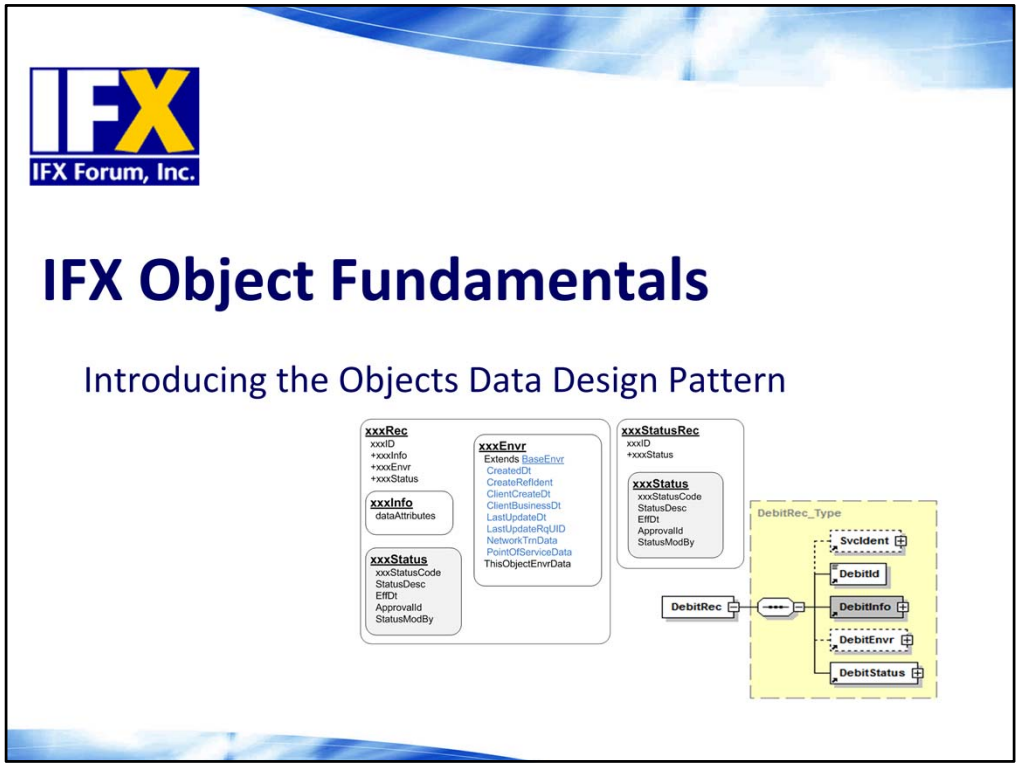
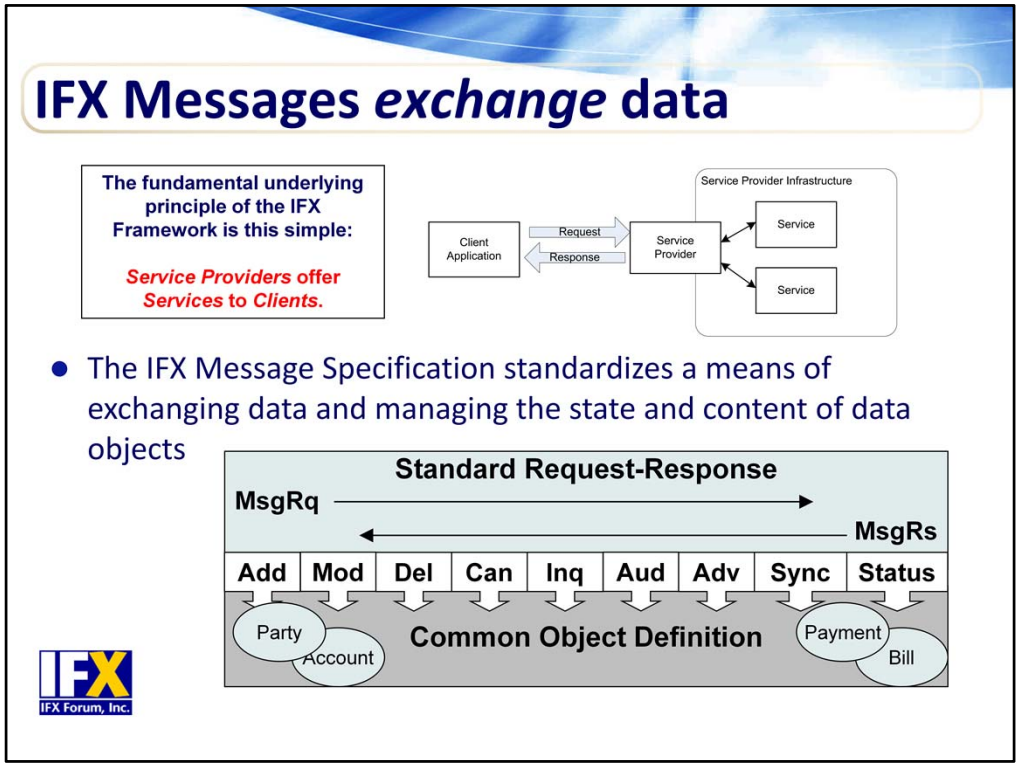


