

ADO/XML Reports Overview

MISys strongly recommends that, if you want to development a custom report using the XML/ADO Interface, you have significant experience with designing reports using Crystal Reports Designer and a working knowledge of XML and ADO, or you have MISys develop one for you (your MISys Customer Service representative can help you get a quotation for one) or you hire an outside expert to develop it for you.

There are many advantages in using the XML/ADO Interface. No SQL query filtering is required; the ADO records are already prepared by the MISys Manufacturing business logic before being sent to the RPT file. Columns can be formatted by MISys Manufacturing (e.g. Item formats, numeric formats), additional computed tables and columns not actually present in the database can be inserted, and specially handled column data such as MISys Manufacturing custom fields can be represented as standard columns in the passed ADO dataset.

The schema for the data is created by a tool in MISys Manufacturing and must be imported into the RPT file if the required data tables or columns need to change.

Custom reports using the MISys Manufacturing XML/ADO Interface use a special second definition file in addition to the Crystal Reports RPT file. This file is named 'CustomReportsENG.xml'; it is also known as the MISys Manufacturing Custom Report Definition file. There is a single Report Definition File that controls all custom reports. If any custom reports are created using the MISys Manufacturing Custom Reports Interface tool, or via the Report Import function, this file will be automatically created/maintained in the <MISys Manufacturing client install directory>\reports\custom' directory.

The Custom Reports Configuration File is an ISO standard XML file and can be viewed with any editor. If you are only interested in looking at the content of the file, you can just open the file in Microsoft Internet Explorer which will display formatted XML automatically.

The report definition file configures the following properties for each report:

1. How the report is accessed via the MISys Manufacturing UI.
2. Which RPT file is used to display the report data.
3. The definition of the UI parameters presented to the user within MISys Manufacturing
4. to specify the report criteria. Note these can use standard MISys Manufacturing finders and masked data input.
5. The definition of the tables used and any special column formatting.
6. The specification of MISys Manufacturing business logic process calls used for any required business logic processing. An example is the business logic process provide data for the exploded view of an MISys Manufacturing Bill of Material.
7. The SQL 'where' clause of the SQL filter expression used to filter the returned records.
8. Any required parameters to pass to the RPT file.

The MISys Manufacturing Custom Report Interface Tool provides a user friendly way to define the first three types of properties; how the reports is accesses through the MISys Manufacturing reports UI, which RPT file is used for the report, and the definitions for UI used to define the report criteria. The remaining properties must be defined by modifying the XML directly.

The Custom Report Interface Tool is designed to help manage the specific portion of the CustomReportsENG.xml file associated with the report interface that is being customized. Only that report's XML section is visible and editable.

Recommendations for Successful Report Customization

- If all you need to do is modify an existing report to access custom fields, you don't need to modify the XML at all. You just need to make sure the custom fields are defined in MISys Manufacturing and then export the current schema from MISys and then import into Crystal Reports. MISys Manufacturing automatically creates the ADO dataset with all custom field inserted and looking like normal data columns in the tables. Note that MISys Manufacturing stores custom fields in a way that cannot be easily extracted if you just read the database table directly without the ADO interface.
- We highly recommend using the Custom Report Interface wizard to look at an existing report you wish to customize and see how the report uses the XML as described in the reference section of this document.
- Use the **Print Data Tables** function in MISys Manufacturing to get the schema information for the tables you are using or want to add. This can be found in the Administration Panel under Housekeeping.
- You should not need to modify the Process attribute, or understand how the business processing logic works, for almost all customizations of existing reports. Most customizations only require addition of other, non-processed tables, or change to layout usage of the existing tables returned from the Processing tables.

Report Configuration XML Reference

The following information details the XML tags & elements required to understand the existing reports and to customize it. This information applies to the subsection of the XML displayed and editable in the XML tab of the Custom Reports AD)/XML Interface wizard available from within MISys Manufacturing if you have the Custom Report license installed. There are additional tags & elements used for custom reports if you examine the CustReportsENG.xml file produced by the Custom Report Interface wizard, but these are automatically managed by the wizard. Most reports provided by MISys require 10 or less lines of XML to define the required tables and data to pass to the RPT file.

