Synchronizing with the Back End

The application adaptor controls the synchronization of the data, state or objects (with the data or state stored in back end stores). The PAA and BOIM application adaptors have many attributes in the configuration of containers that will alter the schemes used to control this synchronization. These attributes are dataCachedInManagedObject, dataCachedInDataObject, useCachingService, terminationPolicy, defaultTransactionPolicy, and sessionPolicy.

Persistence

An object has two dimensions to persistence, persistent state and persistent references. While both of these dimensions refer to the durability, persistent state affects the business object while persistent references affects the user of a business object. All objects are persistent with the exception of the UUID objects.

For the UUID objects their references are either persistent or transient depending on the duration of the usage of the reference. If the reference is expected by the client to always be valid, then the reference is persistent. If however, once the object is passivated then the reference is not expected to be valid any longer, then the reference is transient.

Behavior in the Absence of a Transaction or Session

The BOIM AA and PAA provide the capability of putting UUID objects and persistent objects in the same container. For this to work then the defaultTransactionalPolicy or the sessionPolicy must be set to ignoreCondition.

Some applications may desire a transaction to last for the duration of a method. BOIM application adaptor provides this capability by starting a transaction prior to dispatching the method on an object, and committing the transaction after the method has completed. For this behavior the defaultTransactionPolicy needs to be set to Atomic.

Container Configuration

This describes the combination of settings for Containers that can be used with the DO Implementations provided by Component Broker. All configurations listed are supported and valid; all other configurations are not supported. The information is organized by DO Implementation and Service setting. Information is provided for all settings that the user can adjust in the Container wizard of Object Builder when creating a new container. These settings can be modified after a container has been created using the System Management interface. These modifications are described in “Mapping Object Builder and Systems Management Terminology” on page 256.1. The following are supported for all configurations:

- Both deployment platforms (Windows NT and AIX)
- Workload management
- Both BO Data access patterns (Delegating and Caching)
- Both DO Data access patterns (Delegating and Local Copy)

This section contains the following topics:

- “Transient DO” on page 256
- “DB2 Embedded SQL DO” on page 256
- “DB2 Caching Service” on page 256
- “Oracle Caching Service” on page 256.1
- “Procedural Adapter DO” on page 256.1

Transient DO

Table 8.1. No Object Service

Save all data when server is about to stop	Passivate a component after checkpoint	Enable persistent references	Use Caching Service
Both settings are supported	No	Both settings are supported	No
Both settings are supported	Yes	Yes	No

Table 8.2. RDB Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction	Use Caching Service
Both settings are supported	Start a new transaction	No
Both settings are supported	Throw an exception and abandon	No
Yes	Start a new transaction	Yes
Yes	Throw an exception and abandon	Yes

Table 8.3. PAA Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction
Yes	Start a new transaction
Yes	Throw an exception and abandon

Table 8.4. PAA Session Service

Passivate a component at the end of a session	Behavior for methods called outside a session	Connector type used by a Session
Yes	Throw an exception and abandon	All settings are supported

DB2 Embedded SQL DO

Table 8.5. RDB Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction	Use Caching Service
Both settings are supported	Start a new transaction	No
Both settings are supported	Throw an exception and abandon	No

DB2 Caching Service

Table 8.6. RDB Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction	Use Caching Service
Yes	Start a new transaction	Yes
Yes	Throw an exception and abandon	Yes

Oracle Caching Service

Table 8.7. RDB Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction	Use Caching Service
Yes	Start a new transaction	Yes
Yes	Throw an exception and abandon	Yes

Procedural Adapter DO

Table 8.8. PAA Transaction Service

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction
Yes	Start a new transaction
Yes	Throw an exception and abandon

Table 8.9. PAA Session Service

Passivate a component at the end of a session	Behavior for methods called outside a session	Connector type used by a session
Yes	Throw an exception and abandon	All settings are supported

Mapping Object Builder and Systems Management Terminology

Container settings can be modified using the Systems Management interface after the container has been created using Object Builder. This section provides a mapping of terminology between these two tools. These settings must be used with care so that the result is a supported configuration as shown in the previous tables. Object Builder prevents the user from selecting some unsupported settings (e.g. Caching Service cannot be used with PAA services), any Systems Management setting that does not have an associated setting in the previous tables must not be changed for use in that particular configuration.

