XML ist hervorragend dazu geeignet, generische Anwendungen wie z.B. Installationsroutinen zu erzeugen

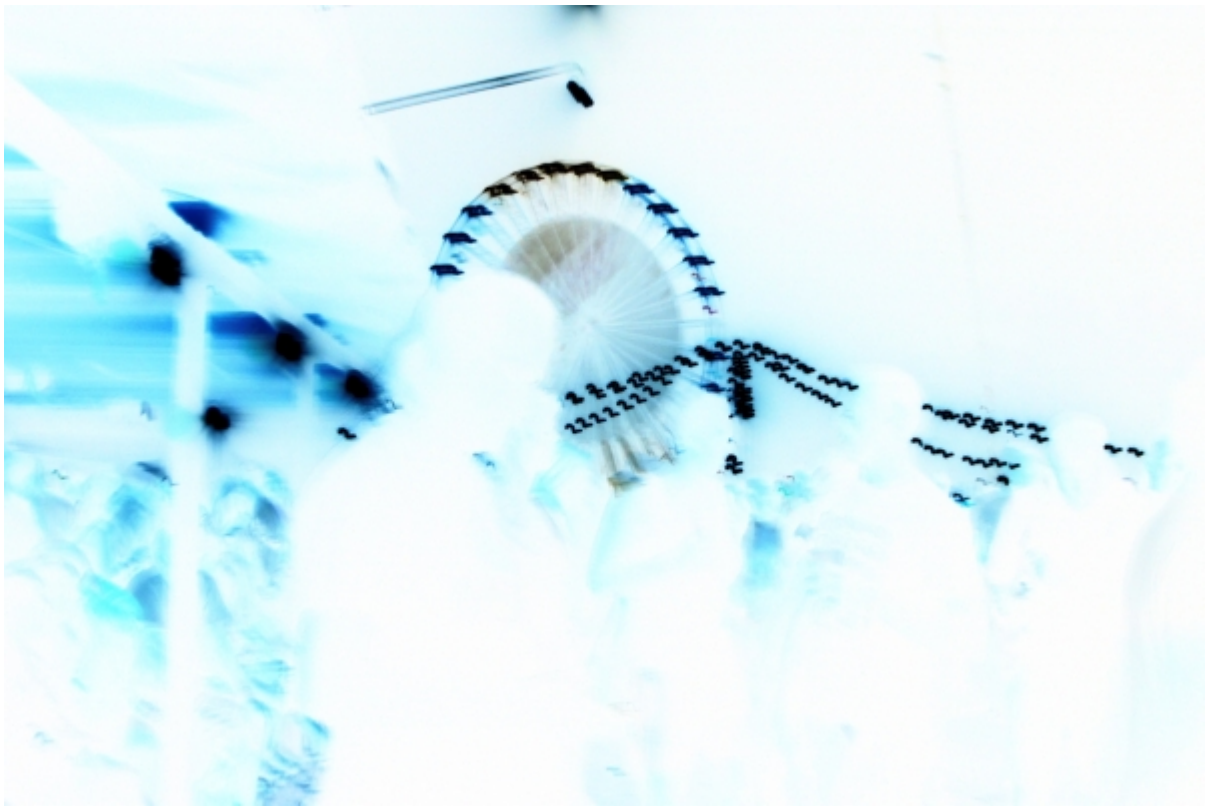
von Craig Rutledge

Der Downloadbereich enthält folgende Codes zu diesem Artikel:

März Utility der NEWSolutions

Load´n´go

- XML ist hervorragend dazu geeignet, generische Anwendungen wie z.B. Installationsroutinen zu erzeugen von Craig Rutledge
- XMLGEN CMD Script
- XMLGEN CLP Befehlsdefinition
- XMLGENC CLP XML-Member generieren
- XMLGENCMD CMD Script-Befehle prompten
- XMLGENH PNLGRP Hilfedatei
- XMLGENINC CMD Zusatzbefehle prompten
- XMLGENMBR CMD Teildatei-Informationen prompten
- XMLGENR RPGLE XML-Code generieren
- XMLGENRV RPGLE Gültigkeitsprüfung für Scripts
- Lies mich Win95



Der Quellcode, die Attribute und die Erstellungsbefehle werden komplett in einem XML-Dokument zusammengestellt, das über das Web, per eMail oder mit dem CL-Befehl SNDNETF (Netzwerkdatei senden) übertragen werden kann. Sie finden den Quellcode (V4) für mein Utility im Web unter www.newsolutions.de .