Welcome to the continuing series of IFX Orientation presentations.

In this presentation we will begin to dig a bit deeper into the IFX Object Model.



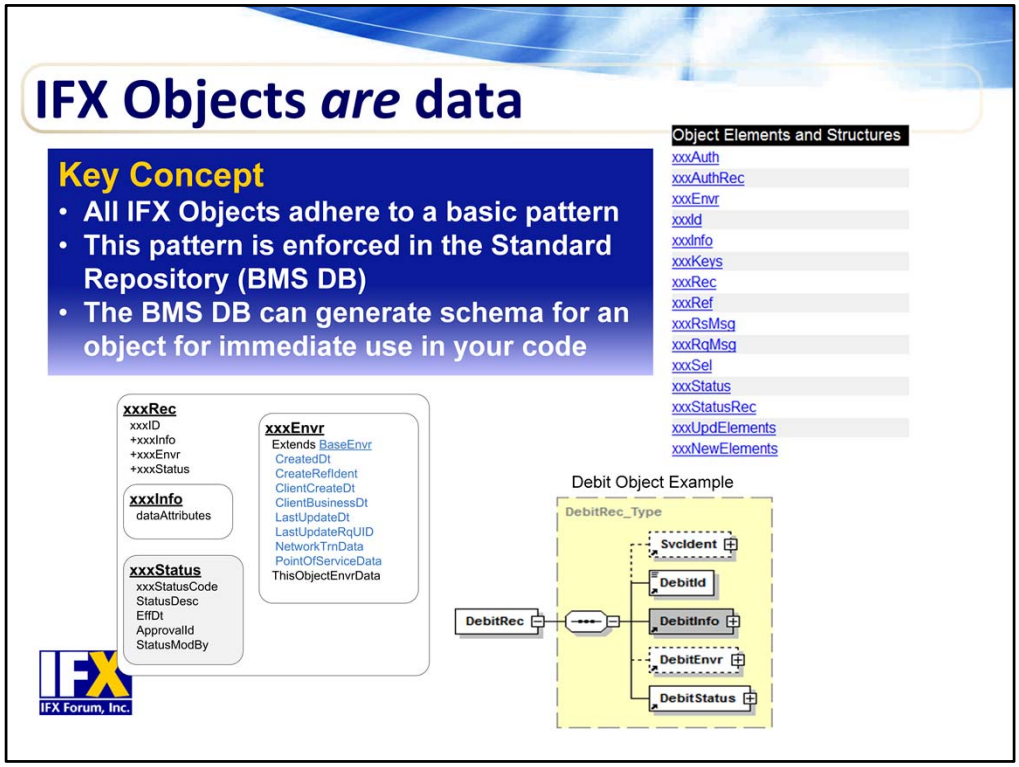
In other presentations we have discussed the underlying principle of the IFX Framework : Service Providers offer Services to Clients and that the IFX Message Model provides a mechanism to exchange data between service providers.

