

```
// OfferServer.h - OFFER SERVER DECLARATION
//
// MAINTENANCE HISTORY
// DATE          PROGRAMMER AND DETAILS
// 23-09-12 JS   Original
//
//-----

// OFFER STRUCTURE

struct Offer_t {
    long    offerCode; // Offer code
    long    customerId; // Customer identifier
    long    bcsId;     // Business customer's server identifier
};

//-----

// OFFER SERVER CLASS DECLARATION

class OfferServer_c {

    // Offer States

    enum OfferState_t {
        OFFER_STATE_OPEN, // The offer is still open
        OFFER_STATE_QUEUED, // Reply is queued to the BCS
        OFFER_STATE_SENT, // The reply has been sent to the BCS
        OFFER_STATE_ANSWERED // The BCS has answered the reply
    };

    // Offer Key Structure

    struct OfferKey_t {
        long    offerCode; // Offer code
        long    customerId; // Customer identifier

        // Default Constructor

        OfferKey_t ()
        {
            // Empty
        }

        // Construct from Values

        OfferKey_t (long offerCodeParm, long customerIdParm)
        {
            offerCode = offerCodeParm;
            customerId = customerIdParm;
        }

        // Comparator

        bool
        operator < (const OfferKey_t &offerKey)
        const
```

```

    {
        return offerCode < offerKey.offerCode ||
            (offerCode == offerKey.offerCode &&
             customerId < offerKey.customerId);
    }
};

```

// Offer Data Structure

```

struct OfferData_t {
    long        bcsId;          // Business customer server id
    OfferState_t offerState;    // Offer state
    long        atmId;         // Answering ATM identifier
    bool        offerAccepted; // Offer accepted flag
    bool        replyAccepted; // Reply accepted flag
    string      reason;        // Reason for rejection
};

```

// Offer Map

```

typedef map<OfferKey_t, OfferData_t> OfferMap_t;
        // Offer map type definition
typedef OfferMap_t::iterator OfferIter_t;
        // Offer iterator type def

```

// Reply Queue

```

typedef list<OfferKey_t> ReplyQueue_t;
        // Reply queue type definition
typedef ReplyQueue_t::iterator ReplyIter_t;
        // Reply iterator type def

```

// Retry Data Structure

```

struct RetryData_t {
    time_t      retryTime;    // Retry time
    OfferKey_t  offerKey;     // Offer key
    ReplyQueue_t replyQueue;  // Reply queue
};

```

// Retry Map

```

typedef map<long, RetryData_t> RetryMap_t;
        // Retry map type definition
        // Key is BCS identifier
typedef RetryMap_t::iterator RetryIter_t;
        // Retry iterator type def

```

// Private Variables

```

class OfferComms_c *offerComms; // Pointer to comms instance
OfferMap_t        offerMap;     // Offer map
RetryMap_t        retryMap;     // Retry map

```

// Public Methods

```

public:
void Initiate (class OfferComms_c *offerComms,
               const Offer_t *offerArr, size_t offerCnt);
        // Process a system restart

```

```
void Tick ();  
    // Process a clock tick  
void ReplyReceived (long atmId, long offerCode, long customerId,  
    bool offerAccepted);  
    // Process a reply received from an ATM  
void AnswerReceived (long businessId, long offerCode, long customerId,  
    bool replyAccepted, const char *reason);  
    // Process an answer received from a business  
    // client's server  
};
```