1. **Difference between SQL direct and JDBC Query**

Ans: 1. SQL Direct support DDL statements where as JDBC Update does not support the same

2. Dynamic queries can be formulated and used as part of SQl Direct, which is not the case with JDBC Update

3. Using SQL Direct you can execute multiple queries at the same time, which is not possible in case of the JDBC Update

4. For SQL Direct you can decide input and output at the run time, where as in case of JDBC Update it should be decided at the design time

1. **what is the difference between ADB adapter and JDBC palette**

The Basic difference is that:  
  
ADB:--event driven  
JDBC:--Demand driven.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Adapter** | **JDBC Palette** |
| Ease of Integration | Design time and run time  environments, which are to be managed separately. | JDBC palette for both run-time and design-time operates in the same way. |
| Completeness | Very scalable component to  handle both inbound and outbound messages. | (No Trigger)BW JDBC  does not have a mechanism to detect database changes in a push  environment. |
| Consistent error viewing and handling | Adapter are  written using SDK and strictly adhere to standard error viewing and  handling. Thereby making monitoring easy. | JDBC provides error  objects and rules have to be written for error viewing and handling. |
| Transactions Volume | Adapters are useful in  high-to-medium load situations for both real-time and batch-updates.  Adapter can be run on the application machine using native lib. | BW palette are useful in medium-to-low level load real-time situation |
| Connectivity Design Time | JDBC | JDBC |
| Connectivity Run-Time | ODBC | JDBC |
| Transaction Support | When asynchronous operations are  acceptable. | Useful when dealing with synchronous model and you  need to take explicit action on transaction failure |
| SQL Statements | Useful when you are using  straight forward and simple SQL statements | Useful when the  database operations are not straight forward i.e. the operations depend  on a lot of "if..then..else" data dependent clauses. |
| Ease of Data Transformation, parsing and mapping | Data  is transformed to desired protocol format (JMS etc) at the source.  Parsing and mapping takes place in process flow. | Data  transformation, Parsing and mapping takes place in process flow. |
| Result-set format | ADB result-set | JDBC result-set, useful if integrating with any 3rd party database custom  application. |
| Exception Handling | If a subscriber adapter cannot write data to its destination table, it will write the data to the **exception table** | You have to implement exception handling in the  process. |
| Database Support | Oracle, MS-SQL, Sybase,  DB2 | Embedded drivers are: •tibcosoftwareinc.jdbc.oracle.OracleDrive •tibcosoftwareinc.jdbc.sqlserver.SQLServerDriver •tibcosoftwareinc.jdbc.sybase.SybaseDriver  There are additional supported drivers, please see release doc for details |
| Ease of Monitoring | Adapter enabled with Hawk framework  (admin) deployed as a service has much fine grain monitoring,  configuration & management than JDBC activity. | Palette  monitoring can be handled as embedded component in the process. |
| Component Management | Adapter is deployed as separate component from the process and requires management. | Palette is embedded in the process along with data parsing, mapping etc. and  does not require separate deployment |
| Protocol support | TIBCO EMS | TIBCO EMS |
| Loop Detection | Adapter has embedded provision to switch Loop detection on when trying to sync data between  source and target applications. | Depending upon implementation  may or may not require Loop Detection implementation. |

•   ADB uses ODBC to connect, JDBC uses JDBC  
•   ADB is more suitable for instances where you have a lot of processing  
•   ADB is more suitable for instances where you want that a particular action on a DB Table triggers a BW process.  
•   ADB adapter is best for publishing from database.  
•   For simple inserts and updates then ADB subscriber is best.   
•   In case of insert or update to database then check if you have complex JDBC inserts, transaction management and other dynamic queries then JDBC activities are best.  
•   JDBC is more suitable for running dynamic code where in runtime you can execute statements with different values depending on process execution

