

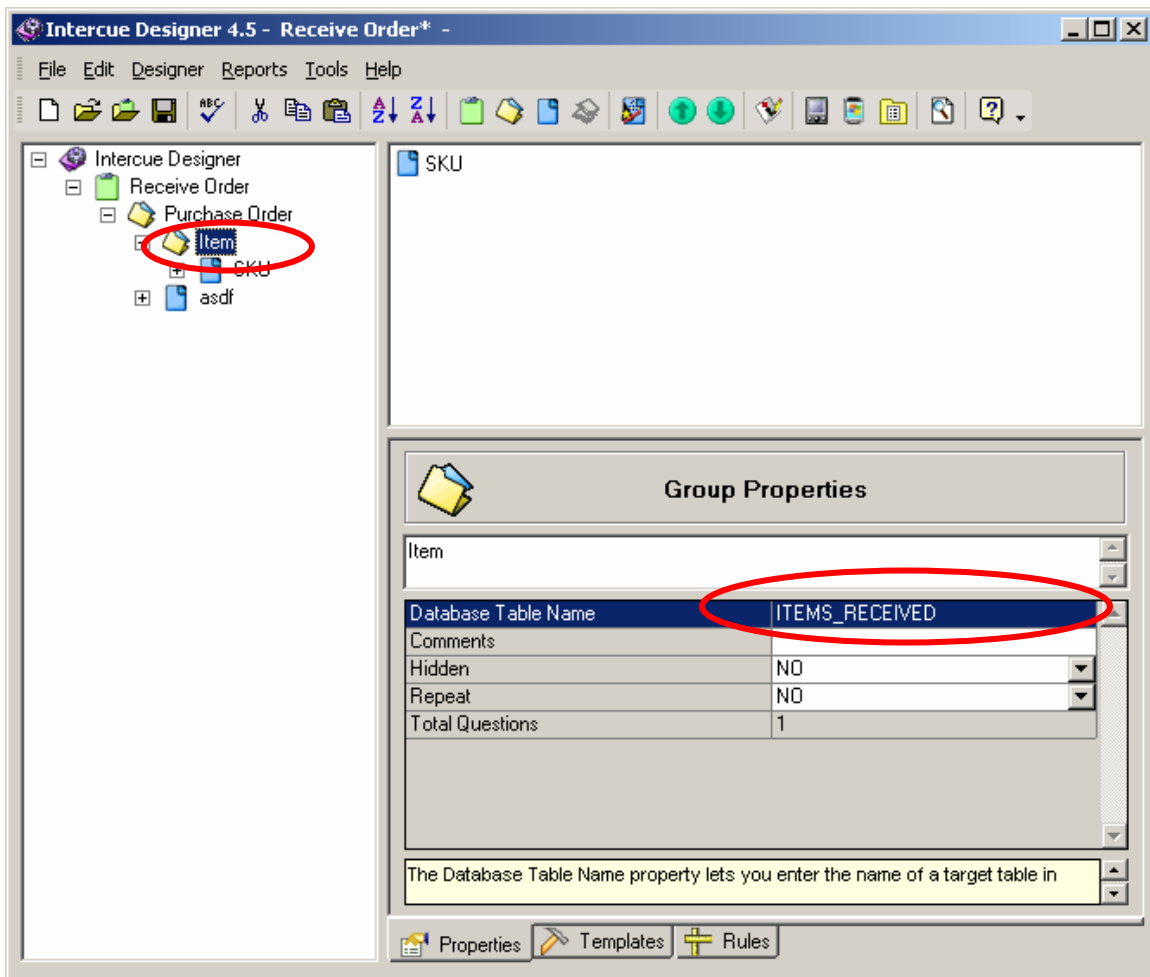
Are database table relationships supported in Intercue Mobility Suite Forms?

MobileDataforce (MDF) provides “export” functionality to get data out of MDF Desktop “Result Sets” and into relational databases including Microsoft Access, Microsoft SQL Server, and ODBC databases including Oracle. Data can also be exported to Microsoft Excel.