The XML structure for the XML tab is as follows:

```

<extra name=, value= />
<table name=, formula= >
  <details>
    <detail name= />
  </details>
  <processparams>
    <processparam name, type, value />
  </processparams>
</table>

```

Note that elements and attributes in Bold above are the only required information. All the other elements and attributes are optional although many standard reports use them.

<Extra>

The <Extra> element is used to specify parameters to be passed to reporting engine via the special EXTRAS table that is always included in the ADO Dataset passed to the RPT file. These are often used to pass some user entered value to specify a parameter for the RPT file to use for formatting or sorting.

Attributes:

Name	Required. Must be unique within report. Only used for internal id.
Value	Required. Each report extra value is entered into column named 'Val2' through 'Val20' in the order they are declared. Note that 'Val1' is reserved and always set with report company name of the database it's being run against. 'Val20' is also reserved and always set with the report title attribute. The value usually refers to the name of a user parameter (e.g. '%1'), but it can be a constant, if desired. Also note that fieldRange parameters replaced in an extra create 2 entries in 'val' columns per extra; first for from value and second for to value.
Source	Optional. If using indirect reference for the value to a regular parameter, and that parameter is of type field range, can use the source attribute to

return just the 'from' value or the 'to' value. If the attribute is omitted, will return the full range query 'BETWEEN (from, to)'.

<Table>

The <Table> element is used to define a table to be included in the ADO dataset passed to the Crystal Reports RPT file. At least one <Table> element must exist for a report. The Formula element (see below) is used to specify the filter for the table and can reference parameter values that the user enters from the UI widgets defined for the report.

Attributes:

Name	Required. Name of database header table (e.g. MIITEM) to be included in passed Dataset for this report.
Format	Optional. Formatted fields to be added to table. Value is vertical bar separated list of field specs; one per field to be formatted. Each field spec is field name optionally follow by semicolon and then format spec. If field specified is a string type; uses mask set for that field. If field is numeric type, field spec determines decimal precision and currency formatting. For numeric types, field specs can be '0' – '6' for fixed precision or 'tdec', 'mdec' or 'qdec' to indicated option defined precision and currency formats. If field type is datetime type, field spec can be either 'date' to format as just date or 'datetime' to show date and time. For each field name specified, column is added to table with column key as original field name concatenated with '_form' and column values are always strings. If format property omitted, no formatted columns are added. Example format property: format='itemId cLast;mdec qStk;qdec'
Lookups	Foreign key return fields (join fields) to be included in table. Can have multiple with each lookup field spec separated by a semicolon. Format for lookup is: {comma separated list of table keys}@{foreign table name}>{comma separated list of fields to return in foreign table} Example – to return item description and type for bill of material detail (from MIBOMD), lookup would be 'partId@MIITEM>descr,type'. Note that '>' character may have to be entered as '>,' depending on XML editor used. Note list of return fields can be replaced by single '*' character which will return all fields for that table. You can add custom fields as lookups for foreign keys by specifying the special 'fldXml' field as one of the semicolon separated or comma separated foreign key field references as per the following:

{comma separated list of table keys}@{foreign table name}>fldXml

This will return all custom fields for the specified table.