Table 8.10 (Page 1 of 2). Object Builder and Systems Management Terminology

Object Builder Terminology	Systems Management Terminology
Behavior for methods called outside a session	Session policy
Behavior for methods called outside a transaction	Default transaction policy
BO data access pattern	Data cached in managed object
DO data access pattern	Data cached in data object
Enable persistent references	Persistent references
Passivate a component after checkpoint	Memory management policy
Passivate a component at the end of a session	Memory management policy

Table 8.10 (Page 2 of 2). Object Builder and Systems Management Terminology

Object Builder Terminology	Systems Management Terminology
Passivate a component at the end of a transaction	Memory management policy
Use caching service	Use caching service

Transient Transactional Data Objects

The recommended transactional container setting to use with Transient Transactional DOs is “start a new transaction.” Using the “start a new transaction” container setting causes the framework to start a transaction before each method is called on the managed object if no transaction exists, and to commit the transaction at the end of the method if the “start a new transaction” mechanism started the transaction. A rollback can also be performed automatically in the event of an exception.

Choosing the Never Passivate Policy for Passivation

The recommended “Passivate a component at the end of a transaction” setting to use for Transient Transactional DOs is "no". The managed object can then be used continuously, and still retain its “start a new transaction” transactional characteristics. If the container has a “Passivate a component at the end of a transaction” setting on "no", the object will exist in the container as long as no one performs a remove() method on the object. When choosing “Passivate a component at the end of a transaction” setting on "no", you should use a RDB container.

Transactional Semantics

If your managed object is an application object that is a singleton, be careful because this will have the effect of serializing any transactions that use the application object. Therefore, if you are using a Transient Transactional Object from a “start a new transaction” container, it is better for every client to use a uniquely keyed application object.

The results of the commit and rollback operations for the Transient Transactional Object mirror those of the DB2 Embedded SQL backed object. The key difference is that no default persistence backs up the data object. In addition, the rollback operation has no effect on any persistent back end.

However, it is important to remember that the syncToDataObject() method of the managed object (if caching is used) and the updateToDatastore() method of the data object are still called as a result of the commit operation. If the customer code has logic in these methods, the commit will cause these methods to execute and that logic will thus be executed.

Settings for Transient Transactional Data Objects

The following summarizes the container settings that are recommended for Transient Transactional DOs:

Table 8.11. Container settings for Transient Transactional DOs

Passivate a component at the end of a transaction	Behavior for methods called outside a transaction
No	Start a new transaction

Summary of Configuration Options on Container

Table 8.12. Configuration Options on Container

Data Object Type	**Memory Management Policy**	**Synchronization Policy**	**Persistence vs. Transient**	**Include UUID, roll your own, or objects in the same container**
Static embedded SQL for DB2	Passivate at end of transaction	noSession, Data is Cached in Data Object	persistent references and persistent objects	default transaction policy set to ignore condition
DB2 with Caching Service	Passivate at end of transaction	noSession, usesCachingService	persistent references and persistent objects	default transaction policy set to ignore condition
CICS and IMS	Passivate at end of session	noTransaction, Data is not Cached in Data Object	persistent references and persistent objects	default transaction policy set to ignore condition
Oracle with Caching Service	Passivate at end of transaction	noSession	persistent references and persistent objects	default transaction policy set to ignore condition
Roll your own passivation	neverPassivate or Passivate after checkpoint	noTransaction, noSession persistent refs	persistent references and persistent objects	
UUID - transient object refs	1. neverPassivate or 2. noTransaction and passivate at end of transaction	N/A	not persistent objects and not persistent refs	
UUID - persistent object refs	1. neverPassivate or 2. noTransaction and passivate at end of transaction		not persistent objects and persistent refs	

Configuring Application Adaptors – RDB

Business objects run in containers as described previously. Containers are part of the bigger component of Component Broker known as application adaptors. Each application adaptor provides a different quality of service. This quality of service manifests itself in a number of ways, one of which is through the containers that are surfaced by any given adaptor. This section enumerates specifics about the relational database adaptor provided by Component Broker.

Updating the Database Manager Configuration for the Transaction Processor Monitor

To use Object Transaction Service (OTS) with DB2, the transaction processor monitor must be configured with the Transaction Service DLL. To perform the configuration, the following should be typed at a DB2 prompt:

```
update database manager configuration using TP_MON_NAME somtrx1i
```

The request updates the configuration for the transaction processor monitor. The change does not take effect until the database is stopped and started. To validate that the configuration has been updated, type the following at a DB2 prompt: