



# Softshare Delta<sup>®</sup>

The open architecture of Softshare Delta lets you create just about any integration solution imaginable. Whether you want to integrate EDI purchase orders with an internal accounting database, convert accounts receivable files into XML invoices, load your inventory into an online Web catalog, or handle Web service requests, Delta has all the tools you need to integrate your business operations.

As a universal data translator, Delta not only supports EDI standards (X12, EDIFACT and TRADACOMS), but a variety of other data formats as well, including data (flat) files, database tables (via OLE DB), and XML. Delta also supports mapping to free-form text formats such as HTML to aid in Web integration.

Because Delta is an any-to-any mapper, not only are you able to integrate all incoming and outgoing e-commerce documents with your internal applications, but you are also able to integrate between your internal applications. And when Delta is paired with Softshare ECS, Softshare's communications server, you have an enterprise application integration (EAI) solution capable of coordinating the many diverse applications, databases, and e-commerce formats found across your enterprise.

select a tab to continue >>>

# Delta Components & Editions

Softshare offers three editions of Softshare Delta: Delta Standard, Delta Enterprise, and Delta Run-Time. The standard and enterprise editions are comprised of two individually-licensed components: Delta Studio and Delta Engine.

## Delta Editions

**Delta Standard** provides complete, integrated support for creating, maintaining, and running translation maps for all of Delta's supported source and target data formats.

**Delta Enterprise** includes all features of Delta Standard, plus the Solution Manager, which allows for centralized management and deployment of Delta resources. In addition, Delta Enterprise includes support for W3C-/IETF-published XML security standards.

**Delta Run-Time** incorporates the same powerful map execution capabilities as Delta Standard and Enterprise, but without any user interface components. Because Delta Run-Time cannot be used to create or edit translation maps, it's typically used by independent software vendors looking to bundle Delta as part of another software solution for the purpose of offering out-of-the-box translation to clients.

## Delta Components

**Delta Studio** is the design component of Delta. It's from this interface that developers access, configure, manage, administer, and develop all data translation maps.

**Delta Engine** is required to execute the maps developed in Delta Studio. As such, Delta Engine is typically installed on the server, whereas Delta Studio typically resides on a developer's workstation.

One Delta Studio and one Delta Engine license are included with the purchase of Delta Standard or Delta Enterprise. Additional studio or engine licenses are available for purchase as needed.

Source Data Format	Target Data Format				
	EDI	Data File	Database	XML	Text
EDI to...	✓	✓	✓	✓	✓
Data File to...	✓	✓	✓	✓	✓
Database to...	✓	✓	✓	✓	✓
XML to...	✓	✓	✓	✓	✓

Data Formats Supported By All Delta Editions

# Delta System Requirements

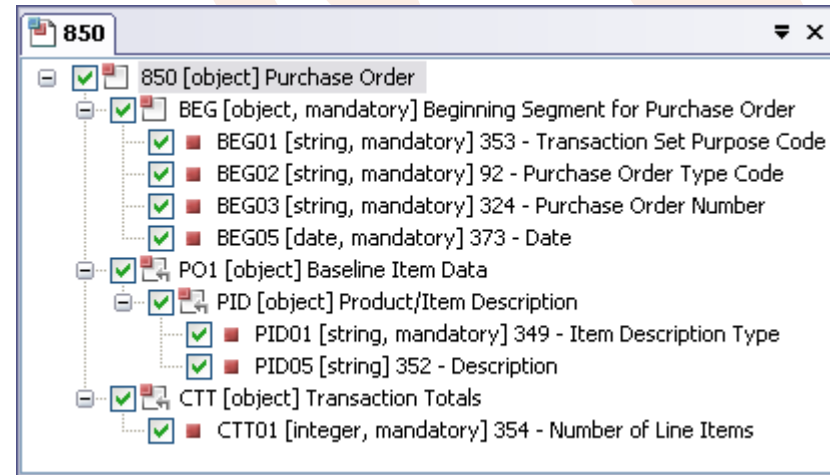
Component	Minimum Requirement
Computer	Intel Pentium-compatible CPU with 1 GHz or higher processor
Memory	1 GB of RAM (2 GB recommended)
Hard Disk Space	100 MB available disk space
Monitor	VGA monitor with a 256-color display
Operating System	Microsoft Windows® XP Professional, Windows 2003, Windows Vista, or Windows Server 2008

# Easy Modeling of Five Data Formats

Before you can map between two data formats, you must first model those formats. Your source model represents the data that you are translating and your target model represents the data after it is translated. For example, if you were to integrate incoming EDI purchase orders into an accounting database, your source model would represent the format of the EDI purchase orders and your target model would represent the format of your accounting database.

From Softshare Delta, you can design models from scratch or you can import sample files or electronic specifications and let Delta design them for you. Either way, Delta's helpful modeler screens efficiently guide you through the process, prompting you for important information about your model and the unique attributes of the data format you are modeling.

Delta displays your models in a user-friendly interface. All objects and elements are given descriptive titles and parent-child relationships are graphically shown via nesting. In addition, each element has its own set of properties that let you control the element's mandatory status, width, range, acceptable values, and so on.



Model Display

		Model Creation Method		
		From Scratch	Import Sample	Import Electronic Specifications
Data Format	EDI	Using Delta's built-in EDI standards	Any EDI document	SEF, gXML, or model from Softshare library
	Data File	Data file modeler	Any fixed-width or character-delimited file	SAP IDoc* or Oracle Interface File*
	Database	Custom SQL query	Automated modeling via OLE DB access	
	XML	XML modeler	Any XML document	XML Schema, DTD, or Oracle .XGD File*
	Text	Text modeler	Not available	

\*With the purchase of the add-on Softshare Integration Bridge for SAP or Oracle

Overview | EDI | Data File | XML | Database | Text Document

select a data format to learn more about Delta's modeling features

Easy Modeling

Mapping

Map Testing

Solution Manager

Trading Partner Management

# EDI Modeling

Softshare Delta ships with the final releases of the ANSI X12 and UN/EDIFACT standards and offers full support for all X12 dictionaries and all recent EDIFACT directories. In addition, Delta supports the TRADACOMS EDI standard.

## EDI Modeling with Ease

There are five easy ways to model your EDI documents:

- Import sample EDI documents
- Import pre-built models from the Softshare library
- Import Standard Exchange Format (SEF) files
- Import Guideline XML (gXML) files
- Use Delta's built-in EDI standards

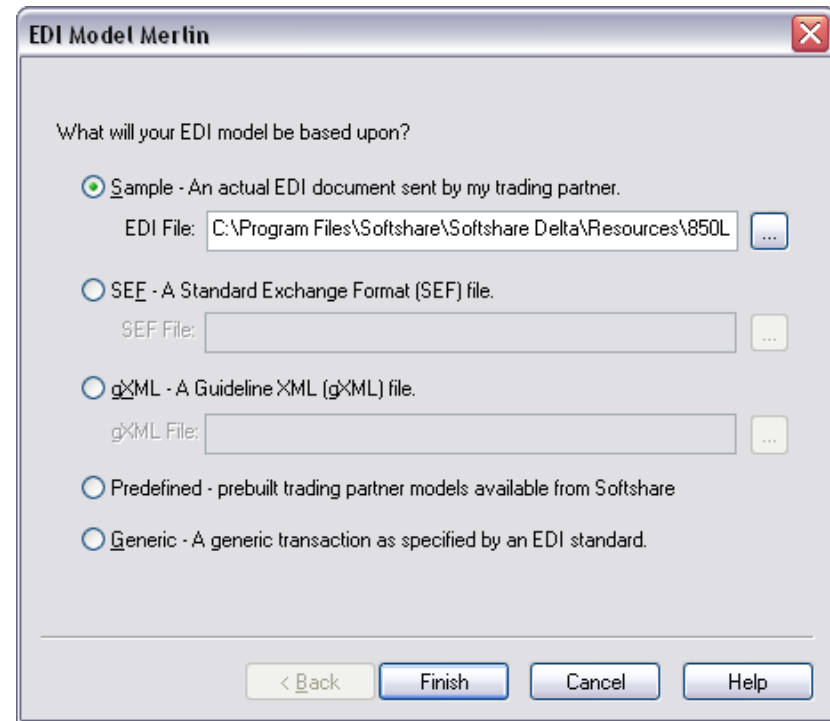
When you import a sample EDI document, Delta creates a baseline model that reflects all the segments and elements found in the sample. This baseline model can then be enhanced with additional segments, elements, and code values as needed.