Formula	Optional. SQL Query Where clause used to fetch filtered data for table. Can contain place holders for named parameters. Should use '%' followed by parameter name so can be found in query string. If no formula specified, table will return all rows in database. You can also specify the specific 'from' or 'to' value from field range types, if needed, rather than the normal BETWEEN clause. To specify just the 'from' value, use '%' followed by the parameter name followed by the 'F' character. To specify just the 'to' value, use '%' followed by the parameter name followed by the 'T' character.
Process	Optional. If specified, indicates that special business process function should be run to derive resulting table rows. Value is to set process type to perform (only 0 supported at this time). Currently, only use should be for MIXBOM table to create computed MIEBX table results, MILOGH table to create filtered MILOGH table results based on MIPORCVR records), or MIXSHOPOP table to create ProcTbl0 for Shop Floor Material Status reports.
Progress	Optional. Used in conjunction with Process to show progress window. Property value determines which kind of window to show. 0 shows only progress bar, 1 shows only text progress indication, and 2 shows both. For use with MIXBOM or MIXSHOPOP tables should use 1.
Flatten	Optional. Special setting to facilitate flattening a header table and all its detail tables to optimize performance for Crystal reports. For each detail table, adds specified columns from header to detail table and then removes header table. Flatten property value can be either "*" to include all header columns in the details or a semi-colon separated list of specific header columns to include. Note that header lookup and format columns are automatically included and do not need to be explicitly listed in the Flatten value list. If no Flatten present, normal tables are produced.
Flush	Optional. Used in conjunction with Process to specify list of tables (comma separated) to be flushed (foreign key return values filled in) after the Process is performed.
Remove	Optional. Used in conjunction with Process to specify list of tables (comma separated) to be removed after the Process is performed and prior to sending to Crystal Reports.
Currencyfield	Optional. Used to define field that specifies which currency is used for columns flagged as in source currency.
procTables	Optional. Special setting to determine list (comma separated) of process tables to return. This is currently only supported for MIXSHOPOP table and will include ProcTbl0.

<Details>

The <Details> element takes no attributes. The <Details> element is an optional element - it only need be included to define detail (child tables) also required in addition to the header table. It is used to enclose one or more <Detail> elements to define detail (child) tables of the header table specified in the parent <Table> element.

<Detail>

The <Detail> element must be enclosed within a parent <Details></Details> element. It is used to define a detail (child) table of the header table specified in the parent <Table> element.

Attributes:

Name	Required. Name of database detail table (e.g. MIILOC) associated with root header table to be included in passed Dataset for this report. Note no formula is allowed, all details for filtered header rows are returned in dataset.
Format	Formatted fields to be added to table. Same rules and syntax apply as described for format property of table tag above.
Lookups	Foreign key return fields (join fields) to be included in table. Same rules and format apply as described for Table tag 'lookups' property above.
Process	Optional. If specified, indicates that special process function should be run to derive resulting table rows. Value is to set process type to perform. Currently, only use should be for MILOGD table to created computed MILOGDX table results and only valid value is 0.
Progress	Optional. Used in conjunction with Process to show progress window. Property value determines which kind of window to show. 0 shows only progress bar, 1 shows only text progress indication, and 2 shows both. For use with MIXBOM table should use 1.
Formula	Optional. SQL Query Where clause used to fetch filtered data for detail table. Can contain place holders for named parameters. Should use '%' followed by parameter name so can be found in query string. If no formula specified, table will return all detail rows in database for specified header(s). Filters for details must have all column names use unambiguous names – e.g. {tablename}.{columnname}.
Currencyfield	Optional . Used to define field that specifies which currency is used for columns flagged as in source currency.

<ProcessParams>

The <ProcessParams> element takes no attributes. The <ProcessParams> element is an optional element - it only need be included to define one or more <ProcessParam> elements which describes each process parameter to be passed to a superview processing function. Only used if the parent <Table> as a Process attribute defined for one of the tables that supports it.

<ProcessParam>

Used if the parent <Table> element has process attribute and is used to specify a column value to be set in a superview table PRIOR to processing.

Attributes:

Name	Required. Name of field in process table to set specified value to.
Type	Required. Defines table column data type so value is properly cast. Only 'string', 'integer', 'date' and 'boolean' are supported. Use section on Process Tables to determine correct format for field.
Value	Required. Value to be set in specified superview table field.