In general, data from the MobileDataforce form is mapped to a database using the following methodology:

1. Groups are mapped to database tables.
2. Questions are mapped to database “fields” or columns.
3. Responses can be configured to send a value other than that shown on the form. For example, a radio button option may be shown on the form with a label of “Yes” or “No,” but the database may receive a value of “1” for yes, and “0” for no.

The work of configuring these mappings happens during the design of the form in the MobileDataforce desktop tool using the properties of Groups, Questions and Responses. The following illustration shows a Group called “Item” in Designer. “Item” maps to a database table called “ITEMS_RECEIVED.”



When you export your data using MobileDataforce Desktop, the Data Connection tools will generate a primary and foreign key field for each table that is exported. These fields are required, and are populated based on a numbering scheme that creates relationships between tables based on the hierarchical structure of the form. When exporting to Access, Excel, or Microsoft SQL Server, the tables and columns are generated automatically on the first export. Other databases require the correct database schema to be created ahead of time; the export process will not create them automatically.

Primary Key for each import

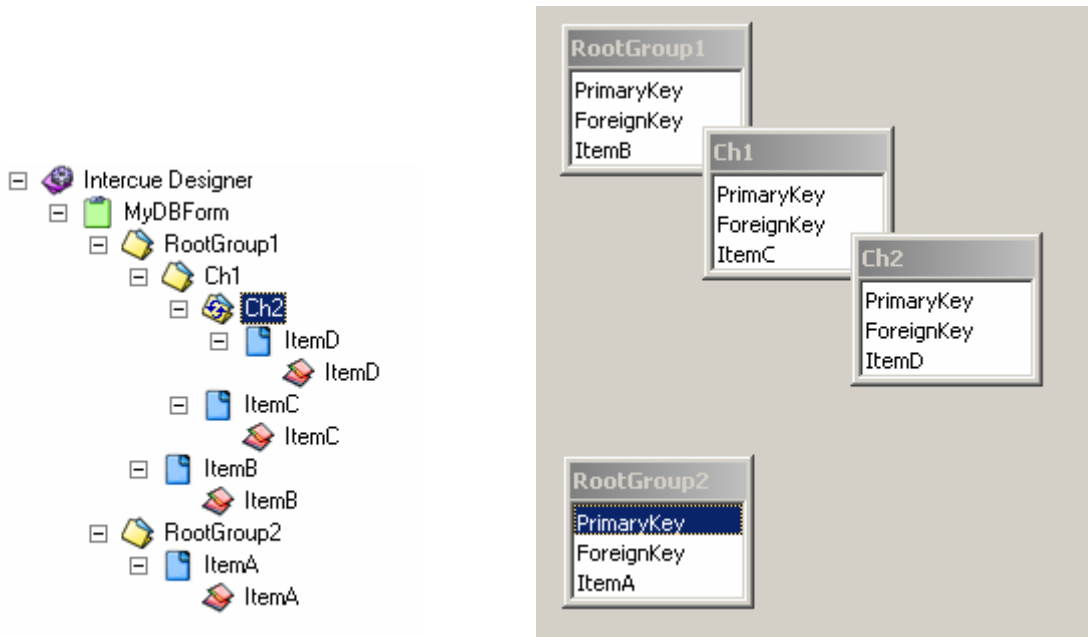
- * At the root level of the form, a unique integer is generated.
- * Each Result Set that is exported gets the next value (incremented by one).
- * This unique key is not saved as a primary key in any table.
- * Groups at the root level will have a foreign key matching this incremented Result Set value.

value.

So, each Result Set that is exported from Desktop and into an external database gets a primary key that is not listed in any table as a primary key, but is a unique number representing an import.

Relationships

Each child group gets a relationship via a foreign key field. The following 2 diagrams illustrate this. In the diagrams, there are two root level groups inside the form: RootGroup1, and RootGroup2. Rootgroup1 then has Child 1 (Ch1), which has Child 2 (Ch2) relationships. You can see here, that we are supporting one-to-many relationships using repeating groups (See Ch2).