1. what is the difference between file adapter and File palette

|  |  |  |
| --- | --- | --- |
|  | **File Adapters** | **File Palette** |
| **Ease of Integration** | Adapter consists design time and run time environments, which are to be managed separately. | File palette does not require separate run-time and design-time. |
| **Completeness** | Not optimal for XML files  Support pre and post event script processing | File palette can handle XML file effectively, can handle script/custom code execution |
| **File Mode** | The adapter supports text files +  \*supports both text and binary files in file transferring mode+ | Palette has a lot more flexibility and support both Text and Binary mode. |
| **Operating Mode** | Not optimized for directory polling and dynamic binding. Adapter supports asynchronous mode | Support directory polling and all other basic file operations |
| **Consistent error viewing and handling** | Adapter is written using SDK and strictly adhere to standard error viewing and handling. Thereby making monitoring easy. | Palette provides error objects and rules have to be written for error viewing and handling. |
| **Transactions Volume** | Adapters are useful in high-to-medium load situations. File adapter can operated to send data in a batched mode | More geared towards reading the whole file and scales well. |
| **Connectivity**  **Design Time** | Adapter Design Time Agent | BW environment |
| **Connectivity**  **Run-Time** | Adapter Run Time Agent | BW environment |
| **Ease of Data Transformation, parsing and mapping** | Data is transformed to desired protocol format (JMS etc) at the source. Parsing and mapping takes place in process flow. | Data transformation, Parsing and mapping takes place in process flow. |
| **Ease of Monitoring** | Adapter enabled with Hawk framework (admin) deployed as a service has much fine grain monitoring, configuration & management than File activity. | Palette monitoring can be handled as embedded component in the process. |
| **Component Management** | Adapter is deployed as separate component from the process and requires management. | Palette is embedded in the process along with data parsing, mapping etc. and does not require separate deployment |
| **Protocol support** | TIBCO EMS | TIBCO EMS |
| **Smart Publishing** | On a slower consumer, Flow controls may be configured to control publishing rate. | In such a case palettes rely on design of the process |
| **Check Point** | The adapter publication service offers the facility to restart processing  and publication from where the publication service left off, just before exiting normally or abnormally. | File palettes may use checkpoints tasks/activity of BW. |
| **Duplication of Messages** | When the publication format is set to MInstance there is a possibility of  getting a duplicate message depending on when the adapter had exit. | File palette provides better options to avoid message duplication. |

1. **Queue v/s Topics**

When determining when to use queues versus topics consider the two fundamental messaging mechanisms. The first is point-to-point messaging, in which a message is sent by one publisher (sender) and received by one subscriber (receiver). The second is publish-subscribe messaging, in which a message is sent by one or more publishers and received by one or more subscribers. The messaging model as listed below will dictate when to use a queue or a topic:  
  
One-to-one messaging                   Queue    point-to-point     
One-to-many messaging   Topic   publish-subscribe     
Many-to-many messaging   Topic    publish-subscribe model

1. **soap over http vs soap over jms**

* HTTP: Synchronous, both server and client should be active when data is sent from server to client. It is very fast compared to JMS.

JMS: It can be both Synchronous and ASynchronous which means when data is sent from server client need not be active.JMS provides a highly reliable transport that can guarantee message delivery. .There will not be any data loss in JMS.

* If you do SOAP over JMS, in fact you can do load balancing..  where as with SOAP over HTTP requires additional hardware like IP Sprayers.
* Using SOAP over JMS gives you some advantages compared to HTTP, specially related to reliability as you may use the persistence and acknowledgment features built in the standard. The same applies if you need to establish asynchronous communication or need to use the load balancing features provided by JMS servers. You can achieve this using http but the implementation would be much more complicated.

Reason for choosing JMS in most cases is reliability. But there are other things that come in mind whether to choose JMS or HTTP.

Reasons to go with HTTP:

* Firewall friendly (web services exposed over internet)
* Supported on all platforms (easiest connectivity in b2b scenario)
* Clients can be simple and lightweight

Reasons to go with JMS:

* Assured delivery and/or only once delivery
* Asynchronous support
* Publish/subscribe
* Queuing if better for achieving larger scalability and reliability
* Better handles temporary high load
* Large volume of messages (EDA)
* Better support in middleware software
* Transaction boundary

In SOA architecture best practice is to use JMS internally (for clients/providers that can easily connect to ESB) and HTTP for connecting to outside partners (over internet).

Using SOAP over JMS gives  some advantages compared to HTTP,

specially related to reliability as you may use the persistence and acknowledgment features built in the standard.

The same applies if you need to establish asynchronous communication or need to use the load balancing features provided by JMS servers.

We can achieve this using http but the implementation would be much more complicated.  
  
If we do SOAP over JMS, in fact we can do load balancing..  where as with SOAP over HTTP requires additional hardware like IP Sprayers.