoryGUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</RepositoryGUID>
    <ParameterName>Data Source</ParameterName>
    <ParameterValue>XP-WS-DEV05\SQL2008XPLANT</ParameterValue>
  </ConnectionParameter>
  <ConnectionParameter>
    <RepositoryGUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</RepositoryGUID>
    <ParameterName>Initial Catalog</ParameterName>
    <ParameterValue>xpCatalogs2</ParameterValue>
  </ConnectionParameter>
  <ConnectionParameter>
    <RepositoryGUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</RepositoryGUID>
    <ParameterName>Integrated Security</ParameterName>
    <ParameterValue>False</ParameterValue>
  </ConnectionParameter>
  <ConnectionParameter>
    <RepositoryGUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</RepositoryGUID>
    <ParameterName>USER ID</ParameterName>
    <ParameterValue>xplantuser2</ParameterValue>
  </ConnectionParameter>
  <ConnectionParameter>
    <RepositoryGUID>D8664316-773B-45a0-BF3D-4EE60C8440F6</RepositoryGUID>
    <ParameterName>PASSWORD</ParameterName>
    <ParameterValue>xplantuser2</ParameterValue>
  </ConnectionParameter>
</Repository>
```

Hier die Erläuterung der verwendeten XML-Tags bzw. Parameter:

Tag	bzw.	Beschreibung des Inhalts bzw. ParameterValue
ParameterName		
GUID	bzw.	Ein gültiger GUID (Global Unique Identifier). Dieser muss innerhalb der XML-Datei eindeutig sein. Innerhalb eines <Repository>-Blocks muss überall derselbe GUID verwendet werden.
RepositoryGUID		
Name		Der Name des Repositories, wie es in den Frontend-Modulen angezeigt wird. Muss innerhalb der XML-Datei eindeutig sein.
DataProvider		Die Voreinstellung bitte nicht verändern.

Type	„MASTERDATA“ für ein Stammdaten-Repository, „PROJECT“ für ein Projektdaten-Repository.
Data Source	Die vollständige Bezeichnung der SQL Server Instanz (Hostname\Instanzname).
Initial Catalog	Der Name der Datenbank.
Integrated Security	Das verwendete Sicherheitsmodell für den Zugriff bzw. Typ des Datenbank-Logins. „False“ = Mit Name und Passwort (Standard, s.u.). „True“ = Mit Windows-Authentifizierung.
USER ID	Der Datenbank-Benutzer, den die Engine für den Zugriff auf das Repository verwendet (nur für Integrated Security = False)
PASSWORD	Das Passwort des Datenbank-Benutzers, den die Engine für den Zugriff auf das Repository verwendet (nur für Integrated Security = False)

Repository entfernen

Sie entfernen ein Repository, indem Sie den entsprechenden Eintrag aus dem Repository-Index (XML-Datei) entfernen.

Das Repository selbst kann durch ein Löschen der zugehörigen Datenbank physikalisch entfernt werden. Nach Möglichkeit sollte dies nur von sachkundigem IT-Personal durchgeführt werden.

Repository anpassen

Anpassungen können im Repository-Index der jeweiligen x-plant Engine durchgeführt werden.

Konfiguration und Anpassungen

Datenmodell

Übersicht

Jedes Repository enthält neben den eigentlichen Nettodaten auch das dazugehörige Datenmodell (die Metadaten), mit dem die verwendeten Objektklassen und deren Attribute definiert werden. Neben diesen grundlegenden Definitionen werden viele weitere Eigenschaften der beteiligten Objektklassen und Attribute definiert, z.B. die lokalisierten Namen, Angaben zum Aussehen und Verhalten eines Datenfeldes, u.v.m.

Die vollständige Beschreibung des Metadatenmodells von x-plant ist Thema eines eigenen Handbuches und soll hier nur in seinen Grundzügen erläutert werden.

Das vollständige Datenmodell wird in Form von XML-Dateien beschrieben. Diese XML-Dateien können in beliebige Repositories importiert werden, wodurch das entsprechende Datenmodell für das jeweilige Repository erzeugt bzw. aktualisiert wird. Dieser Import kann je Repository beliebig oft durchgeführt werden, um Anpassungen am Datenmodell (z.B. neue oder geänderte Attribute) jederzeit übernehmen zu können – auch dann, wenn ein Repository bereits Nettodaten enthält. Dieser Import eines Datenmodells wird auch als Initialisierung eines Repositories bezeichnet.

Die genannten XML-Dateien befinden sich in einer eigenen Verzeichnisstruktur unter dem Basisverzeichnis:

```
<DataBase>\metadata2\xml
```

(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)

Die [Initialisierung eines Repositories](#) kann mit Hilfe von x-plant Catalog durchgeführt werden.

Basisdaten

Jedes Datenmodell benötigt grundlegende Basisdaten, die von den enthaltenen Objektklassen und Attributen referenziert werden. Dazu gehören u.a.:

Unterstützte Sprachen bzw. Lokalisierungen

Physikalische Einheiten und Umrechnungen zwischen Einheiten

Bevor die Objektklassen und Attribute des Datenmodells erstmalig importiert werden, müssen zunächst diese Basisdaten vorhanden sein. Eine Repository-Initialisierung beginnt also stets mit dem Import dieser Basisdaten.

In der Datei

```
<DataBase>\metadata2\xml\xplantbasebase2.xml
```

(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)

werden alle Basisdaten definiert und können dort von entsprechend sachkundigem Personal mit Hilfe eines XML-Editors bearbeitet werden.

Diese Datei wird bei der [Repository-Initialisierung](#) über die Option *Basisdaten* importiert.

Objektklassen und Attribute

Alle in x-plant bekannten Objektklassen werden in einer XML-Datei definiert. In dieser Datei werden die Objektklassen selbst (Namen, Lokalisierungen, usw.) als auch deren vollständiger Attributsatz mit allen Eigenschaften beschrieben.

Der Ablageort und Name dieser XML-Dateien lautet:

```
<DataBase>\metadata2\xml\schema\xplantbaseschema2.xml  
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)
```

Diese XML-Datei sollte nach Möglichkeit nicht geändert bzw. erweitert werden, da sie das Standard-Datenmodell enthält, in dem wir uns jederzeit Änderungen und Ergänzungen vorbehalten, die im Rahmen von Patches und Updates verteilt werden und somit kundenspezifische Änderungen überschreiben würden.

Für **kundenspezifische Änderungen** existiert daher ein spezieller Mechanismus, der über eine optionale zweite XML-Datei im genannten Verzeichnis realisiert wird: `xplantcustomschema2.xml`. Diese Datei kann von entsprechend sachkundigem Personal mit Hilfe eines XML-Editors bearbeitet werden, um das Standard-Datenmodell kundenspezifisch zu erweitern oder zu modifizieren. Wenn Sie daran Interesse haben, nehmen Sie bitte Kontakt mit unserem Support auf. Wir bieten für diese Themen spezialisierte Schulungen an.

Die Objektklassen werden bei der [Repository-Initialisierung](#) über die Option *Objektklassen* importiert.

Abfragen, Ansichten und Reports

Die Darstellung eines Objekts einer bestimmten Objektklasse (z.B. Flansch) in den Frontend-Modulen von x-plant wird durch Abfragen und Ansichten definiert.

Abfragen unterteilen die verfügbaren (Objekt-)Ansichten grundlegend für die Verwendung in Katalogen und Rohrleitungsspezifikationen auf der einen Seite (x-plant Catalog und Select) und in Projekten auf der anderen Seite (x-plant Content), da die sichtbare Ausprägung eines Objektes mit Attributen in einem Katalog und als eingefügte Komponente in einem Projekt durchaus unterschiedlich sein kann.

x-plant wird mit einem Satz von Standard-Abfragen ausgeliefert, die für den normalen Einsatz von x-plant völlig ausreichend sind. Diese Abfragen steuern u.a. die Anzeige von Objekten in Navigationsbäumen, Formularen und Tabellen – und dies getrennt für x-plant Catalog bzw. Select und x-plant Content.

Abfragen werden in der XML-Datei

```
<DataBase>\metadata2\xml\xplantbasequeries2.xml  
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)
```

definiert. Diese Datei wird bei der [Repository-Initialisierung](#) über die Option *Abfragen und Reports* importiert.

Ansichten werden für jede verfügbare Objektklasse definiert. Somit kann für jede Objektklasse individuell festgelegt werden, welche Attribute bzw. Datenfelder in welcher Reihenfolge in den Frontend-Modulen von x-plant angezeigt werden. Jede Objektklasse besitzt mehrere Ansichten – die Anzahl resultiert aus den zur Verfügung stehenden Abfragen (s.o.). So existiert z.B. eine Ansicht für den Navigationsbaum und die Tabelle (Detaildaten) von x-plant Catalog, eine weitere für den Navigationsbaum in x-plant Content, usw.

Die Ansichten werden in einer XML-Datei definiert. Der Ablageort und Name dieser XML-Dateien lautet:

```
<DataBase>\metadata\xml\schema\xplantbaseviews2.xml  
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)
```

und wird bei der [Repository-Initialisierung](#) über die Option *Ansichten* importiert.

Diese XML-Datei sollte nach Möglichkeit nicht geändert bzw. erweitert werden, da sie die Standard-Ansichten enthält, in denen wir uns jederzeit Änderungen und Ergänzungen vorbehalten, die im Rahmen von Patches und Updates verteilt werden und somit kundenspezifische Änderungen überschreiben würden.

Für **kundenspezifische Änderungen** existiert daher ein spezieller Mechanismus, der über eine optionale zweite XML-Datei im genannten Verzeichnis realisiert wird: `xplantcustomviews2.xml`. Diese Datei kann von entsprechend sachkundigem Personal mit Hilfe eines XML-Editors bearbeitet werden, um das Standard-Datenmodell kundenspezifisch zu erweitern oder zu modifizieren. Wenn Sie daran Interesse haben, nehmen

Sie bitte Kontakt mit unserem Support auf. Wir bieten für diese Themen spezialisierte Schulungen an.

Die zur Verfügung stehenden **Reports** und die ihnen zugrunde liegenden Abfragen für Kataloge, Rohrleitungsspezifikationen und Projektdaten (Stücklisten) werden ebenfalls in den Repositorys gespeichert. Neben den bereits beschriebenen Standardabfragen für die Frontend-Module von x-plant existieren daher noch **Report-Abfragen**, mit denen die Datenfelder festgelegt werden, die für Reports bzw. den Report-Designer zur Verfügung gestellt werden, wobei die Datenfelder hier oft Klassen-übergreifend definiert werden können, da ein Report typischerweise Objektdaten mehrerer Objektklassen enthalten kann (z.B. Rohre, Bögen, T-Stücke, usw.).

Die Report-Definitionen und die zugrundeliegenden Abfragen werden mit der Report-Verwaltung von x-plant Catalog und x-plant Content verwaltet. Die Abfragen werden wiederum in XML-Dateien definiert und geändert. Wenn Sie daran Interesse haben, nehmen Sie bitte Kontakt mit unserem Support auf. Wir bieten für diese Themen spezialisierte Schulungen an.

Sonstige Konfigurationsdateien

XPLANTCONFIG2.XML

In der Konfigurationsdatei

```
<DataBase>\config2\xplantconfig2.xml
(<DataBase> ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)
```

werden wichtige Systemparameter für den Betrieb von x-plant definiert. Es handelt sich hauptsächlich um Pfad- und Dateinamen, über die die Struktur einer x-plant Installation definiert wird. Die enthaltenen Pfade können z.B. bei einer verteilten Netzwerkinstallation nachträglich angepasst werden, um diverse Suchpfade auf zentrale Suchpfade umzustellen, die sich auf einem Fileserver befinden.

Die Konfigurationsdatei ist eine XML-Datei und besitzt den folgenden Aufbau:

```
<XPCConfigDataSet>
  <Common>
    <ReportTemplatePath>.....</ReportTemplatePath>
    <TempPath>.....</TempPath>
    <BinPath>.....</BinPath>
    <DataBasePath>.....</DataBasePath>
    <VariantTemplatePath>.....</VariantTemplatePath>
    <SettingsPath>.....</SettingsPath>
    <DataPath>.....</DataPath>
    <SelectCachePath>.....</SelectCachePath>
    <BlockFilesBasePath>.....</BlockFilesBasePath>
    <CADViewerEXE>.....</CADViewerEXE>
    <CADViewerParameters>.....</CADViewerParameters>
    <AutoCADVersion>.....</AutoCADVersion>
    <DNExpression>DN 0;;</DNExpression>
  </Common>
  <Engine>
    <EngineIndexFilename>.....</EngineIndexFilename>
    <EngineIndexXMLSchema>.....</EngineIndexXMLSchema>
    <RepositoryIndexFilename>.....</RepositoryIndexFilename>
    <RepositoryIndexXMLSchema>.....</RepositoryIndexXMLSchema>
    <MetaDataBasePath>.....\metadata</MetaDataBasePath>
  </Engine>
  <AcadProfiles>
    <Standard>x-plant2_complete</Standard>
    <Schema>x-plant2_schema</Schema>
    <Construct>x-plant2_construct</Construct>
  </AcadProfiles>
</XPCConfigDataSet>
```

Hier die Erläuterung der verwendeten XML-Tags:

Tag	Inhalt
ReportTemplatePath	Das Verzeichnis, in dem die Reportvorlage-Dateien für x-plant gespeichert werden.
TempPath	Verzeichnis, in dem x-plant temporäre Dateien ablegt. Auch die bei Exports entstehenden ZIP-Dateien werden hier vorgabemäßig abgelegt.
BinPath	Das Verzeichnis, in dem sich die Programme, DLL-Dateien und ActiveX-Dateien von x-plant befinden.
DataBasePath	Das Basisverzeichnis für alle Datendateien von x-plant.
VariantTemplatePath	Das Verzeichnis, in dem die Variantenvorlagen-Dateien (XML) für x-plant Construct bzw. x-plant Catalog vorgabemäßig gespeichert werden.
SettingsPath	Verzeichnis, in dem x-plant diverse Steuerdateien ablegt, die während der Benutzung entstehen. Diese Dateien können gelöscht werden, wobei aber die entsprechenden Einstellungen verloren gehen.
DataPath	Verzeichnis für diverse allgemeine Datendateien, die von x-plant verwendet werden.
BlockFilesBasePath	Das Basisverzeichnis für diverse AutoCAD-Blockdateien, die von x-plant Construct bzw. x-plant Schema verwendet werden.
CADViewerEXE	Der vollständige Dateiname (inkl. Pfad) für den Start eines externen DWG-Viewers (EXE-, BAT- oder CMD-Datei).
CADViewerParameters	Parameter für Aufruf des externen DWG-Viewers. Der Platzhalter „%1“ wird für den Dateinamen der anzuzeigenden Datei verwendet.
AutoCADVersion	Die verwendete AutoCAD-Version (derzeit sind die Werte ‚2007‘, ‚2008‘, ‚2009‘, ‚2009_X64‘, ‚2010‘ und ‚2010_X64‘ zulässig).
EngineIndexFilename	Obsolet. Wird in einer zukünftigen Version nicht mehr existieren.
EngineIndexXMLSchema	Obsolet. Wird in einer zukünftigen Version nicht mehr existieren.
RepositoryIndexFilename	Der vollständige Dateiname des Repository-Index (XML).
RepositoryIndexXMLSchema	Der vollständige Dateiname des XML Schemas für den Repository-Index.
MetaDataBasePath	Basisverzeichnis für alle Metadaten (Objektklassen, Abfragen, Icons, usw.).
SelectCachePath	Verzeichnis, in dem x-plant Select den Selektions-Cache aufbaut und verwaltet.
Standard (Schema, Construct)	Der Name des AutoCAD-Profiles, das für den Start von AutoCAD aus x-plant Content verwendet wird.
DNEExpression	Die zentrale Maske für die Formatierung von DN-Werten. Mit Hilfe dieses Ausdrucks werden alle DN-Angaben in x-plant (in den Frontend-Modulen und in Zeichnungen) formatiert. Die Angabe „0;“ repräsentiert den DN-Wert. Der übrige Text dient zur Formatierung. Bei einem DN-Wert von 100 produziert die Standardmaske „DN 0;“ den Text „DN 100“.

SCALES.XML

In der Konfigurationsdatei

```
<DataBase>\data2\scales.xml
```

(`<DataBase>` ist das bei der Installation angegebene Basisverzeichnis für Datenbestände)

werden die von der Maßstabsfunktion in x-plant Construct angebotenen Maßstäbe definiert.

Die Konfigurationsdatei ist eine XML-Datei und besitzt den folgenden Aufbau:

```
<Scales ...>
  <Scale>
    <Text>1:1</Text>
    <Factor>1</Factor>
  </Scale>
  <Scale>
    <Text>1:2</Text>
    <Factor>2</Factor>
  </Scale>
  <Scale>
    <Text>1:5</Text>
    <Factor>5</Factor>
  </Scale>
  <Scale>
    <Text>1:10</Text>
    <Factor>10</Factor>
  </Scale>
  <Scale>
    <Text>1:20</Text>
    <Factor>20</Factor>
  </Scale>
  .....etc...
</Scales>
```

Jeder Maßstab wird durch einen `<Scale>`-Tag definiert. Die bestehenden Maßstäbe können frei editiert oder es können neue Maßstäbe hinzugefügt werden.

Hier die Erläuterung der verwendeten XML-Tags:

Tag	Inhalt
Text	Der angezeigte Maßstabs-Text (z.B. „1:100“)
Factor	Der Kehrwert des Maßstabes (z.B. 100 für 1:100)