Standard Reports using Processing

Accounting/Subledger (MILOGH002)
 Bills of Material/Indented Bill of Material (MIBOMH002)
 Bills of Material/Exploded Bill of Material (MIBOMH004)
 Bills of Material/Aggregate Parts List (MIBOMH005)
 Bills of Material/Indented Pick List (MIBOMH006)
 Bills of Material/Aggregate Pick List (MIBOMH007)
 Items/Item Where Used (MIITEM012)
 Jobs/Job Purchasing Costs (MIJOBH004)
 Jobs/Job Production Costs (MIJOBH005)
 Manufacturing Orders/Indented Mfg. Order (MIMOH007)
 Manufacturing Orders/Aggregate Parts List (MIMOH008)
 Work Orders/Work Order (MIWOH004)
 Work Orders/Indented Pick List (MIWOH003)
 Work Orders/Aggregate Pick List (MIWOH005)
 Serial/Lot Tracking/SL Assembly Pick List - Top Level Only (SLT006)
 Shop Floor Operations/Work Center Dispatch List (MIMOWC004)
 Shop Floor Operations/Work Center Loads (MIMOWC005)
 Shop Floor Operations/Overloaded Work Centers (MIMOWC005A)
 Shop Floor Operations/Work Center Loads - by Date (MIMOWC006)
 Shop Floor Operations/Overloaded Work Centers - by Date (MIMOWC006A)
 Shop Floor Operations/Bill of Manufacturing (MIEBX001)
 Shop Floor Operations/Exploded Bill of Manufacturing (MIEBX003)
 Shop Floor Operations/Bill of Manufacturing Cost Breakdown (MIEBX005)
 Shop Floor Operations/Mfg. Order Bill of Manufacturing (MIEBX002)
 Shop Floor Operations/Mfg. Order Exploded Bill of Manufacturing (MIEBX004)
 Shop Floor Operations/Mfg. Order Material Status (MIMORD003)
 Shop Floor Operations/Costed Mfg. Order Material Status (MIMORD003A)
 Shop Floor Operations/Mfg. Order Bill of Manufacturing Cost Breakdown (MIEBX006)
 Shop Floor Operations/Mfg. Order Cost Comparison (MIMOH010)
 Shop Floor Operations/Mfg. Order Cost Comparison - by Job (MIMOH011)

Process Table Usage

Process Table	Results Table	Process Parameters
MIXBOM (BOM Explosion)	MIEBX (BOM Explode Work table)	
MIJOBH (Jobs)	MIJOBH (Jobs)	
MIXSHOPOP (Shop Floor Processing)	ProcTbl0	
MIXDSP (Routing Dispatch)	MISHOPDP (Shop Dispatch)	
MIXLOG (Expand Master Log)	MILOGDX (Expanded Master Log Details)	

BOM Explosion Superview (MIXBOM)