XML-Grundlagen

Bevor ich den Generator und den Installer, die wichtigsten Bestandteile meines Tools, genauer beschreibe, möchte ich kurz auf einige XML-Grundbegriffe eingehen, die Sie kennen müssen, um alles weitere verstehen zu können. Das Grundkonzept von XML ist wie bei HTML das Einschließen von Daten in aussagefähige Markierungen, die sogenannten Tags. Diese Tags können bei XML ein beliebiges Wort enthalten. Sie sollten aber darauf achten, dass Tags die enthaltenen Daten beschreiben. Ein Beispiel:

<nachricht>Hallo Welt</nachricht> oder <sonderpreis>1.30</sonderpreis>

Tags können auch verschachtelt werden, um Informationen zu gliedern:

<source_member> <memer_name>ABCMBR</member_name> </source_member>

Außerdem kann die Bedeutung von Tags genauer definiert werden, indem man ihnen Attribute zuordnet:

<mbr mbrname="CNTRPGDP" mbrtype="PF"> Daten </mbr>

In diesem Beispiel wurden dem Tag die Attribute mbrname und mbrtype zugeordnet. Im Zusammenhang mit XML sind häufig die Begriffe well formed und validated zu finden. Well formed (wohlgeformt) bedeutet, dass der gesamte Inhalt eines Dokuments in Tags eingeschlossen ist, und dass zu jedem öffnenden Tag (<>) ein zugehöriger schließender Tag (</>) existiert. Validated (gültig, geprüft) bedeutet, dass das XML-Dokument einer DTD (Document Type Definition) entspricht. Die DTD ist ein Text, mit dessen Hilfe der Browser feststellen kann, ob eine generierte Datei nur richtig definierte Tags enthält. XML ist einfach ein Übereinkommen zwischen zwei Prozessen über die Bedeutung von Tags und XMLGEN basiert auf diesem Prinzip.

XMLGEN verwenden

XMLGEN packt auf dem Quellsystem ausgewählte iSeries-Quellcodes und -Befehle zwischen XML-Tagszusammen und bereitet sie als generierte Ausgabedatei auf. Der Parser/Installer verwendet diese Tags, um die Wiederherstellung der Quellcodes und der Objekte auf dem Zielsystem zu steuern. Um mit dem Befehl XMLGEN ein XML-Dokument erstellen zu können, das well formed und validated ist, sind zwei Schritte erorderlich: Als erstes muss ein Scripting-Eintrag erstellt werden, und danach muss der Befehl XMLGEN ausgeführt werden. Mehr müssen Sie nicht tun, um ein komplett abgeschlossenes, gültiges und wohlgeformtes XML-Dokument zu erstellen, das sich selbst auf jeder iSeries intallieren kann!

Einen Script-Eintrag erstellen

XMLGEN verarbeitet ein einfach zu erstellendes Script aus einer Quellenteildatei. Dieses Script wird vom XML-Generator gelesen, der dann den Quellcode und die Befehle lädt, die im Paket enthalten sein sollen. Diese Aufgabe kann auf verschiedene Art erledigt werden. Martin Rowe hat ein flexibles Verfahren entwickelt, das drei neue Befehle zur Unterstützung der Script-Erstellung beinhaltet. Der erste Teil eines Scripts, den Sie bei jedem XMLGEN-Befehl verwenden werden, ist die Anweisung XMLGENINC (XML Generator Include), mit der festgelegt wird, ob der XML-Parser zum Paket hinzugefügt werden soll. Wenn der Parser-Code beötigt wird, um die XML-Datei wieder in Quellcodes und Objekte aufzulösen, so wird er mit XMLGENINC zum Bestandteil des Pakets gemacht. Der Parser muß nur einmal auf dem Zielsystem installiert werden. Der zweite im Script zu verwendende Befehl ist XMLGENMBR (XML Generator Member), der beschreibt, welche Dateien

zum XML-Paket hinzugefügt werden sollen.

