AMRI.APP - Assistent für Referentielle Integrität

Markus Auer, Markus Winhard

Die neue Version des AMRI-Builders steht im Dateibereich zur Verfügung. Beim Suchen im Dateibereich bitte AMRI (in Großbuchstaben) eingeben, sonst findet die Suche nichts.

Version 4.00

Voraussetzung VFP 9.0, VFP 8.0, VFP 7.0, VFP 6.0, VFP 5.0 oder VFP 3.0b

Beschreibung:

Ersatz für den VFP-eigenen Builder.

Vorteile dieses Builders:

- Integriertes Audit Trail (Protokollieren aller Änderungen).
- RI Code abschaltbar (_RI_Off=.T.). Wichtig z.B. für Datenimporte die sonst zu lang dauern würden.
- Auswerten der Ursache beim Fehlschlagen eines Triggers, z.B. beim Löschen eines Datensatzes: Weshalb wurde das Löschen verhindert? Welcher Datensatz in welcher anderen Tabelle hat das Löschen verhindert?

_RI_GetMessage

- Allgemeine Hooks für Insert/Update /Delete die bei jedem Trigger aufgerufen werden. Damit kann manuell die RI sichergestellt werden, wenn Schlüsselwerte aus anderen Datenbanken oder freien Tabellen in diesem DBC verwendet werden.
- Debug-Modus für Trigger (TriggerDebug.txt, TriggerInfo.txt).
- Dieser Builder erzeugt einen wesentlich kleineren Code der von VFP schneller verarbeitet wird.
- Fehler und Schwaechen des VFP-Builders behoben.
- Der erzeugte Code ist fuer Debugging-Zwecke leicht verstaendlich.
- Durch wahlweise Empty()- oder Null()-Unterstuetzung ist das Anfuegen mit APPEND BLANK moeglich, bzw. koennen Referenzen in den Daten bewusst offen gelassen werden. Empty()-Unterstützung ist sinnvoll bei der Übernahme von FP2x-Datenbeständen in denen ein leerer Fremdschlüssel bedeutete, daß für diesen Satz keine Beziehung zur Parent-Tabelle besteht.
- Durch einen allgemeinen Trigger ist es möglich, die Manipulation der Daten global zu sperren.
- Im Gegensatz zu VFPs RI-Generator wird der Trigger-Code für Compound Keys korrekt erzeugt. Ausnahme: Cascading Update, Delete Nullify, Delete Default. Zur Erinnerung: VFPs RI-Generator erzeugt mit Compound Keys fehlerhaften Code. Delete Nullify und Delete Default beherrscht VFPs RI-Generator überhaupt nicht.
- Im Gegensatz zu VFPs RI-Generator führt Cascading Delete TabelleA>TabelleB gefolgt von Delete Restrict TabelleB>TabelleC nicht zu inkonsistenten Daten.
- Datenbank wird vor der Generierung des Triggercodes automatisch gepackt.

HISTORY:

V4.00 26.02.2005, Markus Winhard

- RI_Setup.app erkennt jetzt Vfp9 und zukünftige Vfp Versionen.
- Errorhandler der Applikation aufrufen, wenn im Triggercode ein Laufzeitfehler aufgetreten ist (abschaltbar per #DEFINE in StdCode.prg).
- _RI_Error() füllt jetzt auch m._RI_Message. Damit bekommt man z.B. nach dem Fehlschlagen eines Triggers aufgrund eines defekten CDX die zugehörige VFP Fehlermeldung (und damit die eigentliche Fehlerursache) heraus.
- _RI_GetMessage kann jetzt mit mehrzeiligen Meldungen in m._RI_Message umgehen.
- Bugfix: _RI_GetMessage ignorierte SET MEMOWIDTH.
- Bugfix: Aufsplitten des Strings der _RI_Message zugewiesen wird in mehrere Teile, wenn er länger als 255 Zeichen ist.
- Bugfix: _RI_Update() aktualisierte seit Version 3.10 meist nicht alle Fremdschlüssel.

V3.30 08.03.2004, Markus Winhard:

- Delete Nullify konnte nicht mit Feldern umgehen die kein e Werte unterstützen. Jetzt wird in diesem Fall das Feld geBLANKt.
- Ri_End() hat als ersten Parameter jetzt zusätzlich den Namen der Tabelle für die gerade ein Trigger aufgerufen wurde.