Field	Data Type	Size	Precision	Default	Values	Attributes
Superview dummy key (dummy)	string	12		''		Primary Key, KV
Mode (mode)	integer	2		0	0 - Bill of Material, 1 - Bill of Manufacture, 2 - Bill of Material (Composed), 3 - Bill of Manufacture (Composed)	
Process Type (type)	integer	2		0	0 - Single Level BOM, 1 - Exploded BOM, 2 - Where Used BOM, 3 - Recursive BOM Check, 4 - Replace BOM Components, 5 - Where Used Work Orders, 6 - Single Level MO, 7 - Exploded MO, 8 - Recursive MO Check, 9 - Partial MO Cost Roll Up, 10 - Full MO Cost Roll Up, 11 - Single Level WO, 12 - Exploded WO, 13 - Partial BOM Cost Roll Up, 14 - Full BOM Cost Roll Up	
Process Subtype (subType)	integer	2		0	0 - Replace, 1 - Add, 2 - Remove	
From Item (itemLo)	string	24		NULL		Nullable, Mask: >CCCCCCCCCCCCCCCCCCCC
To Item (itemHi)	string	24		'Z'		Nullable, Mask: >CCCCCCCCCCCCCCCCCCCC
From BOM Revision (revnLo)	string	6		NULL		Nullable, Mask: >CCCCCC
To BOM Reference (revnHi)	string	6		'Z'		Nullable, Mask: >CCCCCC
Location (locd)	string	6		NULL		Nullable, Mask: >CCCCCC
Preview (preview)	boolean	1		0		
Effective From Date (dateLo)	string	8		NULL		Nullable, D
Effective To Date (dateHi)	string	8		NULL		Nullable, D
Force New Revision (revnSw)	boolean	1		0		
New Revision From Date (sDate)	string	8		NULL		Nullable, D
New Revision To Date (eDate)	string	8		NULL		Nullable, D
Blank Revision Seed (seed)	string	6		''		Mask: >CCCCCC
Increment Mode (incMode)	integer	2		0	0 - Alpha (A-Z), 1 - Numeric (0-9), 2 - Both	
Create New Revision Only for Current	boolean	1		1		
From MO Number (mohldLo)	string	15		NULL		Nullable, Mask: >CCCCCCCCCCCCCCCC
To MO Number (mohldHi)	string	15		'Z'		Nullable, Mask: >CCCCCCCCCCCCCCCC
From Job Number (jobldLo)	string	12		NULL		Nullable, Mask: >CCCCCCCCCCCC
To Job Number (jobldHi)	string	12		'Z'		Nullable, Mask: >CCCCCCCCCCCC
From WO Number (wohldLo)	string	12		''		Mask: >CCCCCCCCCCCC
To WO Number (wohldHi)	string	12		'Z'		Mask: >CCCCCCCCCCCC
To WO Line Number (wodLnHi)	integer	2		0		
From WO Line Number (wodLnLo)	integer	2		0		
Component Item Number (oldPart)	string	24		NULL		Nullable, Mask: >CCCCCCCCCCCCCCCCCCCC
Replacement Item Number (newPart)	string	24		NULL		Nullable, Mask: >CCCCCCCCCCCCCCCCCCCC
Quantity (addQty)	number	22	6	0		
Return All Tables (bAllTables)	boolean	1		1		
Child MO Cost (bChildCost)	boolean	1		0		
Result (result)	integer	2		0	0 - Process Completed Successfully, 1 - Maximum BOM level overflow error, 2 - Other error	

Shop Floor Processing (MIXSHOPOP)

Refs: toolId@MITOOL; wclD@MIMOWC; xvarToolId@MTOOL; xvarWclD@MIMOWC

Field	Data Type	Size	Precision	Default	Values	Attributes
Superview dummy key (dummy)	string	12		''		Primary Key, KV
Process Type (type)	integer	2		0	0 - Substitute Tool, 1 - Substitute Work Center, 2 - Update Route, 3 - Post Shop Op Batch, 4 - Material Status Info	
From MO No. (mohLo)	string	15		''		Mask: >CCCCCCCCCCCCCCCC
To MO No. (mohHi)	string	15		''		Mask: >CCCCCCCCCCCCCCCC
From BOM No. (bomLo)	string	24		''		Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
To BOM No. (bomHi)	string	24		''		Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
From Revision (revLo)	string	6		''		Mask: >CCCCCC
To Revision (revHi)	string	6		''		Mask: >CCCCCC
Substitute Tool No. (toolId)	string	12		''		Refs MITOOL, Nullable, FV Mask: >CCCCCCCCCCCC
Substitute Work Center (wclD)	string	24		''		Refs MIMOWC, Nullable, FV Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
Include BOMs (includeBOM)	boolean	1		0		
Include Mfg. Orders (includeMO)	boolean	1		0		
With Tool No. (xvarToolId)	string	12		NULL		Refs MITOOL, Nullable, FV Mask: >CCCCCCCCCCCC
With Work center (xvarWclD)	string	24		NULL		Refs MIMOWC, Nullable, FV Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
Use BOM List (useBOMList)	boolean	1		0		
Use Mfg. Order List (useMOList)	boolean	1		0		
BOM Run Time (bomCycleTime)	boolean	1		0		
BOM Setup Time (bomSetupTime)	boolean	1		0		
BOM Queue Time (bomPreopTime)	boolean	1		0		
BOM Wait Time (bomPostopTime)	boolean	1		0		
MO Run Time (moCycleTime)	boolean	1		0		
MO Setup Time (moSetupTime)	boolean	1		0		
MO Queue Time (moPreopTime)	boolean	1		0		
MO Wait Time (moPostopTime)	boolean	1		0		
Batch User ID (btUserId)	string	8		NULL		Nullable, Mask: >CCCCCCCC
Batch Number (btBatchId)	string	6		NULL		Nullable, Mask: >#999999
Result (result)	integer	2		0	0 - Completed successfully, 1 - Failed	
Log Entry (entry)	integer	4		0		
Bin Assignment Count (binCnt)	integer	2		0		
SLT Assignment Count (sltCnt)	integer	2		0		