> XMLGENMBR XMBR(Member) + XMBRATR(MemberType) + XFROMSRCF(FromSourceFile) + XFROMSRCL(FromLibrary) + XTOSRCF(ToSourceFile) + XOBJTYPE(ObjType) + XGENCRT(Generate *YES/*NO)

Diese Anweisung enthält alle Informationen, die benötigt werden, um eine Quellenteildatei zum Paket hinzuzufügen und das Objekt auf dem Zielsystem wiederherzustellen. Wenn z.B. bei XOBJTYPE der Wert *PGM eingetragen ist und die Quellenart der Quellenteildatei RPGLE ist, dann wird automatisch der erforderliche Umwandlungsbefehl CRTBNDRPG zwischen zwei „Compile“-Tags in die XMLDatei eingefügt:

> – <compile> – <![CDATA[CRTBNDRPG PGM(&tolib/WEB_CRYPT) SRCFILE(&tolib/QRPGLESRC)]]> </compile> – <sendmsg sendmsgid=“CPF9897″ sendmsgtype=“*COMP“>

(Dieses Beispiel zeigt die generierte XML-Datei, die weiter unten noch genauer beschrieben wird.) Unter bestimmten Umständen sind sowohl vor als auch nach dem Kompilieren zusätzliche Aktionen erorderlich. Zusätzlich zu den normalen Umwandlungs-Optionen kann es z.B. erforderlich sein, dass QTEMP-Objekte, Bindeverzeichnisse und Serviceprogramme erstellt werden. Außerdem kann es vorkommen, dass Code mit bestimmten Optionen als *MODULE oder *SRVPGM kompiliert wird. Die Lösung für derartige Probleme ist die Anweisung XMLGENCMD (XML Generator Command), die das Scripting von bestimmten Befehlen in der vom Programmierer festgelegten Reihenfolge ermöglicht. XMLGENCMD erwartet als Parameter einfach eine komplette Befehlszeichenfolge. Ein Beispiel:

> XMLGENCMD XCMD(CRTPF FILE(QTEMP/TEMPOUT) RCDLEN(528))

Mit dieser Anweisung wird eine Datei in der Bibliothek QTEMP erstellt, die z.B. zum erfolgreichen Erstellen eines Objekts erforderlich sein könnte. Sie können auch einen Befehl wie den folgenden verwenden, um Befehle mit nicht standardmäßigen Einträgen zu erstellen:

> XMLGENCMD XCMD(CRTCMD CMD(&TOLIB/CPYSAVFXML) PGM(&TOLIB/CPYSAVFXMC) SRCFILE(&TOLIB/QSRC))

Das Script wird in einer normalen CL-Quellenteildatei erstellt, so dass Sie Ihre Paketbeschreibungen genauso einfach editieren, speichern und kopieren können wie normale Quellenteildateien.
Hinweis: Wenn Sie vorhaben, eine größere Anzahl von Quellen und/oder Befehlen in Ihr Script einzubinden, empfehle ich Ihnen, das Utility XMLSVIEW herunterzuladen. Es fasst alle im Script enthaltenen Informationen in einer übersichtlichen Subfile-Anzeige zusammen. Damit können Sie sich sehr einfach einen Überblick über die von Ihnen gewählten Parameter verschaffen.

Aufrufen des Befehls XMLGEN

Nachdem Sie das Script fertiggestellt haben, können Sie den Befehl XMLGEN aufrufen. Abbildung 1 zeigt die Bedienerführung für XMLGEN.

Gültigkeitsprüfung

Das erste auszuführende Programm ist das in in ILE-RPG geschriebene Gültigkeitsprüfprogramm XMLGENRV. Seine Aufgabe ist es, die Gültigkeit und die Verfügbarkeit der im Script enthaltenen Angaben zu prüfen. Als erstes werden die vom Befehl XMLGEN übergebenen Parameter verarbeitet. Der Teildateiname des Scripts und derName und die Bibliothek der Quellendatei werden in die

Felder gestellt, die von der Subroutine srValidate verwendet werden. Diese Subroutine arbeitet mit dem Fehlercode, den das API QUSRMBRD (Retrieve Member Description) zurückgibt. Abhängig von der ausgegebenen Nachrichten-ID wird mit dem API QMHSNDPM eine Fehlernachricht an den Benutzer zurückgesendet. Als nächstes werden die Namen der Ausgabedatei und ihrer Bibliothek geladen, und die Subroutine srValidate wird noch einmal ausgeführt. Danach wird die Eingabedatei mit dem Namen der ausgewählten Script-Teildatei überschrieben und für Eingabe geöffnet. Bei der satzweisen Verarbeitung der Script-Datei wird jeder gelesene Satz nach dem Namen des Script-Befehls durchsucht, der entweder bestimmte Teildateiangaben oder auszuführende Befehle enthält. Wenn ein Satz gefunden wird, der einen dieser Befehle enthält, wird die Subroutine srLoadString ausgeführt.