We also discussed the fact that IFX Messages are always addressed to a specific type of object – and usually act on a single, specific instance of an object.

With that background we can now delve into how data is structured in the IFX standard to facilitate consistent representation of – and effective management of – financial data .

In the IFX framework, data is consistently represented as objects, and all objects are structured according to well-defined rules.

So let's dig in!

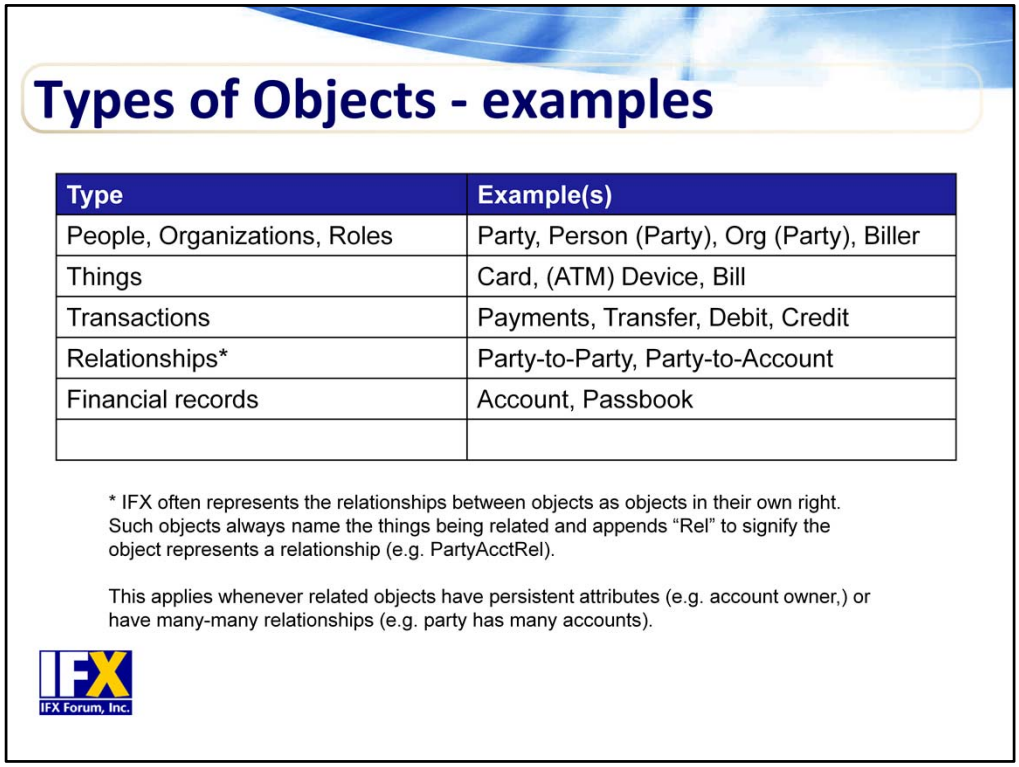


This slide shows several ways to visualize IFX Objects.

1. On the upper right portion of the slide is a complete list of elements that will be found in IFX Objects. (the xxx represents an object name)
2. The lower right shows how an object looks in an XML Schema; and
3. The lower left shows a container hierarchy to help visualize data 'packaging'

There is a lot of detail packed into this slide and by the end of this presentation you should find that you have a pretty solid understanding of how it represents these key facts:

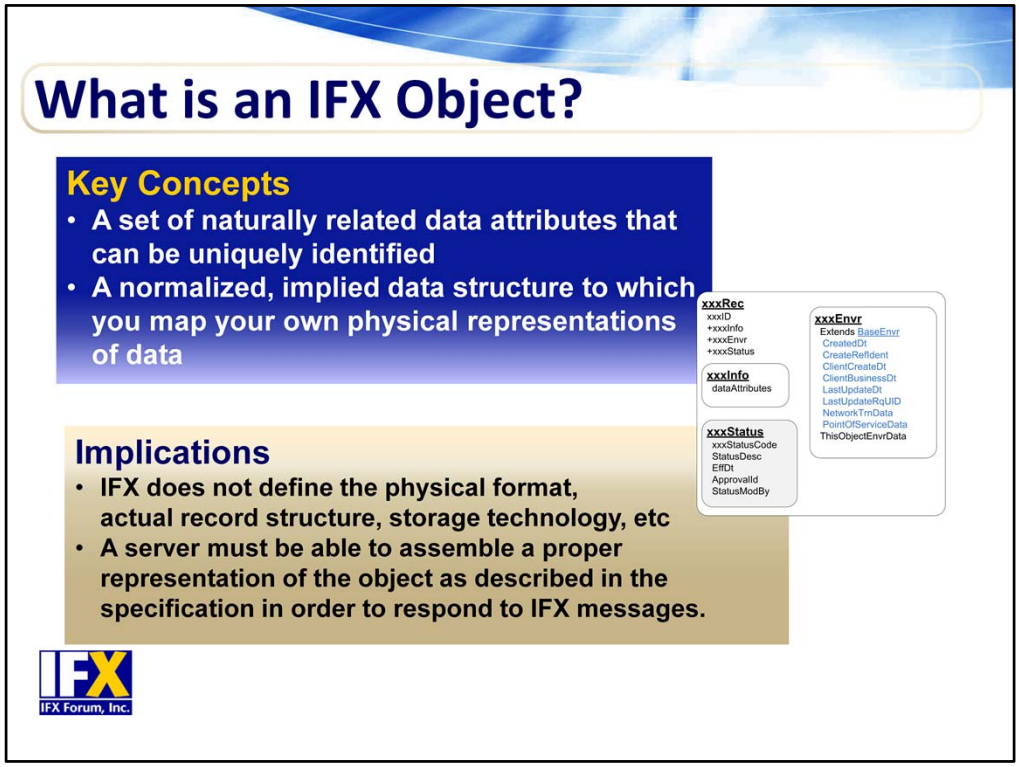
- Every IFX object adheres to this exact same design pattern
- The names of the data elements in every object are constructed exactly the same way. You have only to substitute the name of the object for the xxx pattern in these lists to arrive at the exact tag name for specific group of data elements. (Notice how the word "Debit" is combined with the element name in the schema example.)
- This consistent application of rules and patterns makes it easy to extend the standard and makes it clear where your data extensions will reside
- In the same spirit, this consistency shows up in the schema generated from the BMS making it easy for developers to find data and work with the standard.



Before we get too deep into the object structure, lets take a quick look at how the IFX Object model maps to the real world.

The table is mostly self explanatory. In the left-hand column are types of objects typically found in the financial services domain. On the right are some specific examples. (By the way, the IFX implementation guide – available to members and subscribers – actually provides a fairly detailed list of financial objects, their synonyms and the IFX object used to represent them.)

It's worth a special note to recognize that 'Relationship' objects adhere to all of the same rules and patterns as other IFX objects. The naming of relationship objects requires us to name the 2 object types being related and, for clarity, we add the suffix "REL" to indicate the object contains data about the relationship rather than the objects themselves. Data base modelers will recognize this pattern as an intersection table in 3rd normal form data models which also supports many-to-many relationships.

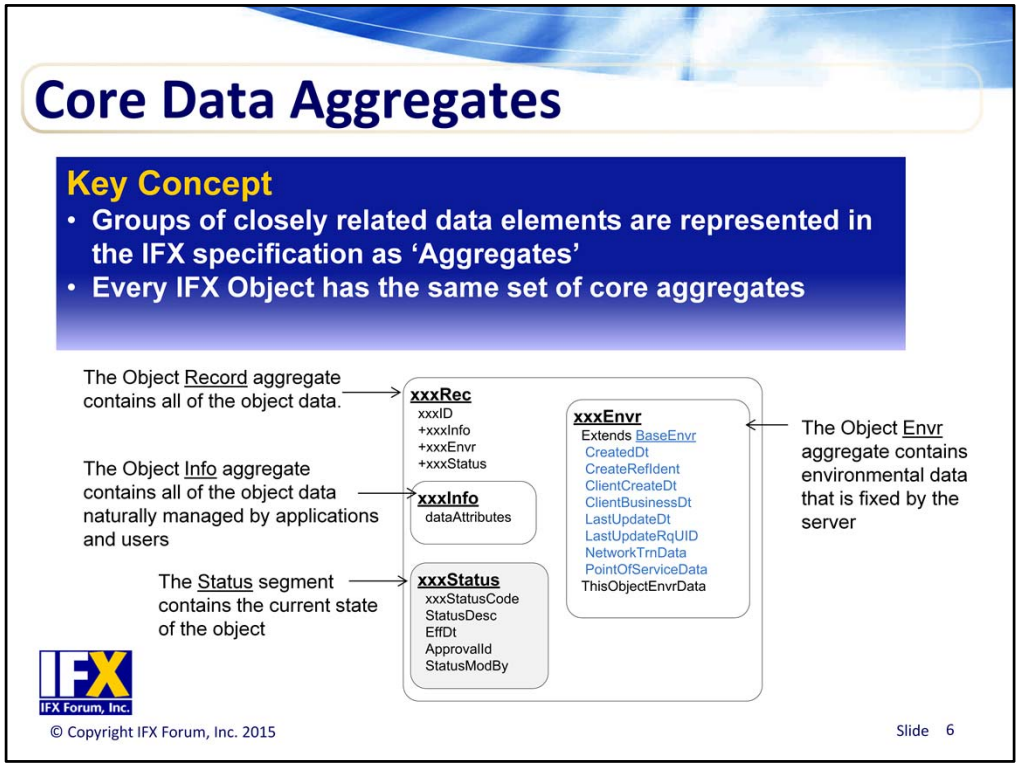


So what is an IFX Object? Essentially, an IFX object is **an implied record structure** that represents a set of well defined data usually stored as a unit on a server with persistent storage. **It is <u>not</u> assumed that the object is stored exactly as shown in the object model.**

IFX defines the data contents necessary to represent the object, but IFX does not define the physical format, actual record structure, storage technology, etc.

This is an important point, because it is at this juncture that you may recognize that you can and must map your own physical data structures to the IFX object model in order to ensure smooth implementation of the standard.

The IFX assumption, therefore, is that **a server is able to assemble a proper representation of the object.** This capability is necessary of course in order to respond appropriately to all messages about an object.



Groups of closely related data elements are represented in the specification as 'Aggregates'