If your trading partner makes their ICs (implementation conventions) available in an electronic SEF or gXML file format, Delta can use this information to create a complete EDI model in a matter of seconds, no adjustments needed.

Even creating an EDI document from scratch is easy. Using its built-in EDI standards, Delta provides you with a baseline EDI model that contains the mandatory segments and elements required for the EDI version and document type you are modeling. If your trading partner's ICs call for "illegal" EDI, you can create and save a custom EDI version to use as a basis for your EDI model.

## Migrating to New EDI Versions

If your ICs or those of your trading partners are updated to reflect a new EDI version, Softshare Delta can automatically convert your models to the new version. Called map migration, this feature saves you the time and trouble of recreating your EDI models and the maps associated with them.



# Data File Modeling

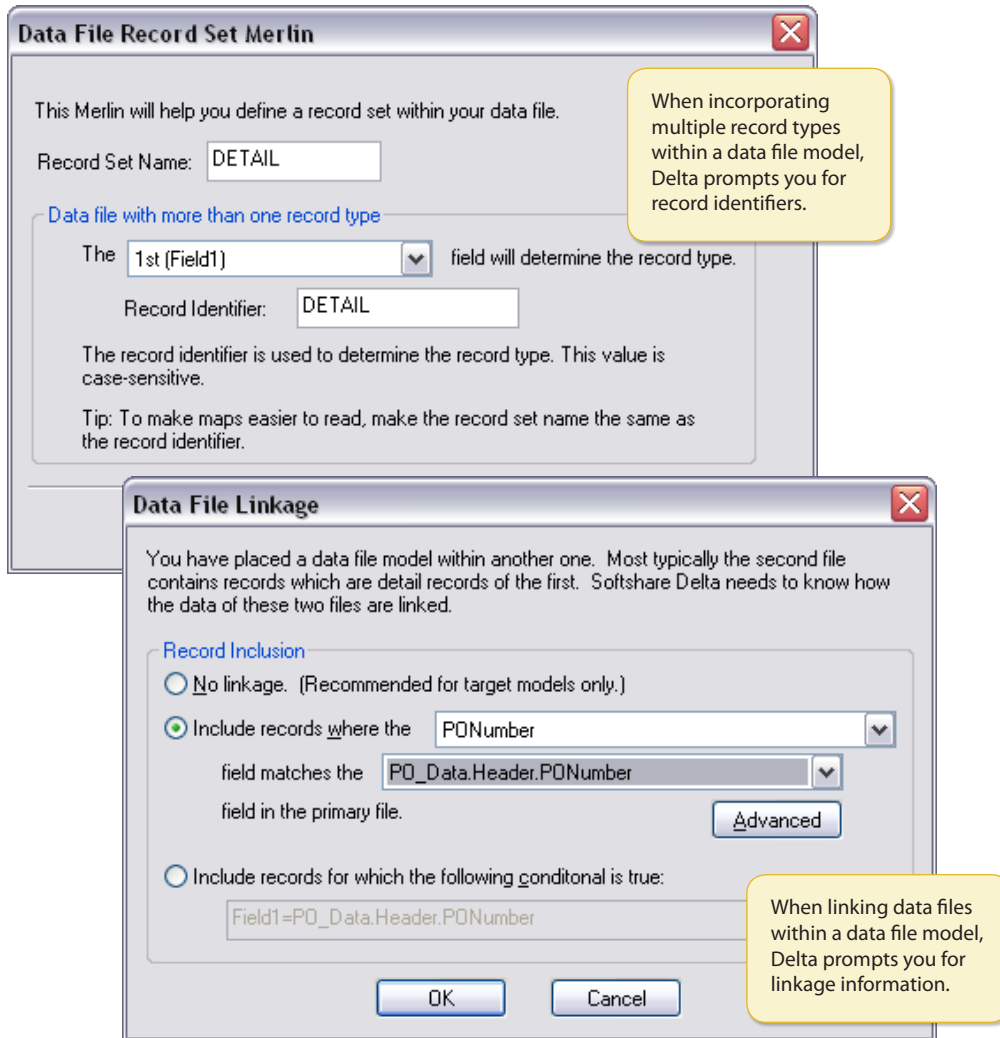
Like EDI models, data file models are easily created by importing sample files. Softshare Delta recognizes both fixed-width and character-delimited file formats, including comma-separated values (CSV) and tab-delimited files.

## Hierarchical & Structured Data Files

Softshare's data file models can consist of multiple, linked data files (also known as hierarchical data files) or multiple record types (also known as structured data files). This is crucial for support of the header-detail relationships often found between data files.

When linking data files within a data file model, Delta prompts you for linkage information. Typically, linkage is accomplished through matching field values. Then, during mapping, Delta uses the linkage criteria to read in or write out all of a header record's detail records before moving on to the next header record.

When incorporating multiple record types within a data file model, Delta prompts you for record identifiers so that it can distinguish record types during mapping.



# XML Modeling

There are four easy ways to model your XML documents:

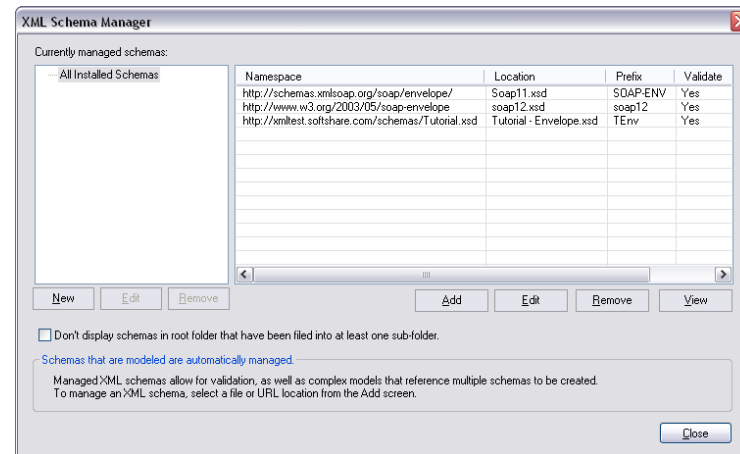
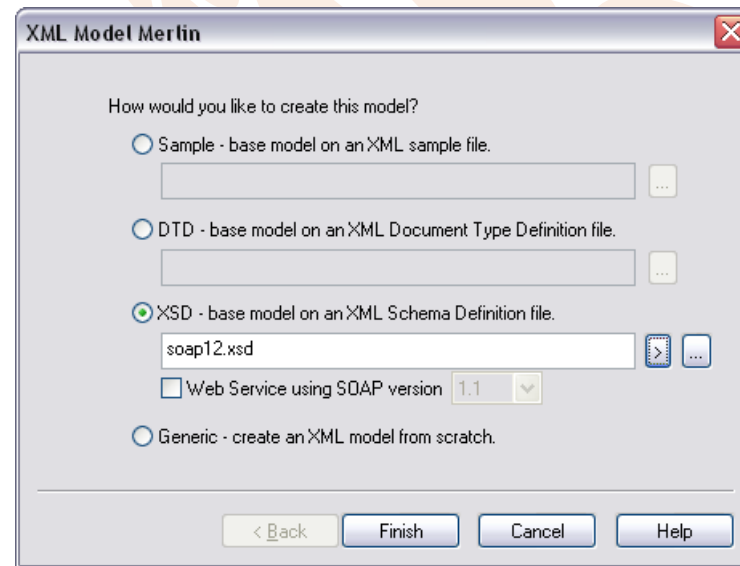
- Import sample XML documents
- Import XML schemas
- Import DTDs
- Use Delta's XML modeler to create an XML model from scratch