Looking at the sample data, Ch2 was entered 5 times in the Ch1 entry for primary key of 4.00004:

The screenshot displays four data tables in a software interface:

- RootGroup1 : Table**

PrimaryKey	ForeignKey	ItemB
1.00002	1	651561
2.00002	2	132566
3.00002	3	984878
4.00002	4	654165
- Ch1 : Table**

PrimaryKey	ForeignKey	ItemC
1.00004	1.00002	Cradle
2.00004	2.00002	Device
3.00004	3.00002	Flash Card
4.00004	4.00002	Stylus
- Ch2 : Table**

PrimaryKey	ForeignKey	ItemD
1.00006	1.00004	10
2.00006	2.00004	25
3.00006	3.00004	19
4.00006	4.00004	62
4.00007	4.00004	46
4.00008	4.00004	100
4.00009	4.00004	5
4.0001	4.00004	35
- RootGroup2 : Table**

PrimaryKey	ForeignKey	ItemA
1.00002	1	3/1/2004
2.00002	2	3/3/2004
3.00002	3	3/5/2004
4.00002	4	3/8/2004

Red circles highlight the primary key '4.00002' in RootGroup1, '4.00004' in Ch1, and '4.00002' in Ch1. Red lines connect these to the corresponding foreign key '4.00004' in Ch2. A red box highlights the entire Ch2 table.

Many-to-many relationships are supported if you have two repeating groups in a single form. So, for example, you could collect all the Notes *and* Contacts for a single Item by having two repeating groups (1 for Notes and 1 for contacts) in a single Item group.

These database features are somewhat rudimentary, and provide functionality sufficient for basic data collection requirements. Our new product, to be released at the end of Q3 (September 2004), will provide full relational database synchronization technology, and our engineering team has made a great deal of headway in this area. Please contact us with specific requirements to see if we can provide additional functionality for your customers.

What is 2-way Data Transfer?

The MobileDataforce Intercue Mobility Suite 4.5 feature described as 2-way data transfer, has more to do with Result Sets than database information. It is two way data transfer from Desktop because if you have a previous XML "Result Set" that you collected information on in the past, you can send that data back to the device for review or correction. This is done from the Desktop by right-clicking on a Result Set file, and choosing Load to [Device] from the context menu. Customers have asked for this feature, and it was released in the 4.5 version.

Two way data synchronization goes deeper, and will be available in our new product due in the September 2004 timeframe. This is where a database is available on the device, and data is updated to this local (device) database. "Lookups" are also dynamic using this database synchronization. Full two-way synchronization of the complete database model from enterprise database to device and back will be supported. Database snapshots on the device are used to minimize the storage of the data on the device to what is needed by the user and according to the constraints of the device. We have already made great headway on designing and creating this solution.

In MDF Mobility Suite 4.5, updating lists in the form... We can provide this as a custom solution in a short timeframe for a specific customer. Also you can use XML Stylesheet Transformations and database mapping to generate a new form with the updated list of options. To do this with the current product could use one of several options. One option is outlined below:

1. Create your form in designer.

2. Using an XML tool that supports it, generate a schema based on the form structure. XML Spy does this.
3. Using the XML Schema for your form, use a tool such as MapForce, also by Altova, to regenerate the identical form with a new list of options.
4. Load the new form to the device on the next ActiveSync connection. The user's form is automatically up to date.

How do I print from MobileDataforce?

Printing is not addressed in version 4.5. One Solution Provider used the following solution for printing when the result set is saved:

We've built a very simple .Net CF application that loads and parses the XML result set for our shipping docket form, extracts the values for a sub-set of the form elements and then generates an RTF file with a little bit of formatting wrapped around the values and sends it to the printer. It's a very basic program and provided the customer doesn't modify the structure of the form too much then it should be OK.

Even if they do change it then it's only a five minute job to update the IDs in the printing app, and it's only viewed as a stop-gap measure until something better comes along. It's a big benefit of using XML, it makes it so much easier and more flexible for customers to work with.