Routing Dispatch Superview (MIXDSP)

Refs: classId@MIMOWCC

Field	Data Type	Size	Precision	Default	Values	Attributes
Superview dummy key (dummy)	string	12		''		Primary Key, KV
Process Type (type)	integer	2		0	0 - BOM Dispatch, 1 - MO Dispatch, 2 - BOM GanttChart, 3 - MO GanttChart, 4 - Work Center Daily Dispatch, 5 - Work Center Overload, 6 - Set MO Date, 7 - Work Center Gantt, 8 - Plan Work Center Gantt, 9 - Plan Work Center Substitution List, 10 - Plan Work Center Load, 11 - Plan Reschedule Order, 12 - Plan Update Orders	
Inverse Direction (inverseDir)	boolean	1		0		
Include Optional Shifts (inclOptShift)	boolean	1		0		
Overlap Operations (overlap)	boolean	1		1		
From Date (dateLo)	string	23		NULL		Nullable, D
To Date (dateHi)	string	23		NULL		Nullable, D
MO Number (mohId)	string	15		NULL		Nullable, V Mask: >CCCCCCCCCCCCCCCC
WO Number (wohId)	string	15		NULL		Nullable, V Mask: >CCCCCCCCCCCCCCCC
WO Detail (wohId)	integer	4		0		
Operation Number (opCode)	string	24		''		Nullable, Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
From Work Center (wclDLo)	string	24		''		Nullable, V Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
To Work Center (wclDHi)	string	24		'Z'		Nullable, V Mask: >CCCCCCCCCCCCCCCCCCCCCCCC
Order Quantity (ordQty)	number	22	6	0		
Selection Mode (selMode)	integer	2		0	0 - Range, 1 - List, 2 - Class	
Work Center Class (classId)	string	12		NULL		Refs MIMOWCC, Nullable, FV Mask: >CCCCCCCCCCCC
Operation Start DateTime (altStartDt)	string	23		NULL		Nullable, D
Replacement Work Center (altWclD)	string	24		''		Nullable,
Late Ship Alert (lateShipAlert)	boolean	1		''		
Work Center List (wclList)	integer	0		''		
Result (result)	integer	2		0	0 - Process Completed Successfully, 1 - Process Failed	

Expand Master Log Detail (MIXLOG)

Field	Data Type	Size	Precision	Default	Values	Attributes
Superview dummy key (dummy)	string	12		''		Primary Key, KV
Journal Number (jmlNo)	integer	4		0		
Type (type)	integer	2		0	0 - Full Detail by Account, 1 - Full Detail by Date, 2 - Summary by Account, 3 - Summary by Date	
Process Results (result)	integer	2		0	0 - Process Completed Successfully, 1 - Process Failed	

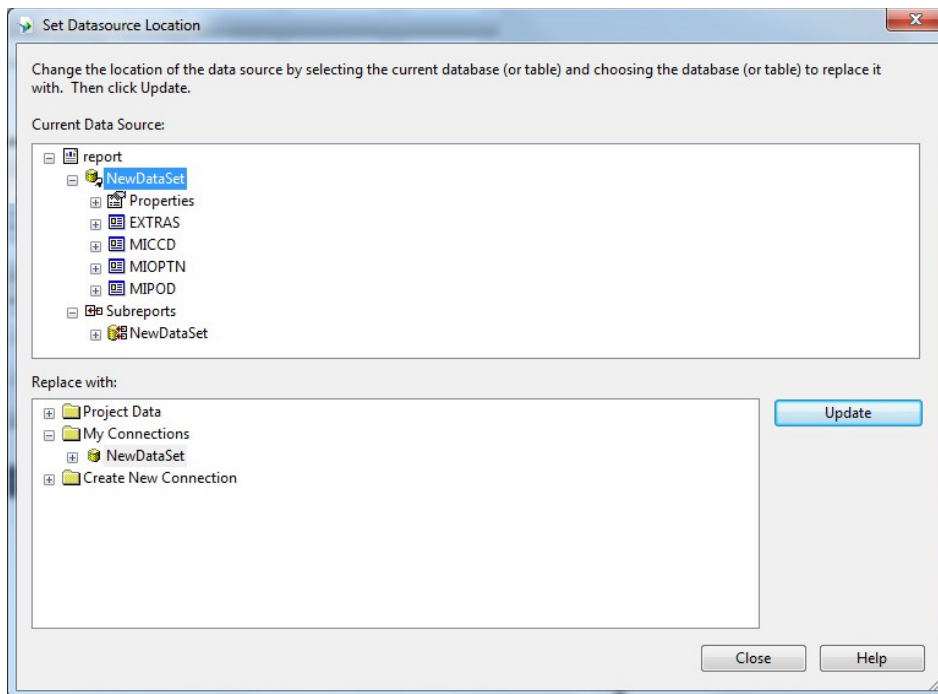
ProcTbl0 Schema when returned from MIXSHOPOP

Field	Data Type	Size	Precision	Default	Values
mohId	string	15			
mohDescr	string	60			
buildItem	string	24			
buildItemDescr	string	60			
jobId	string	12			
jobName	string	60			
locId	string	6			
locName	string	60			
ordQty	Number	22	6		
Priority	integer	2			
moStat	integer	2			
lineNbr	integer	2			
opCode	string	24			
wcId	string	24			
wcDescr	string	60			
startQty	Number	22	6		
compQty	Number	22	6		
batchSize	Number	22	6		
toolId	string	12			
opCmnt	string	60			
showDetail	Integer	2			
initReq	boolean	1			

Importing ADO Schemas into a Crystal Reports RPT file

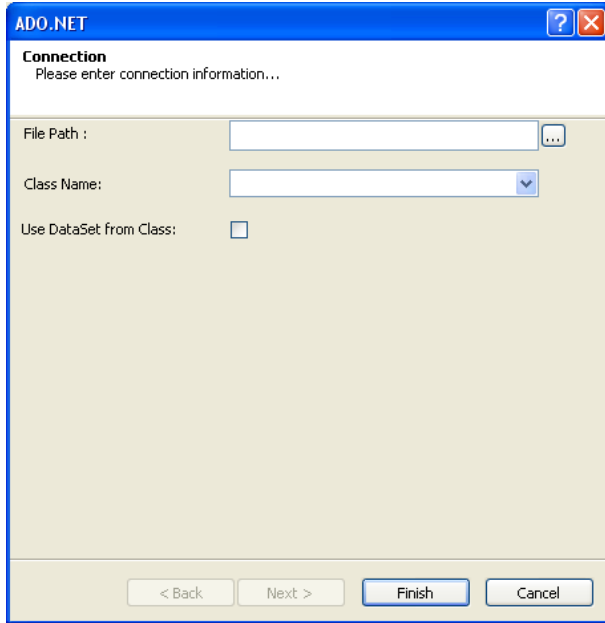
The following information applies to use of the Crystal Report Designer application. You may want to become familiar with Crystal Report's own documentation on importing schemas, but the following illustrates how it is usually done.