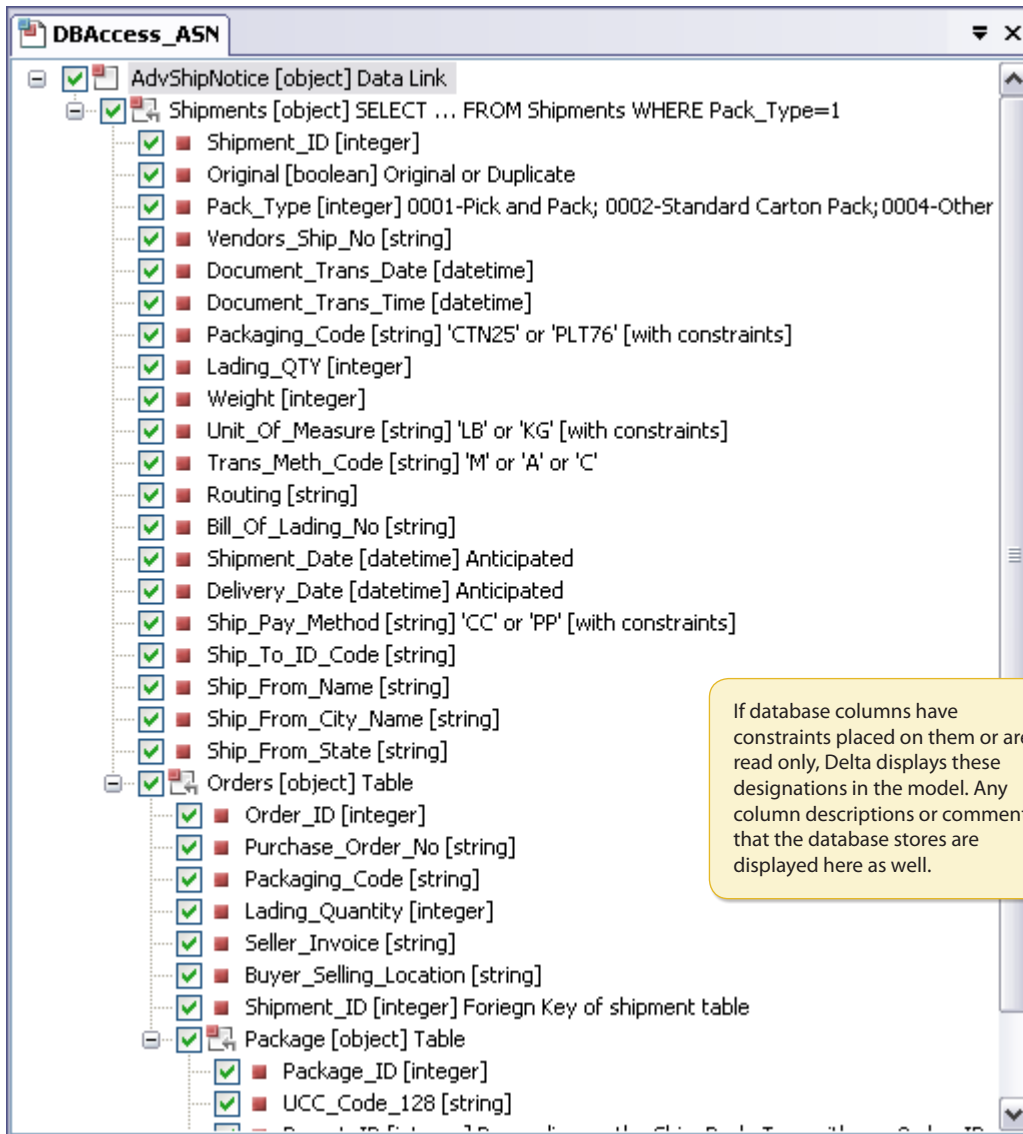
## Web Services

Softshare Delta, when used in conjunction with Softshare ECS, provides a complete solution for the exchange, creation, and translation of XML-based business frameworks such as RosettaNet, SOAP, BizTalk, and ebXML. In other words, Delta's support of XML opens up the world of Web services to you.

One feature that is integral to Delta's support of XML-based business frameworks is the XML Schema Manager. The XML Schema Manager locally stores the schemas you work with so that, during mapping, Delta can dynamically validate XML data against its associated schema.

Another feature integral to Delta's support of XML-based business frameworks and Web services is the ability to overlay XML models with identity models. An identity model identifies the location of key elements in an XML business document such as the document's sender, receiver, and control number. This gives business context to XML data so that (1) Delta can recognize the sender/receiver of incoming XML data and trigger the appropriate maps and (2) ECS can track XML data as it passes through your enterprise.





## Database Modeling

Softshare Delta supports Microsoft's OLE DB standard for universal data access, allowing you to avoid intermediary data files and integrate directly with your internal database applications. Since OLE DB is backward compatible with the ODBC standard, Delta can access hundreds of relational database systems, including SQL Server, Oracle, IBM's DB2, and Informix, as well as file-based databases such as Microsoft Access, dBASE, and FoxPro.

Modeling databases is easy in Delta. To connect to and model a database, you create a data link in the Data Link Manager utility. Data links identify database location, user name and password, and other relevant database information.

