1. Provide a general description of the maturity of the product and the history of the tool connecting to DB2/400 and SQL Server databases. Include the tool’s operating environment, and how it connects to the iSeries and whether it can update data in real time. Describe your data transformation layer and product and how it translates data from a source database such as DB2/400 to SQL Server 2000/2008 tables. Describe your product development cycle. How often are there upgrades available? How often are features deprecated?

**Environment**

1. In what environment does the tool run?
2. In what programming language is the tool built?
3. Can the software invoke native functionality on the iSeries?
4. What are your hardware requirements -- both minimum and recommended requirements?
5. Provide message server reliability statistics.
6. Does the solution offer guaranteed message delivery and how does this impact server performance?
7. How scalable is the vendor system? Describe your experience with starting with a small number of apps and adding applications as time moves on. What was the effort involved?
8. Is our investment in the data integration business logic coding preserved in the event that we need to change operating system environments or change processor speed, i.e., 32 bit to 64 bit, etc?
9. Describe your connectivity layer, such as ODBC or JDBC data connectivity, and your transportation layer.

**Ease of Development/System Usage**

1. Describe a typical development and integration project.
2. Describe the interface for business process modeling, business rules, workflow, and data transformations. Describe how to develop a rule-based messaging system.
3. Describe any platform specific limitations and restrictions.
4. Describe how the software will handle null field data.
5. Our DB2/400 date fields are defined as 8 character numeric fields formatted as YYYYMMDD. How will the software handle converting this to true date formatted fields such as MM/DD/YYYY?
6. Is there a GUI that allows for development?
7. Can you develop interfaces without being a Java or C# programmer?
8. What are the supported programming languages?
9. Does the system have the ability for sophisticated IT developers to extend functionality of the system without resorting to an external development environment?
10. Does the system support versioning and rollback in the GUI?

**System Management**

1. Describe system management once rolled into production. Describe the environment for viewing messages consumed/delivered. Are there dashboards for this?
2. Describe how your tool will update data and stay in sync in real time. Do you support real time data updates?
3. Are there dashboards configurable for different types of users?
4. What is the mechanism to promote development changes to production?
5. Does the tool offer versioning and version control? Is the recommended configuration to have a separate development environment?
6. Are there automated alerts sent via email upon detection of significant events/exceptions?
7. Describe the alerts and exception handling configuration.
8. Does the system have the ability to aggregate, analyze, and report on messages flowing throughout the system without relying on an external third party business intelligence software add-on?
9. Does the system have the ability to stop, start, and upgrade individual interfaces while the system is running?
10. Does the system have the ability to trace any message through the system, to see how it is transformed at each integration point and see any errors associated with it, and with the duration of each step in the process flow?
11. Does the vendor support a wide range of communication standards and database adapters out of the box?
12. Does the software automatically persist all data messages to a database?
13. Provide your technical support model, i.e., location, hours, etc.