If you are editing a copy of a MISys Manufacturing standard report RPT file, or if you need to make changes in the schema after you have already customized the RPT file, you can update the schema by selecting the 'Set Datasource Location...' menu item in the Crystal Reports application Database menu. The following dialog will be displayed:

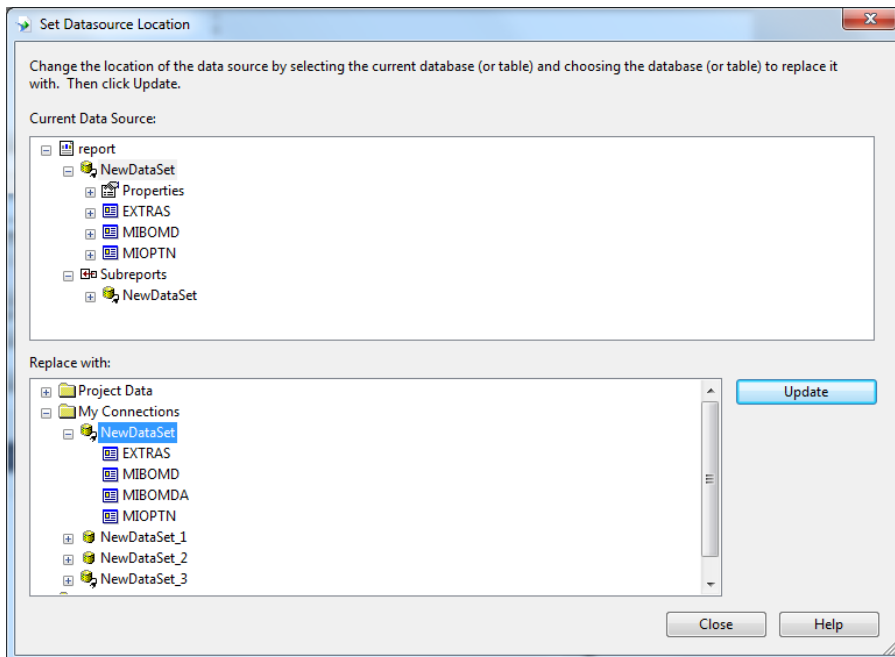


Note that depending on which version of the Crystal Reports Designer application you are using, and which RPT file you are importing the schema into, the screens may vary slightly from what is shown above.

Click the + icon to the left of 'NewDataSet' in the Replace With tree at the bottom. This will display the following form which will allow you to select the schema file to import and replace the current RPT schema with:



You should leave the 'Use Dataset from Class' checkbox unchecked and the Class Name combo empty. Use the browse button to find the schema file you created from the MISys Manufacturing Custom Report Interface tool. Schema files, including those exported from MISys Manufacturing, normally have an .XSD file extension. After selecting the schema file, clicking Finish will cause the schema to be imported, the Set Datasource Location dialog to be updated, and you should see the 'NewDataSet' in the Replace With section expanded with a list of tables from the imported schema.



Make sure 'NewDataSet' is still selected in the Current Data Source Section and 'NewDataSet' is selected in the Replace With section and then click the Update button. This causes the report's current schema to be replaced with the just imported schema.

After the Schema Import is completed, Crystal Reports will show the imported table and fields in the standard field explorer view and you can place them on the report layout or refer to them in formulas using standard Crystal Report functionality.

NOTE: you do not need to connect to any database directly in Crystal Reports when importing a schema, MISys Manufacturing will supply the data for the Dataset when the report is run from MISys Manufacturing.

NOTE: You must export the schema from MISys Manufacturing and re-import into the Crystal Reports RPT file if you make any changes to the set of tables or fields in the Configuration XML for the report interface (e.g. adding references to custom fields) or you are using a new or different RPT file.