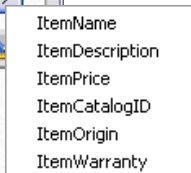
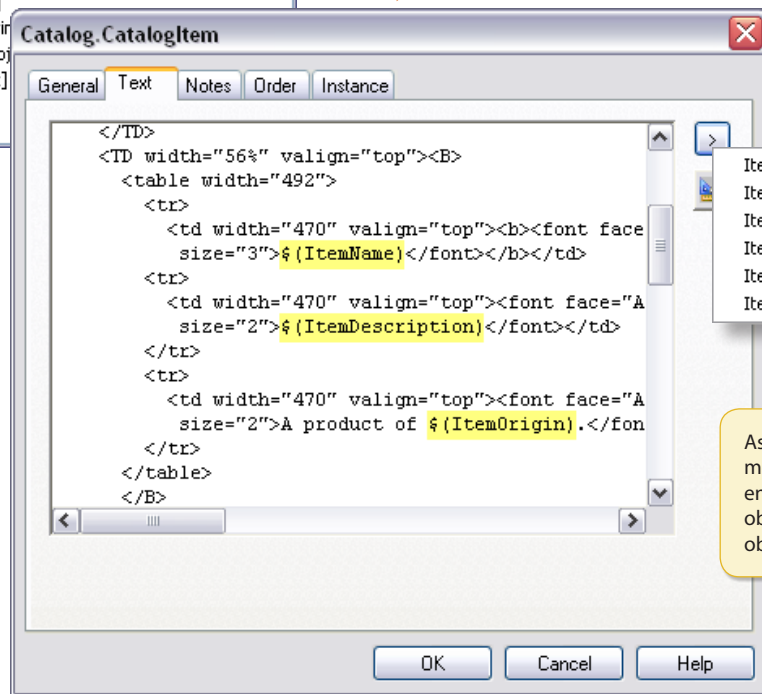
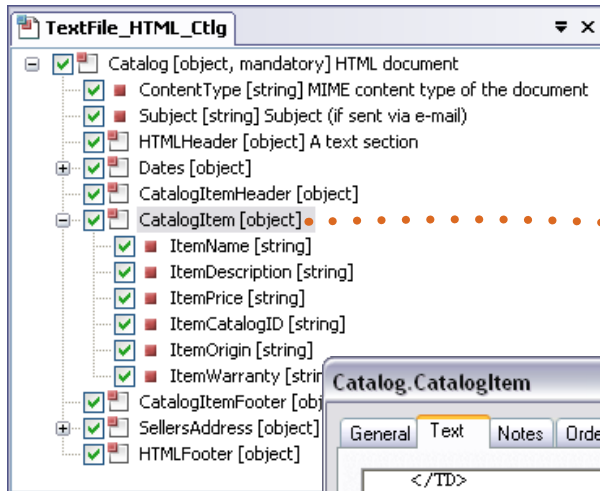
As with data file models, database models can consist of multiple, linked tables. This is crucial for support of the header-detail relationships typically found between database tables. A single database model can incorporate tables that do not reside in the same database, or even on the same machine.

Affording you more power, Delta lets you write your own structured query language (SQL) statements for running complicated queries or calling stored procedures.

# Text Document Modeling

For use as a target model only, text document models allow you to translate any of the previously-highlighted model formats into text files. For example, you can use this model type to create faxes, e-mails, form letters, and even HTML-formatted Web pages.

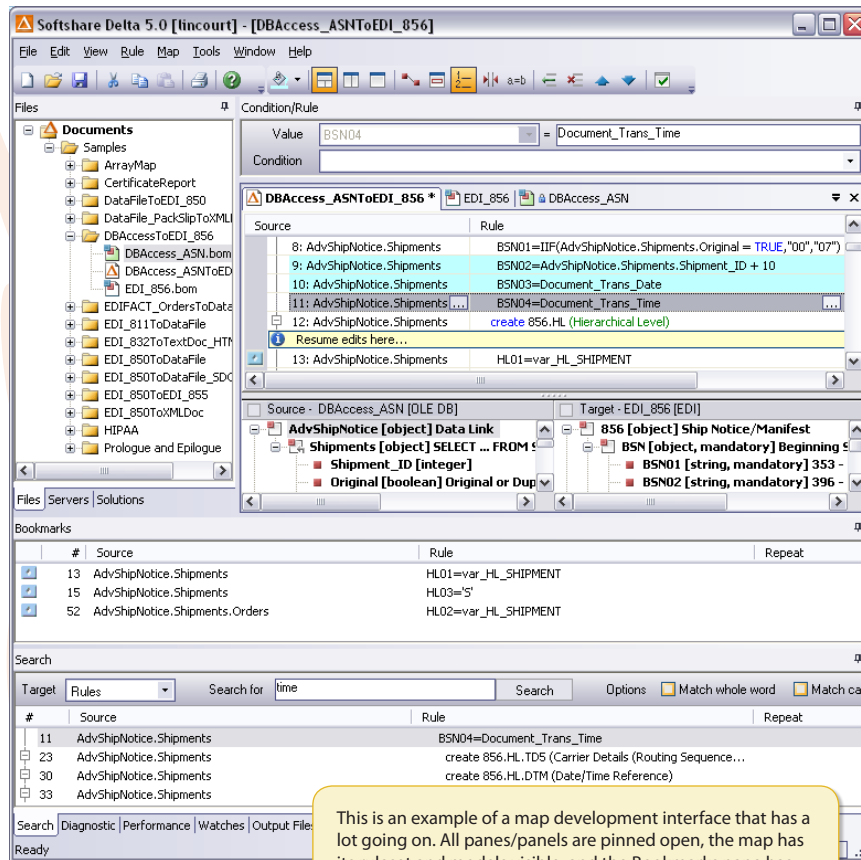
Softshare Delta's text document models are unique in that the model elements are referenced within the text sections that you author. During mapping, these elements are substituted for the data that is mapped to them, letting you populate the text output with information that is specific to the source data.



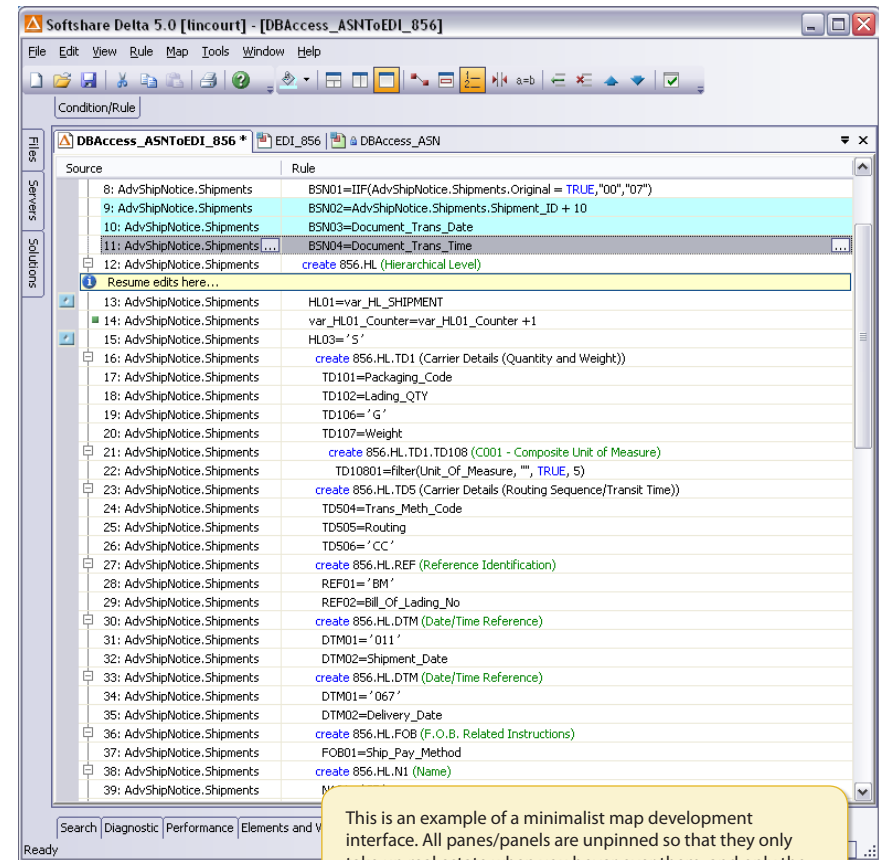
As shown in this HTML model example, the text entered for a text section object references the object's child elements.

# Flexible Interface

From the map development window, all of Softshare Delta's diagnostic and debugging tools are at your fingertips, along with a number of other task-oriented panes. These panes can be grouped with other panes to create a panel or singled out for maximum access. Panes (and panels) can be docked to any area of the interface—or undocked so that they float over the interface. In addition to a flexible workspace, Softshare Delta also offers a variety of map-viewing modes and rule-viewing options on a per-map basis to help you create the ideal work environment for map design and navigation.



