

Progress to XML based schema conversion routines

www.newnhams.com

Pro2Xmlschema

Version 1.7

20th Oct 2004

Product Overview

Pro2Xmlschema is a simple group of PROGRESS source code, and PHP routines that can be run against any PROGRESS database to produce a series of files containing database independent XML schema and data files. These files can be used to produce a database that is a mirror of the PROGRESS database.

The following databases have been tested against the sports2000 database

- MySQL
- SQLite
- Oracle
- IBM DB2
- Microsoft SQL server

Other databases may be supported. For a list of database that may work, check the documentation at adodb.sourceforge.net, which is the home page for the database abstraction layer that is used to create the new databases.

The routines provide a method of:

- Extracting the data schema from the PROGRESS database
- Checking and converting PROGRESS data to prevent MySQL database load errors.
- Extracting the data from the PROGRESS database.

System Requirements

PROGRESS

The Pro2xmlschema conversion routines dynamically create and run PROGRESS programs. They also include routines that update data in the database. To run these routines, a PROGRESS development license Version 9 must be installed on the machine from which the routines are run.

Other Database

You must have the appropriate database management software installed

PHP

Download and install PHP from www.php.net. These routines only use the PHP CLI interface, so a web server is not required. However, this document doesn't help you establish a PHP connection to your chosen database. There are some resources available to help.

- The adodb documentation and support forum at adodb.sourceforge.net
- The PHP documentation and mailing lists
- Google
- If all else fails, email the author and I'll try to help.

Licensing

Pro2XMLscheme is issued under the terms of the BSD license. This means you can incorporate it into your own software, commercial or otherwise.

Transferring the schema.

Dumping the schema from PROGRESS.

1. Start a progress editor session against the database you want to dump.
2. Load the pro2xml.p file and run it.
3. Select a directory to dump the schema to.
4. Select Dump Type of Schema.
5. If you want to transfer the existing indexes to the new database, select **Dump index definitions with schema**. In any case, a unique primary index is created based on the existing PROGRESS recid. The names of the indexes will not be reused, as many databases require unique naming conventions across all indexes.
6. You can force the new database to try to create unique indexes that match those in the Progress databases by checking the **Create Unique Indexes Where Necessary** option. This may cause problems with some databases.
7. If you want to transfer the data dictionary information, select **Create Data Dictionary** option. This converts the PROGRESS _field and _file tables for reference purposes.
8. Select the Tables you want to convert.
9. Select OK.
10. The contents will be dumped into an XML file named *xmlschema_dbname.xml* in the selected output directory. You can review this file in any browser or xml editor.

Dealing with data that is wider than the progress format

If you store character data that does not conform to the progress format. (fields wider than the display size defined in the dictionary), you can force the dump program to use the SQL width by setting USE_SQL_WIDTHS to true at the top of pro2xmlschema.p. Search the progress knowledgebase for more information on SQL widths

Data Type Conversion

The data types are converted to 'MetaTypes', that is, database independent types that can be recreated as the appropriate data type in the target database. The ADODB database library handles this transparently.

Progress data type	Meta Type	Notes
<i>Character</i> field length <= 255	<i>C</i>	Includes appropriate size information.
<i>Character</i> field length > 255 characters.	<i>X</i>	
<i>Date</i>	<i>D</i>	Converted to YYYY-MM-DD format
<i>Logical</i>	<i>L</i>	False = 0 True = 1.
<i>Decimal</i>	<i>N</i>	Include length and precision
<i>Integer</i> less than 3 digits	<i>I2</i>	
<i>Integer</i> between 3 and 4 digits	<i>I4</i>	
<i>Integer</i> over 4 digits	<i>I8</i>	
<i>Recid</i>	<i>I8</i>	
<i>Raw</i>	<i>X</i>	The data is not converted.

Index conversion.

There are 3 specific index conversion items

1. The creation of a new non-null primary index using the PROGRESS recid.
2. Word indexes are ignored
3. Indexes are assigned a unique name based on the table.

Other schema items

Mandatory fields are written as NOT NULL

Default values are written directly into the target database with the equivalent value, except for the PROGRESS unknown value, which is written as NULL for character and date fields, and zero for numbers.

Arrays

The Pro2Xmlschema routines expand array variables into individual fields. Each field is named as follows:

```
array[1], array[2],array[3],.....array[n]
```

Becomes

```
array__1,array__2,array__3,.....array__n
```

It should however be noted that in most languages that are used to develop programs against other databases such as PERL or PHP, arrays and loops normally start at value 0. To achieve this, change the variable `ARRAYS_START_AT_0` to TRUE at the top of `pro2xml.p`. The data will be dumped with a corresponding index shift.

Field and File Name conversion

The Pro2XMLSchema routines make various changes to the field and file names to ensure they do not conflict with reserved words in the target database. Many databases do not currently have a reserved words list available, if they do not, they use the MySQL reserved words table. These words are stored in `pro2xml.p`. Before the conversion, the following rules are applied.

1. All field/file/database names are converted to lower case. Depending on your requirements, change the variable `NAME_CASE` to the appropriate value at the top of `pro2xml.p`
2. All occurrences of the "-"(minus) character are converted to an "_"(underscore).
3. The resulting variable name is compared against a list of reserved words for the database. If the variable is a reserved word, it is prefixed with "X_", which will at least allow the schema to load into the database.
4. Any fields that begin "_" are prefixed with an X. DB2 (At least) doesn't like fields beginning with _.
5. To see all these features in action, run ***pro2xml.p*** program against the SPORTS2000 database, and view the output. Note that the Order table has been renamed X_ORDER.

TIP In case of problems, is far easier to change the PROGRESS database using the dictionary tools before dumping, than to dump the database and try to manipulate the schema using tools in the target database.

Dumping the data

Dumping the data from PROGRESS.

1. Start a progress editor session against the database you want to dump.
2. Load the pro2xml.p file and run it.
3. Select a directory to dump the data to.
4. Select Dump Data.
5. If you want to transfer the data dictionary information, select *Create Data Dictionary* option. This converts the data in the PROGRESS _field and _file tables for reference purposes.
6. Select the Tables you want to dump.
7. Select OK.
8. The contents will be dumped into XML files named **table.xml**, **table_1.xml**, **table_2.xml**.. in the selected output directory. You can review these files in any browser or XML editor.

Limitations of the data dump.

The data dump currently cannot process raw data fields. Data in fields defined as raw format is replaced by a null value on dump.

OK – So why did it dump all these XML datafiles?

There is a limitation in the XML handler, where it loads all of the data into memory first before processing it. Files bigger than about 500k (approx) cause the load routine to blow up. Don't worry, however, the load program knows which order to load them.

Loading the schema and data

The schema and data files that you have created are compatible with the *xPollinate* load routines. Read the section on loading schema and data files in the *xPollinate* manual

The Rest

That's it ! enjoy, report bugs and successful conversions to other databases, as well as suggested improvements to mark@newnhams.com. Thanks to the ADODB, and ADODB-XMLSCHEMA projects for such great work