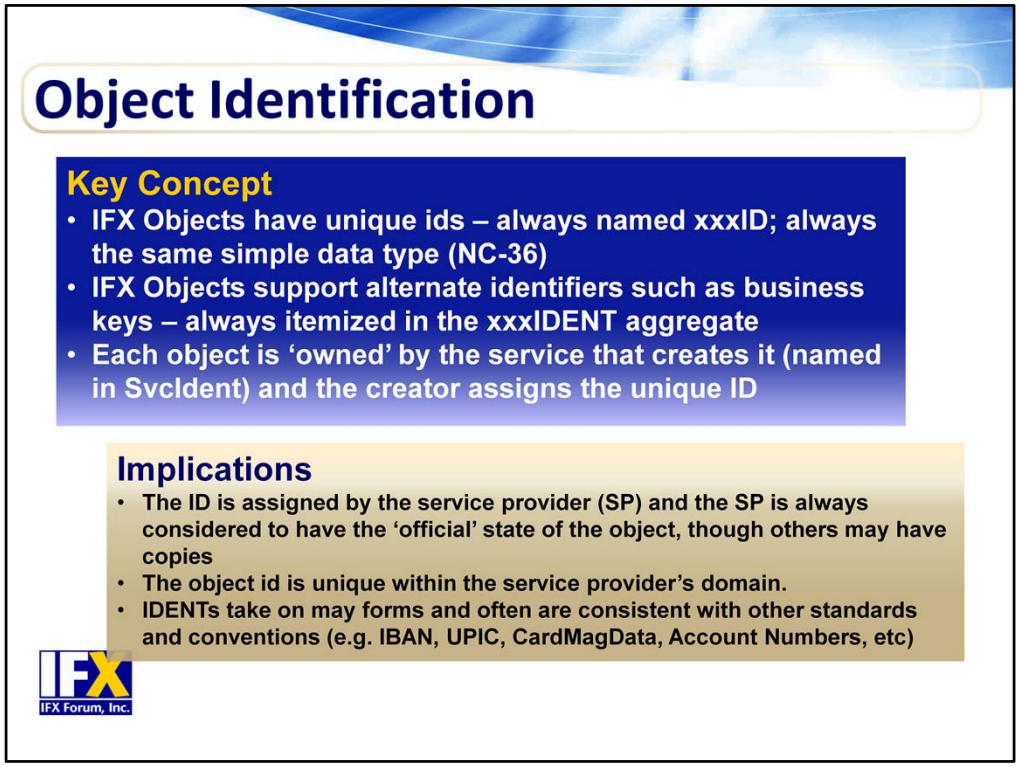
The core aggregates of every IFX object include:

•The **Info** segment which contains the properties of the object that are directly maintainable by a client

•The **Envr** segment which contains environmental data that is fixed by the server

•The **Status** segment which contains the current state of the object.

There are other important core aggregate elements as well, but understanding this set of **mandatory** aggregates is sufficient for now.



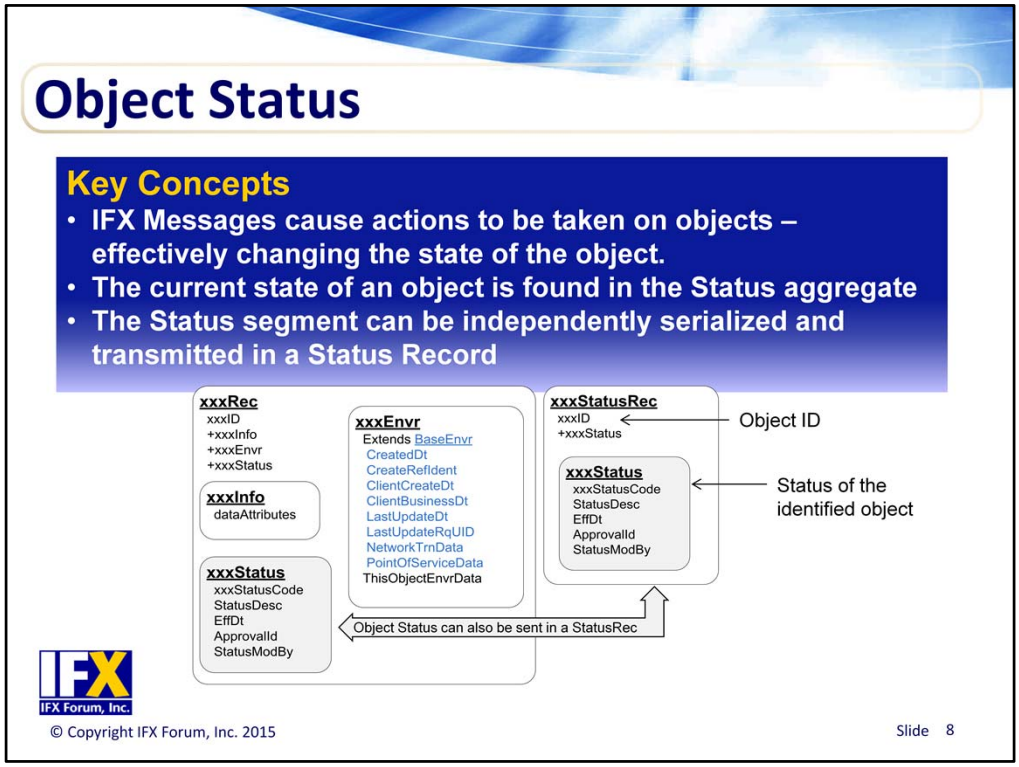
IFX assumes that each object is uniquely identified by its owner. In the IFX Framework, the **service** that creates an object **assigns the object ID** and is considered the '**owner**' of the object.