Busy Mapping Interface



Clean Mapping Interface

Interface | Drag-and-Drop | Rule Components | Mapping Tools

◀◀◀ select a topic to learn more about Delta's mapping features

Easy Modeling

Mapping

Map Testing

Solution Manager

Trading Partner Management

# Drag-and-Drop Mapping

Once your business object models are defined, the next step is to identify how the target model will use information from the source model. This is done by writing rules that link the source model's data elements to their proper positions in the target model. In Softshare Delta, rules can be written by dragging and dropping a source element onto a target element, or they can be written manually.

Delta supports both source- and target-driven mapping. By default, Delta performs rules based on source data structure, but if the structures of your data formats are too disparate, you can switch to target-driven mapping. Delta's target-driven mapping feature lets you reorder the objects in the source model so that they more closely match their counterparts in the target model.

The screenshot displays the Delta software interface for mapping a source database to a target EDI format. The top pane shows a list of rules for the mapping process. The middle pane shows the source and target object models. The source model is a table with columns: Shipment\_ID [integer], Original [boolean] Original or Duplicate, Pack\_Type [integer] 0001-Pick and Pack ; 0002-Standard, Vendors\_Ship\_No [string], Document\_Trans\_Date [datetime], Document\_Trans\_Time [datetime], Packaging\_Code [string] 'CTN25' or 'PLT76' [with constraint], Lading\_QTY [integer], Weight [integer], Unit\_Of\_Measure [string] 'LB' or 'KG' [with constraints], and Trans\_Meth\_Code [string] 'M' or 'A' or 'C'. The target model is a hierarchical structure with segments: BSN [object, mandatory] Beginning Segment for Ship Notice, BSN01 [string, mandatory] 353 - Transaction Set Purpose, BSN02 [string, mandatory] 396 - Shipment Identification, BSN03 [date, mandatory] 373 - Date, and BSN04 [time, mandatory] 337 - Time. A rule is shown in the top pane: 'create 856.BSN (Beginning Segment for Ship Notice)' with sub-rules: BSN01=IIF(AdvShipNotice.Shipments.Original = TRUE, "00", "07"), BSN02=AdvShipNotice.Shipments.Shipment\_ID + 10, BSN03=Document\_Trans\_Date, and BSN04=Document\_Trans\_Time. A yellow callout box contains the text: 'To map an element from your source model to your target model, drag it to the target model. The resulting rule appears in the rule pane.'

Interface | Drag-and-Drop | Rule Components | Mapping Tools

select a topic to learn more about Delta's mapping features

Easy Modeling

Mapping

Map Testing

Solution Manager

Trading Partner Management

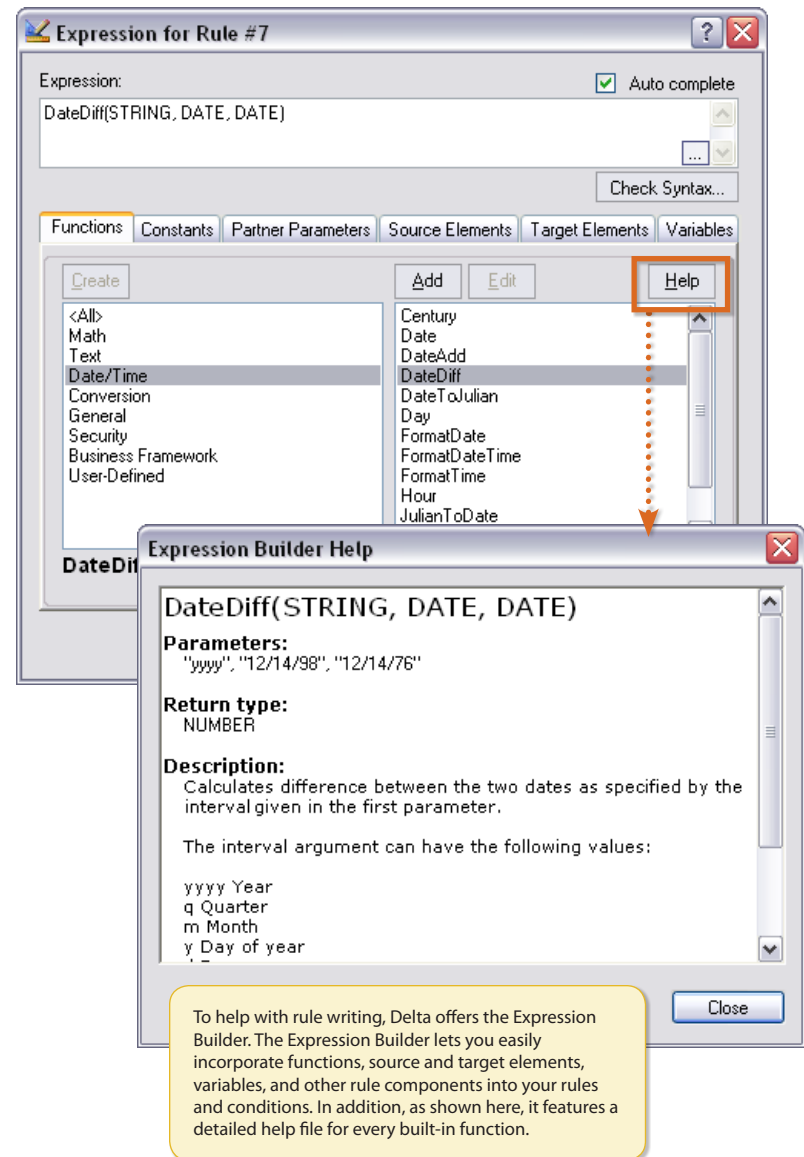
# Advanced Rule Components

Mapping is rarely as straightforward as linking one data element to another via dragging and dropping. Usually, your target data format requires data that is not present in your source data. Or, you may need to manipulate your source data before it meets the requirements of your target data format.

## Rule Components

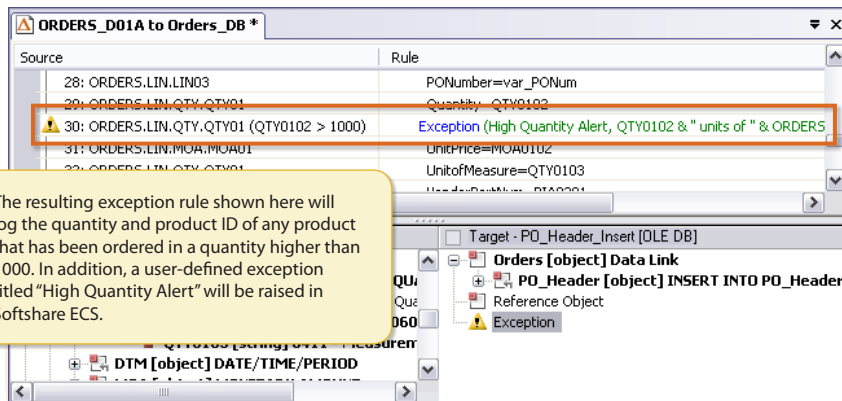
For those scenarios in which source data needs to be manipulated or dynamically produced, Softshare Delta offers a host of sophisticated rule components for you to incorporate into your maps, including:

- **Functions.** Functions are preprogrammed calculations that are carried out during map execution. Delta offers over 200 built-in functions to cover a wide range of mapping needs. For example, the *DBLookup* function retrieves information from a database during map execution, while the *MapData* function passes data to another map and returns that map's target data to the original map. If the calculation you require is not performed by any of Delta's built-in functions, you can write your own.
- **Variables.** Variables represent values that change as a map is executed. For example, you can create variables that track line item counts or value statuses as Delta moves through a map. In addition to the custom variables you create, Delta has many built-in variables that store data relevant to the map (e.g. trading partner data; source and target data object counts; and map success or failure).
- **Conditions.** Conditions are expressions that determine whether or not rules are performed by returning TRUE or FALSE values. Delta is very attuned to conditions, providing you with opportunities across many different contexts to designate mapping behavior when conditions are not met.
- **Arrays.** Arrays store a collection of similar data. For example, a common use of arrays would be to store all quantities for all line items in a purchase order. This array would typically be accompanied by one or more additional arrays that stored all part numbers or all descriptions or all store locations for all line items in the purchase order. Delta supports one-dimensional and multi-dimensional arrays.
- **Sequences.** You can use Delta's built-in *Sequence* function to track temporary and permanent sequences throughout your maps. A temporary sequence is automatically reset to its initial value each time a map executes. A permanent sequence is stored in Delta's database and continues where it left off each time a map executes.



# Mapping Tools

Softshare Delta offers a large number of specialized tools to assist you in your mapping efforts. These tools range from simple rule disabling commands to more complicated ones that alter default rule timing or trigger external processing. Regardless of complexity, these tools exist to make Delta's mapping process as flexible as possible to suit your unique integration requirements. Next, we've highlighted just a few of the mapping tools that you'll enjoy as a Delta user, but there are many others that you'll discover once you begin using Delta.

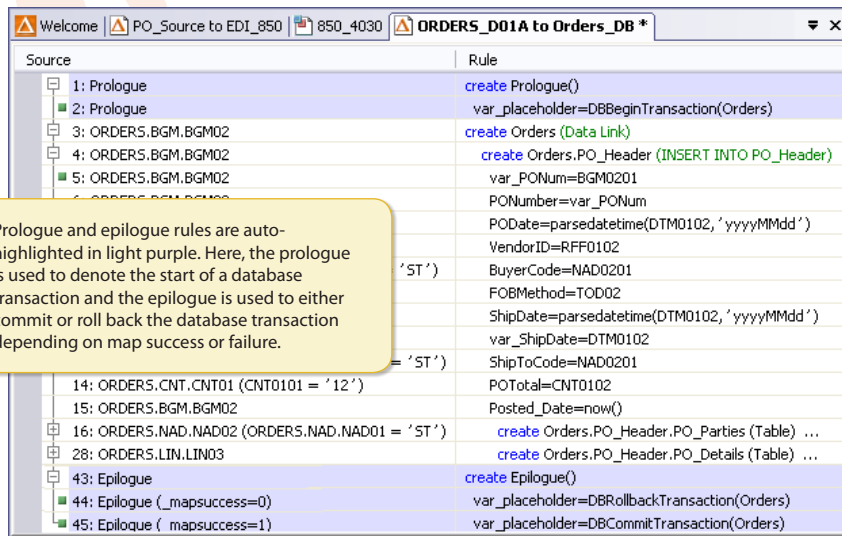


## Exception Handling

At the bottom of every target model that has been incorporated into a map is an object titled "Exception." Mapping to this target object allows you to write exception rules that determine (1) the mapping condition that will cause an exception, (2) the action that occurs when this condition is met, and (3) the message that is logged as a result. In addition, if you're running Softshare ECS, you can specify that the exception be raised and handled by ECS' sophisticated exception handling system, which is capable of e-mail notification of the exception.

## Prologue and Epilogue Rules

Prologue and epilogue rules allow you to perform in-map processing. Prologue rules run before any source-driven rules fire and epilogue rules run after all source data is exhausted. For example, prologue rules could be used to update database tables before map execution. Epilogue rules could be used to execute scripts, external commands, or even other maps upon map completion. Delta even offers a built-in `_mapsuccess` variable for the purpose of triggering epilogue rules based on map success or failure.



## Altering Rule Timing

When you write a rule, rule timing defaults to *always*. A rule timing of always indicates that Delta will perform the rule each time it encounters the source object that the rule fires from. If this default timing doesn't suit your needs, you can change it to *first time* or *last time*. These alternate rule timings instruct Delta to only perform rules the first or last time the source objects that the rules fire from are encountered, rather than every time.

## Repeat Column

The Repeat column allows you to indicate the number of times a rule repeats. In most maps, the Repeat column isn't needed because your source data correctly determines the number of times a rule repeats. But there may be some instances when you want to run a rule or set of rules more often than the source data dictates. Unloading arrays is a good example of when the flexibility of the Repeat column comes in handy.

Interface | Drag-and-Drop | Rule Components | Mapping Tools

◀◀◀ select a topic to learn more about Delta's mapping features

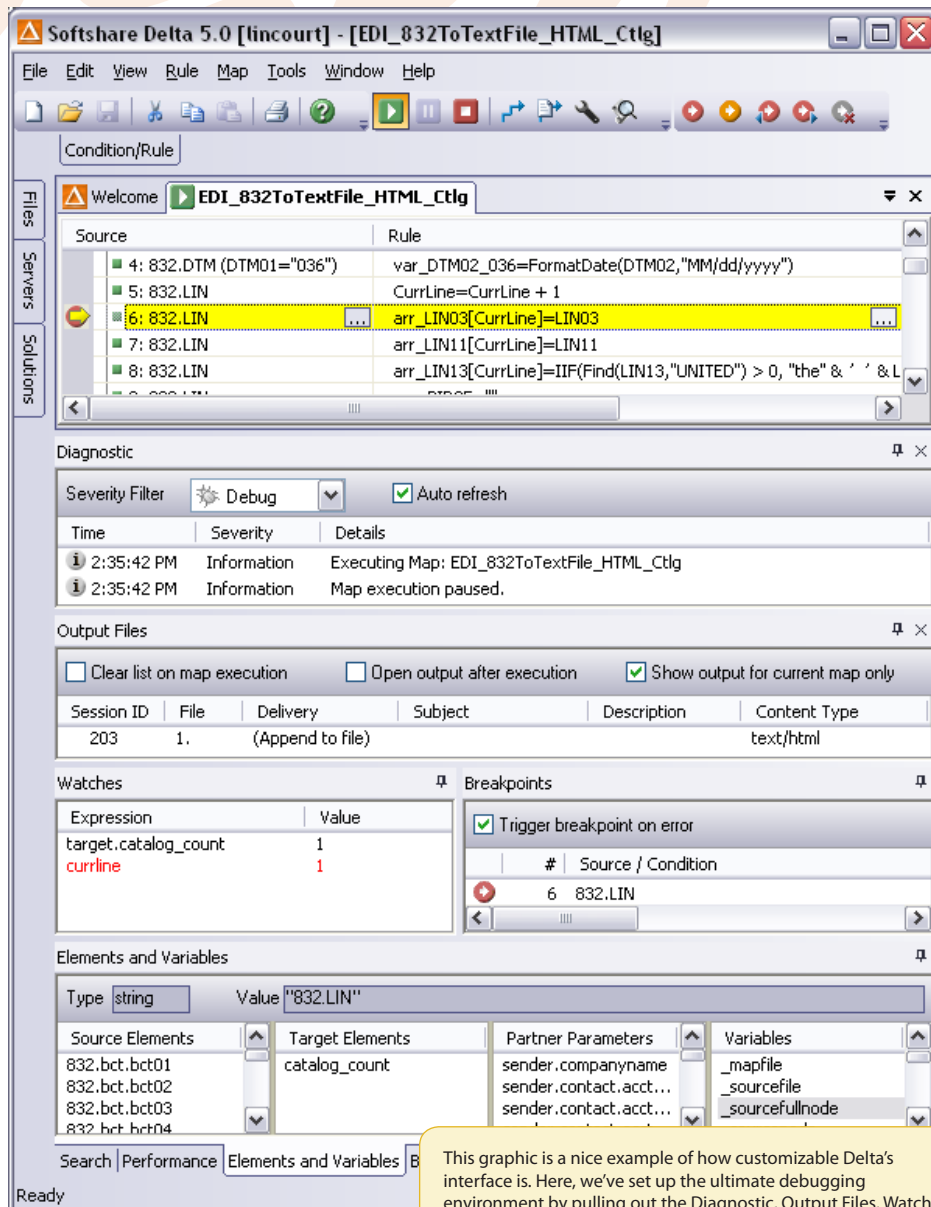
Easy Modeling

Mapping

Map Testing

Solution Manager

Trading Partner Management



# Map Testing & Debugging

Softshare Delta's map testing and debugging features help ensure that your map is accurate before you place it into production.

## Map Testing

Using Delta's map testing feature, you can perform a dry run of your map. This dry run lets you view the map translation process and the resulting map output, but does not deliver the map output or otherwise impact any operations outside of Delta.

The Diagnostic pane logs the details of the test run, alerting you to any errors that were encountered. You decide the level of detail that is logged, ranging from a severity level of debug, which tracks every action Delta takes for informational purposes, to a severity level of error, which indicates that map execution was halted due to an unrecoverable error.

Immediately after the test run, the Output Files pane displays the temporary file(s) that were created as a result of the test run. You can double-click these files to view test results, save them to your system, or, as they accumulate, compare them to one another using Delta's file diff feature.

## Map Debugging

To pinpoint elusive errors, Delta allows you to set breakpoints. Breakpoints pause map testing at either user-defined locations in the ruleset or when user-defined expressions evaluate to TRUE. While paused at a breakpoint, there are a number of debugging activities you can perform, including:

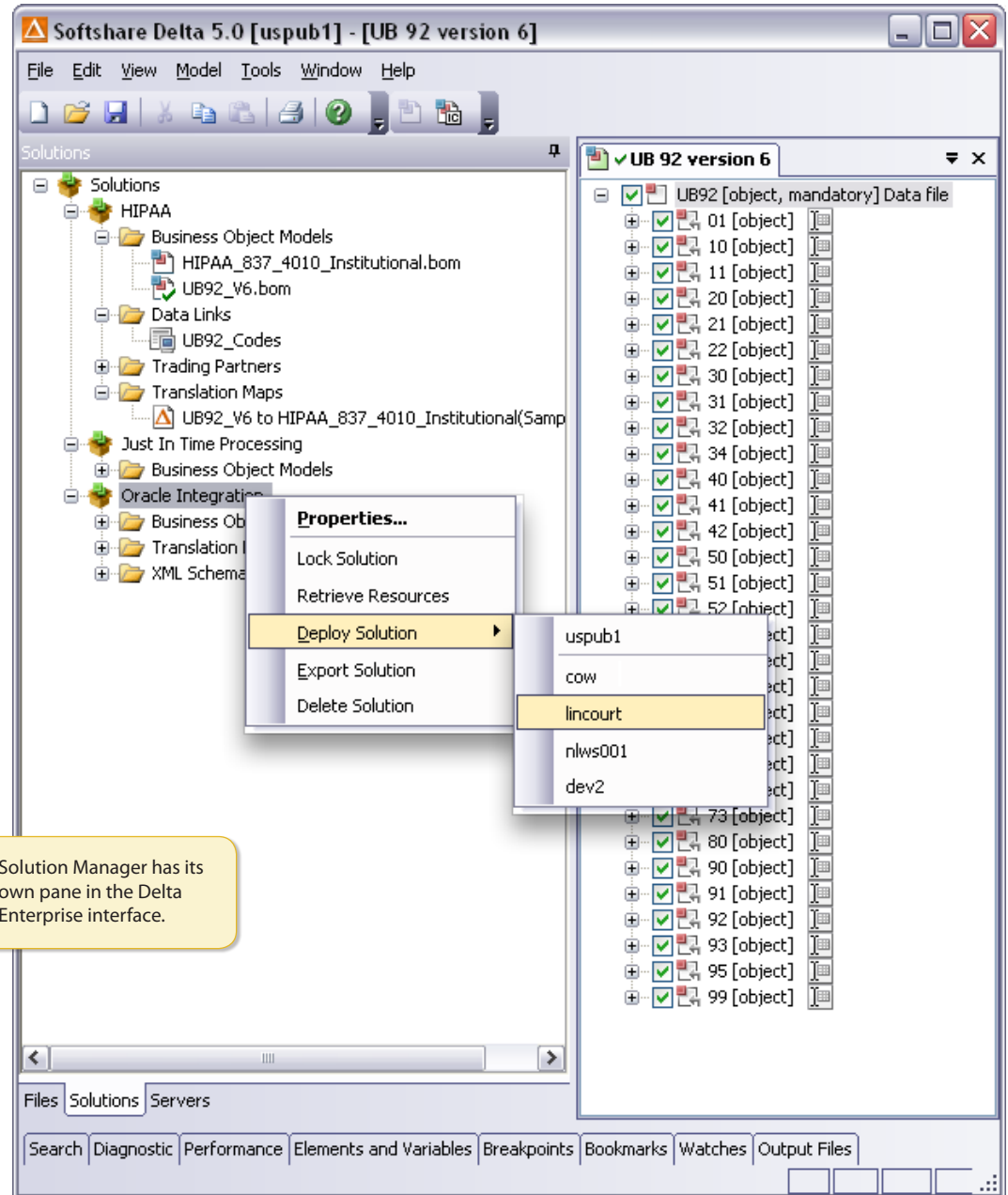
- Viewing the values assigned to the elements and variables in your source and target models at the moment the breakpoint is encountered
- Viewing the current evaluations of any expressions you are specifically monitoring (known as watches)
- Viewing the source data and Delta's current position within it
- Stepping through one or more of the remaining rules one at a time

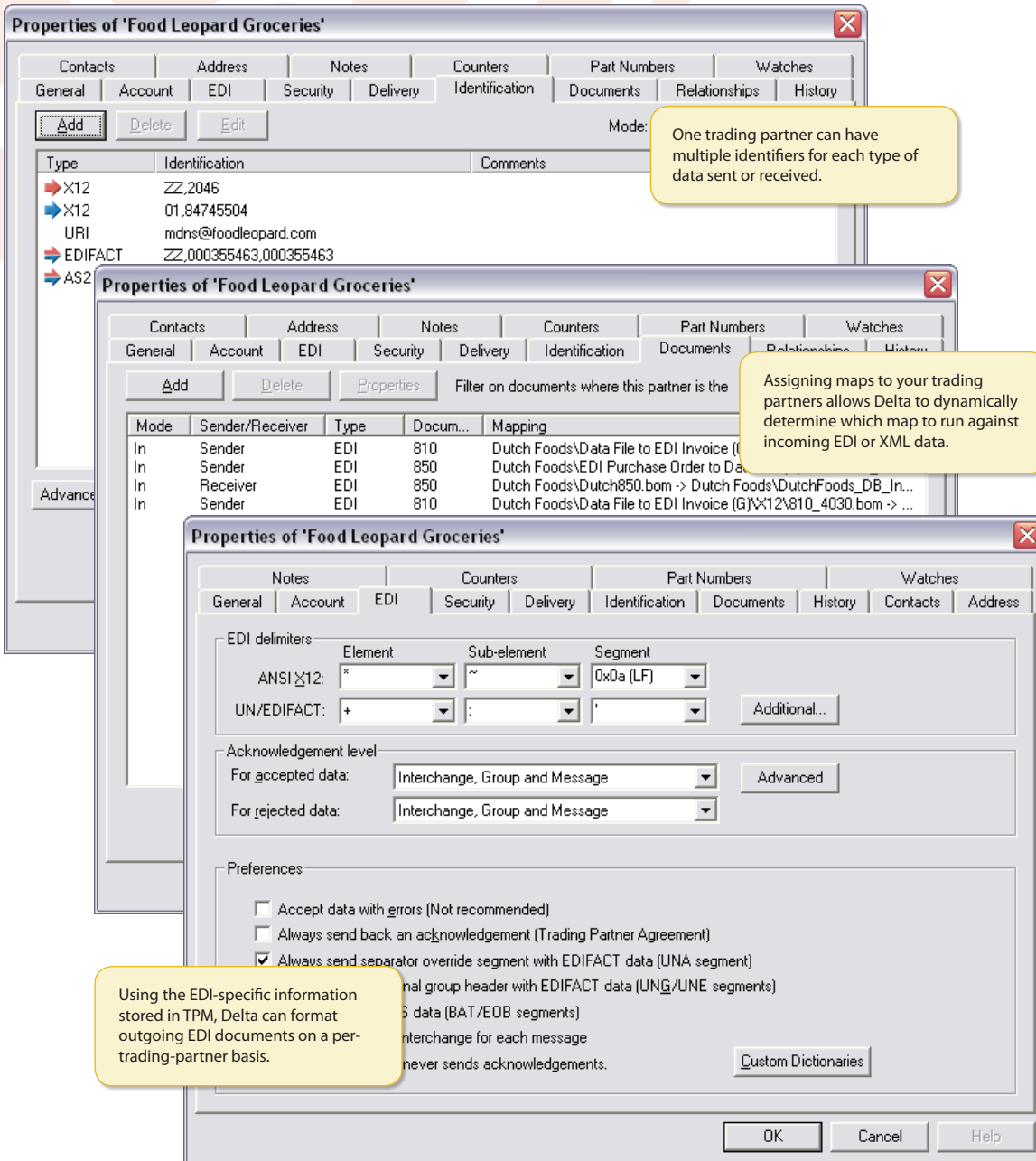
# Solution Manager

Solution Manager is an integrated component of the enterprise edition of Softshare Delta (it is not available in Delta Standard). Intended for users of both Delta Enterprise and ECS, it manages solutions and their associated resources in a central repository. For example, a solution could consist of a map, the map's source and target models, as well as the trading partners and ECS output channel assigned to it.

The ability to identify solutions and store them as collections of resources in one central data repository gives you an incredible amount of control over your integration operations, including the ability to:

- Group maps that are related in some way
- Easily deploy solutions to any ECS server
- View all solutions and resources currently deployed
- Implement versioning and detailed tracking of edits to a solution or solution resource
- Roll back to an older version of a solution or resource, or roll forward to a newer version
- Instantly see and repair resources that aren't resolving for a particular ECS server
- Export/import solutions





## Trading Partner Manager

Using Trading Partner Manager (TPM), an application that ships with Softshare Delta, you can store a large amount of data for each of your trading partners such as contact information, part numbers, sender and receiver IDs, delivery instructions, and internal IDs assigned by your company. For EDI trading partners, you can also store EDI-related data such as qualifiers; ISA and GS identifiers; element, sub-element, and segment delimiters; control number sequences; and functional acknowledgment generation procedures. During map creation and execution, Delta gives you full access to the data in TPM so that you can incorporate trading partner data into your map rules.

In addition to storing trading partner data, TPM is also the link between a map's source or target data and the trading partner that is sending or receiving that data. When you link your maps to their related trading partners in TPM, Delta is able to automatically address outgoing translated documents, automatically translate incoming documents, and automatically generate functional acknowledgments.

Delta's tight integration with TPM makes trading partner lookups possible. Trading partner lookups let Delta dynamically determine, upon map execution, where a map's EDI, XML, or text document output should be delivered based on information found in the source data. By taking advantage of trading partner lookups, you can automatically turn around translated output to the original sender or deliver different portions of a map's output to different trading partners.

# Integration With Softshare ECS™

Pair Softshare Delta with Softshare ECS, our communications server, and you have an enterprise application integration (EAI) solution capable of coordinating the many diverse applications, databases, and e-commerce formats found in your enterprise. In this role as middleware, the two applications work seamlessly together, with ECS delivering data to Delta for translation and then picking it up again for delivery to external trading partners or internal back-end systems. When ECS passes arriving EDI or XML data to Delta, Delta uses the data's content to automatically determine the appropriate map(s) to run. In addition, you can establish map execution schedules from ECS.

When Delta and ECS are run in tandem, you'll have the following capabilities, technologies, and features at your fingertips:

- Automated application-to-application integration
- Automated e-commerce-to-application integration
- Data pickup and delivery using a wide variety of transfer methods
- Sophisticated event-driven processing
- Detailed error tracking and notification
- Data storage and archiving
- Web services\*
- Failover and scale-out processing\*\*
- Extensive solution management \*
- System management via the Web

\*Requires the enterprise version of Delta

\*\*Requires the enterprise version of ECS

## Mapping from EDI

The following six translation scenarios feature EDI as source data. The specifications for the source EDI file that these tests were performed with are:

- ANSI X12 4010 version
- 1,000 total 850 transactions (purchase orders) in 1 ISA envelope
- 6,064 total line items
- 692,224 bytes file size

Target	Translation Time (Secs)	Orders/Sec	Bytes/Sec
EDI (855 v 4010) (Env. to Env.)	10.950	91.324	63,216.80
Data File	9.450	105.820	73,251.22
XML	11.650	85.837	59,418.37
Database SQL Server	SQL Insert	25.400	39.370
	OLE DB Model	175.000	5.714
	Parameterized Model	49.350	20.263

## Mapping from XML

The following five translation scenarios feature XML as source data. The specifications for the source XML file that these tests were performed with are:

- 1,000 total XML documents (purchase orders)
- 6,064 total line items
- 11,000,180 bytes file size

Target	Translation Time (Secs)	Orders/Sec	Bytes/Sec
EDI (850 v 4010)	5.350	186.916	207,479.63
Data File	5.850	170.940	189,746.32
Database SQL Server	SQL Insert	9.650	103.627
	OLE DB Model	176.500	5.666
	Parameterized Model	15.100	66.225

## Mapping from Data File

The following two translation scenarios feature a data file as source data. The specifications for the source data file that these tests were performed with are:

- Structured data file (i.e. contains two record types)
- 1,000 total header records (representing purchase orders)
- 6,064 total detail records (representing purchase order line items)
- 593,920 bytes file size

Target	Translation Time (Secs)	Orders/Sec	Bytes/Sec
EDI (850 v 4010)	4.350	229.885	136,533.33
XML	3.950	253.165	150,395.49

# Softshare Delta Benchmarks

Softshare tested a number of common translation scenarios to provide you with relevant time benchmarks for Softshare Delta. Each scenario was executed several times and the following tables represent the averages.

Benchmark testing was performed on a Dell Optiplex 745 with an Intel Core 2 4300 (1.8 GHz) processor and 3 GB of RAM. Maps were run in Delta test mode with as little else as possible running on the system. Test results were achieved by placing logging into the maps. Each map logged a start time, a "split time" for each 1000 orders processed, and a final translation time. Because benchmarks were tracked using this logging method, map load time is not included in these results.