As a practical matter, this may simply mean that the Service Provider uses an arbitrary database key as the object ID. But the model supports more interesting use-cases as well. For example, a financial institution may have subsidiaries with specialized service capabilities, or rely upon outsourcing vendors for services; or specific servers within a bank infrastructure can be identified as Service Providers.

The SvcIdent, which acts as a qualifier for IDs, also ensures that messages can be routed back to the owner to ascertain an object's status and that the context for the unique id is clear.

Finally, any object may have alternate keys such as account numbers, card ids and such. IFX supports and identifies these as optional attributes of the object's IDENT structure. As the name implies, these are identifying attributes but they are distinct from the ID assigned by the service that created the object.

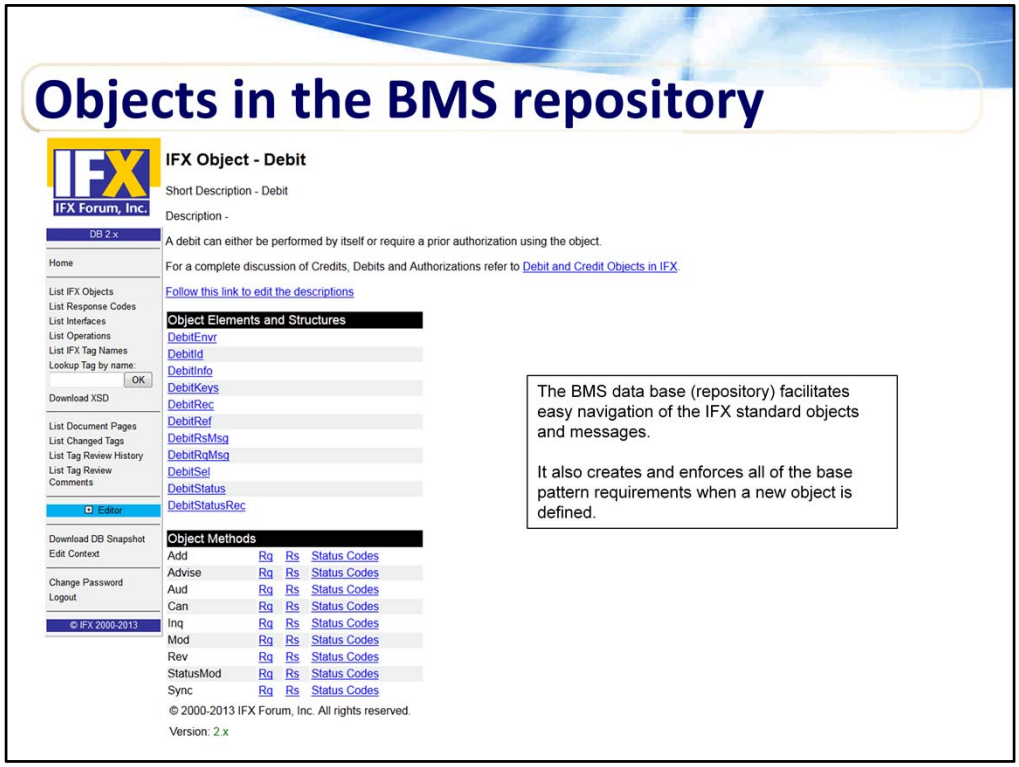
One final note, there is actually no prohibition against using a business key such as account number as the object id; the only requirement is that the SP can ensure it is entirely unique within its domain.



Every persistent object in IFX is required to maintain Status.

Importantly, the standard supports direct communication of an object status through the StatusRec.

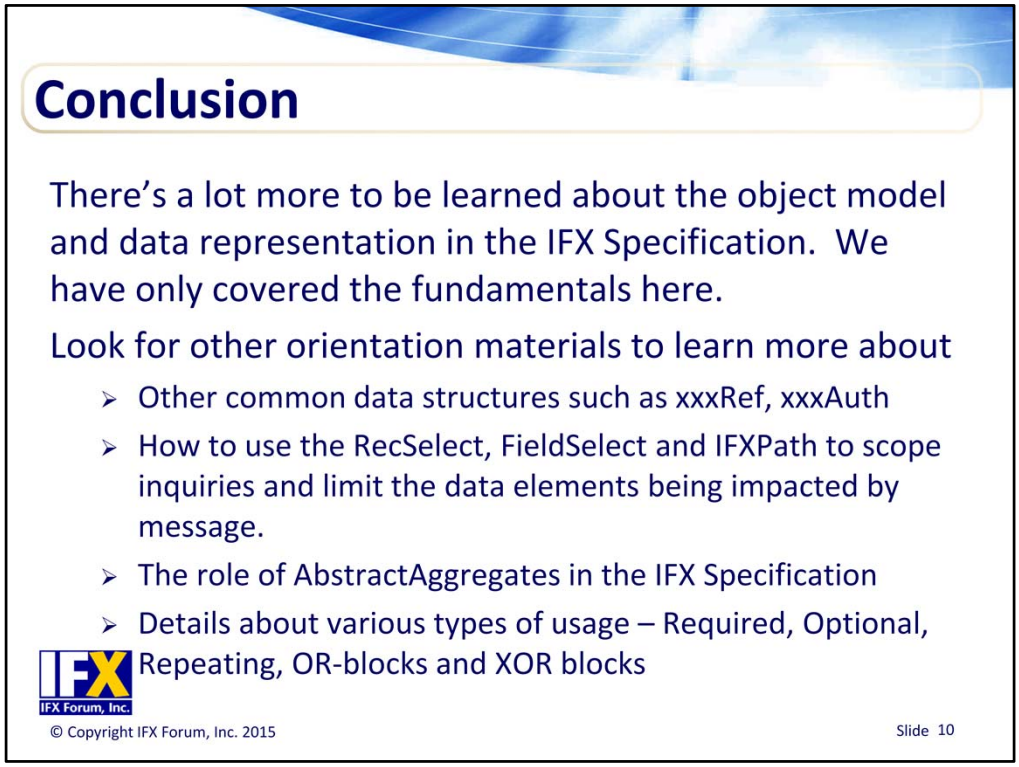
This allows direct inquiry of object status and a simple mechanism to return the status in response to such an inquiry. As shown here, it is a simple matter of including the object's Status aggregate in a StatusRec along with the object ID.



Now that we have a good understanding of the underlying concepts of IFX objects – their fundamental structure, how they represent data, how they are identified, how data processing in response to messages impacts object state, and how that state can be communicated, we can examine how the IFX BMS Data Base represents them and how to navigate the UI.

In the menu on the left side of this page, you can select “List IFX Objects” and then choose any object of interest.

Shown here is a straightforward example of an IFX Object – the Debit object. Note that in addition to itemizing the data structure this page also shows exactly which methods, or messages are implemented in standard for this object. It’s also an easy matter to navigate to any object data or related message from this point, but you are likely to find that you frequently jump to the Info segment where all of the user managed data resides.



There's a lot more to be learned about the objectmodel. We have only covered the fundamentals here.

Look for other orientation materials to learn more about these topics.

Thank you for your attention, your interest and your support of the IFX Standard.

More information can be found at these locations and at www.ifxfourm.org.