



ENOVIA SmarTeam

SmarTeam – Briefcase Programmer's Guide

This guide, issued in previous versions of SMARTEAM, is relevant in R17 for briefcase customization only.

In SMARTEAM V5R17, all BOM naming is replaced by the term “Briefcase”.

Important Notice

© Dassault Systèmes, 2004, 2008. All rights reserved.

CATIA, ENOVIA, SMARTEAM and the 3DS logo are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

PROPRIETARY RIGHTS NOTICE: This documentation is the property of Dassault Systèmes. This documentation shall be treated as confidential information and may only be used by employees or contractors of the Customer in accordance with the terms of the End-User License Agreement accepted by Customer.

Any use of the Licensed Program contained in this media or accompanying it, is subject to the terms of the End User License Agreement accepted by Customer. The Licensed Program is protected by international copyright laws and international treaties. Unauthorized use, reproduction and/or distribution of any of the Licensed Program, or any part thereof, may result in severe civil and/or criminal penalties, and will be prosecuted to the maximum extent possible under the law. Company names and product names mentioned herein are the property of their respective owners and certain portions of the Licensed Program contain elements subject to copyright owned by these entities. See the Documentation CD provided with the Licensed Program for details and/or additional terms and conditions relating to these entities.

Part Number: BOM-A1-180007

Table of Contents

1	SmartBom2 Library.....	1
	Introduction	1
	Overview of Objects – ISmBomPackage.....	1
2	SmartBomIxf Library	43
	Introduction	43
	Overview of Objects – ISmBomIxfHelper	43
	ISmProgress	44
3	SmartBriefcase Library	45
	Introduction	45
	SMARTBriefcase	45
	Overview of Objects – ISmBriefcase.....	45
	ISmBriefcaseFiles	48
	ISmConfiguration.....	49
	ISmBriefcaseProgress	50
4	SmartBomServices Library	52
	Introduction	52
	Overview of Objects – SmartBomServices.....	52
	ISmBomConfiguration	55
	ISmBomPreferences.....	56
	ISmBomPreference	56
	ISmBomGeneration.....	61
	ISmBomSynchronization	62
5	SmartBomGuiServices Library	64
	Introduction	64
	Overview of Objects – SmartBomGuiServices.....	64
	ISmBomGenerationWizard	67
	ISmBomSynchronizationWizard	69
	ISmTreeCompareDialog	70
	ISmBomConsolidationWizard	72
6	An Automotive Application	75

1 SmartBom2 Library

Introduction

The SmartBom2 Library enables you to perform the following functions:

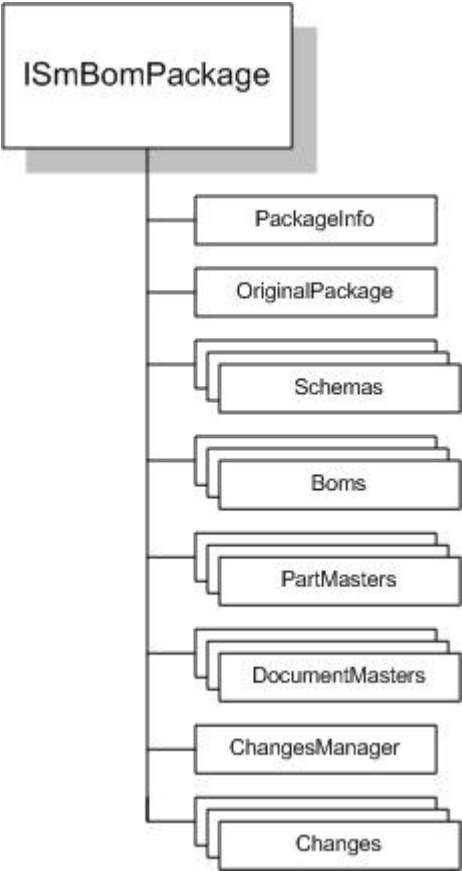
- Maintain an Assembly BOM product structure (data and schema).
- Maintain BOM referenced information (documents and additional data)
- Maintain BOM changes.

Overview of Objects – ISmBomPackage

This section presents an overview of the main SmartBom2 objects including a description of the associated objects that are useful for the programmer:

The ISmBomPackage is the highest-level object; its main purpose is to contain the other objects.

The major ISmBomPackage components are shown in the following object diagram:



ISmBomPackage Object Diagram

Properties

The ISmBomPackage object has the following properties:

Property	Description
PackageInfo	Information about the BOM Package. Returns ISmPackageInfo.
OriginalPackage	Reference to original BOM package, excluding all changes. Returns ISmBomPackage.
Schemas	Collection ISmSchemas of all available Schemas. Returns ISmSchemas.
Boms	Collection of all available BOMs. Returns ISmBoms.
PartMasters	Collection of all available Part Masters. Returns ISmPartMasters.

DocumentMasters	Collection of all available Document Masters. Returns ISmDocumentMasters.
ChangesManager	Manages all changes activity and operations Returns ISmChangesManager.
Changes	Collection of all changes made on the BOM package. Returns ISmChanges.

Methods

The ISmBomPackage object has the following methods:

Method	Description
Clone	Create a cloned BOM2 Package

Obtaining the ISmBomPackage Object

To create an ISmBomPackage object:

```
Set BomPackage = CreateObject("SmartBOM2.SmBomPackage")
```

Example

```
Dim BomPackage as ISmBomPackage
```

```
Set BomPackage = CreateObject("SmartBOM2.SmBomPackage")
```

ISmPackageInfo

The ISmPackageInfo object is a collection of properties about the current BOMPackage.

Properties

The ISmPackageInfo object has the following properties:

Property	Description
CreationDate	Date the BOMPackage was created
LastModificationDate	Date the BOMPackage was last modified.
Properties	Reference to package info properties. Returns ISmPackageInfoProperties.
UserDetails	Information about user. Returns ISmBomUserDetails.

ISmPackageInfoProperties

The ISmPackageInfoProperties object provides a reference to package info properties.

Properties

The ISmPackageInfoProperties object has the following properties:

Property	Description
Item(Index)	Retrieves item from collection according to Index, where Index can be integer or DisplayName.
Add (DisplayName, Value)	Adds custom user entry with entry name DisplayName and value Value.

Example

```
BomPackage.PackageInfo.Properties.Add( "Price", "5$" )
```

ISmBomUserDetails

The ISmBomUserDetails object represents information about the current user.

Properties

The ISmBomUserDetails object has the following properties:

Property	Description
Name	User name
E-Mail	User E-mail (SMTP)

Example

```
BomPackage.PackageInfo.UserDetails.Name = "MyName"
```

```
BomPackage.PackageInfo.UserDetails.E-Mail = "MyEmail"
```

ISmSchemas

An ISmSchemas object is a collection of ISmSchema objects. There is only one ISmSchema object in the collection and is a non-creatable object. [ISmBOMPackages and ISmParentLink are the only creatable objects]

ISmSchema

A schema is a work plan for the data in the BOMPackage and specifies the classes whose objects can exist in the package.

Properties

The ISmSchema object has the following properties:

Property	Description
SchemaURI	URI of the schema (serves as a unique namespace)
Classes	Collection ISmBomClasses of classes in the schema

Obtaining the ISmSchema Object

To obtain an ISmSchema object:

```
Set Schema = BomPackage.Schemas.Item(0)
```

ISmBomClasses

An ISmBomClasses object is a collection of ISmBomClass objects which exist in the Schema. It specifies the classes whose objects can exist in the package.

Properties

An ISmBomClasses object has the following properties:

Property	Description
Item	Accesses a class from the collection

Methods

An ISmBomClasses object has the following methods:

Method	Description
Add	Adds a new ISmBomClass object to the collection
Remove	Removes a ISmBomClass object from the collection

Adding a Class to the Schema Classes

Use the Add method of the ISmBomClasses object to add a class to the Schema Classes. The Add method returns an object of type ISmBomClass.

The Add method is called as follows:

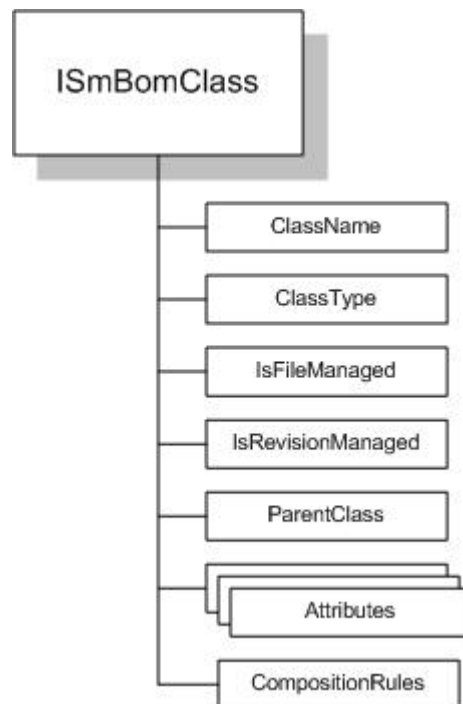
```
Set Class = Schema.Classes.Add(ClassName)
```

ISmBomClass

A class is a representation of a generic class that contains attributes and composition rules, which govern the associations between classes and general references.

A class can be either represent a Part or a, Document, General Link or Assembly Link according to the setting in the ClassType property.

A class can be associated with a Parent Class, where the class inherits the attributes of the Parent Class. The following figure shows the object diagram for the ISmBomClass Object.



ISmBomClass Object Diagram

Properties

An ISmBomClass object has the following properties:

Property	Description
ClassName	Class name
ClassType	Class type in BOM model (can be either Part, Document, AssemblyLink or GeneralLink)
IsFileManaged	Boolean. If true, a file-managed mechanism exists. Relevant only for classes of type Part or Document.
IsRevisionManaged	Boolean. If true, a revision-managed mechanism exists. Relevant only for classes of type Part or Document.
ParentClass	Reference to parent class in hierarchy. Returns ISmBomClass. If the class is a superclass, the ParentClass is “null”. Relevant only for classes of type Part or Document
Attributes	Collection ISmBomAttributes of class attributes.
CompositionRules	Information about permissible links between objects. Returns ISmBomClassCompositionRules.

Obtaining the ISmBomClass Object

To obtain an ISmBomClass object:

```
Set Class = BomPackage.Schemas[0].Classes[Index]
```

ISmBomAttributes

The ISmBomAttributes is a collection of ISmBomAttribute objects in a class definition.

Properties

An ISmBomAttributes object has the following properties:

Property	Description
Item	Accesses an attribute in the collection

Methods

An ISmBomAttributes object has the following methods:

Method	Description
Add	Adds a new ISmBomAttribute object to the collection
Remove	Removes a ISmBomAttribute object from the collection

Adding an Attribute to a Class

Use the Add method of the ISmBomAttributes object to add an attribute to the class. AttributeName can be any valid string. The Add method returns an object of type ISmBomAttribute. It is called as follows:

```
Set ISmBomAttribute = BomClass.Attributes.Add([AttributeName])
```

Once you have added the attribute you can specify the attribute definition.

ISmBomAttribute

An ISmBomAttribute is a class attribute of the ISmBomClass

The ISmBomAttribute object has the following properties:

Property	Description
Name	Attribute name
AttributeType	Attribute type
IsPrimary	Boolean – identifies primary attribute
DisplayName	Attribute display name
IsReadOnly	Read only identifier

Example

```
Set MyAttribute = MyClass.Attributes.Add("From" )  
MyAttribute.AttributeType = atString  
MyAttribute.DisplayName = "From Address"
```

ISmBOMClassCompositionRules

The ISmBomClassCompositionRules object specifies sets of classes whose objects can be linked with objects of the current class. You use ISmBomClassCompositionRules if you want to limit the objects that can be linked to an object of a class. For example, you can specify that an object of class Pen (of type Part) can be linked with a general link to an object of class Word Document (of type Document) and can be linked with a Assembly link to an object of class Ink (of type Part). Then the object of class Pen cannot be linked to any other object of other class types.

If you try to establish a link that does not conform to the rules, an exception is raised.

There are two types of rules: rules for specifying Assembly links between BOM Parts and rules for specifying general links between Parts and Documents. If no composition rules are set for a specific class, no limitations are enforced when creating links associated with this class.

The ISmBOMClassCompositionRules object has the following properties:

Property	Description
AssemblyLinks	Returns a collection of ISmBomClasses whose objects are allowed to be linked by a hierarchical part-to-part link as children to objects of the class. When this collection is empty, an object of any class can be linked by a hierarchical part-to-part link as a child to an object of the class.
GeneralLinks	Returns a collection of ISmBomClasses whose objects are allowed to be linked by a general part-to-document link to an object of the class. When this collection is empty, an object of any class can be linked by a part-to-document link to an object of the class.

Example

```
Set MyClass = Schema.Classes.Add("Folder")
MyClass.CompositionRules.GeneralLinks.Add ("Message")
MyClass.CompositionRules.AssemblyLinks.Add ("Folder")
```

Common Tasks

The following sections describe methods and properties that are used to perform common tasks related to a ISmSchema, ISmBomClass, ISmBomCompositions, ISmBomAttributes, and ISmBOMClassCompositionRules.

ISmSchema:

Creating a Schema with Classes and Class Attributes

The following procedure creates a basic schema file containing classes and class attributes.

1. Get the schema property (see Obtaining the ISmSchema Object.)
2. Create a class

```
Dim AClass As ISmBomClass

'Create an Items Class:
Set AClass = Schema.Classes.Add("Item")
'Set its attributes
AClass.ClassType = ctPart
AClass.IsFileManaged = False
'Add composition rule that allows general link between objects of
'Item class and objects of Document class
AClass.CompositionRules.GeneralLinks.Add ("Document")
'Add composition rule that allows hierarchical link between objects
'in Item class
AClass.CompositionRules.AssemblyLinks.Add ("Item")
```

3. Create a Class attribute

```
Dim AAttribute As ISmBomAttribute

' Add custom attributes to the Item class:
Set AAttribute = AClass.Attributes.Add("Manufacturer")
AAttribute.AttributeType = atString
Set AAttribute = AClass.Attributes.Add("Height")
AAttribute.AttributeType = atInt
Set AAttribute = AClass.Attributes.Add("Weight")
AAttribute.AttributeType = atInt
```

For another example of creating classes and class attributes, see Sub SetSchema(Schema As ISmSchema) in Chapter 4, A Mailbox Application.

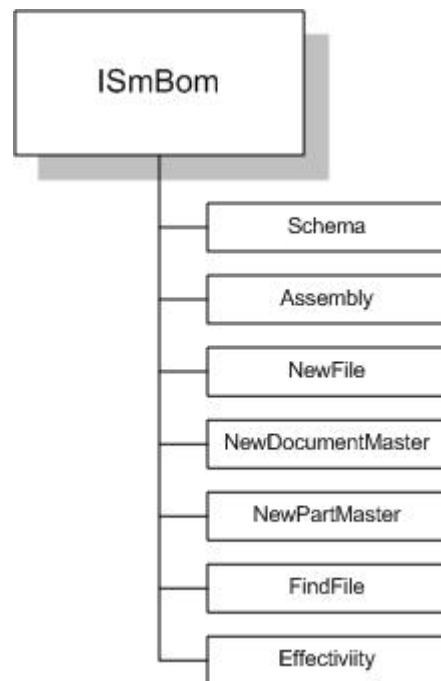
ISmBoms

A ISmBoms object is a collection of ISmBom objects. There is only one ISmBom object in the collection and it is a non-creatable object.

ISmBom

A BOM is Bill of Material structured storage for the assembly data and includes references to the schema, the root part of the assembly and provides methods for creating associated files, parts and documents.

The following figure shows the object diagram for the ISmBom Object.



ISmBom Object Diagram

Properties

An ISmBom object has the following properties:

Property	Description
Schema	Schema used for BOM. Returns ISmSchema.
Assembly	Root part of BOM assembly. Returns ISmPart.
Effectivity	Reference to effectivity data. Returns ISmEffectivity

Methods

An ISmBom object has the following methods:

Method	Description
NewFile	Adds a new ISmFile object to the BOM to be associated with a Part or a Document. Returns ISmFile.
NewDocumentMaster	Creates a new Document Master in the BOM. Returns ISmDocumentMaster
NewPartMaster	Creates a new Part Master in the BOM. Returns ISmPartMaster.
FindFile	Retrieves an existing ISmFile object from the BOM, according to the full file name. Returns ISmFile.

Obtaining the ISmBom Object

To obtain an ISmBom object:

```
Set Bom = BomPackage.Boms.Item(0)
```

ISmEffectivity

ISmEffectivity represents the Effectivity data.parameters.

Properties

An ISmEffectivity object has the following properties:

Property	Description
DateEffectivity	Returns ISmDateEffectivity

ISmDateEffectivity

ISmDateEffectivity represents the Date Effectivity data.

Properties

An ISmDateEffectivity object has the following properties:

Property	Description
StartEffectivity	Start Effectivity Data
EndEffectivity	End Effectivity Data

ISmPartMasters

An ISmPartMasters object is a collection of ISmPartMaster objects. A Part Master is an object containing Part information across multiple revisions.

Properties

An ISmPartMasters object has the following properties:

Property	Description
Item	Accesses a ISmPartMaster from the collection

Methods

An ISmPartMasters object has the following methods:

Method	Description
IndexOf	Returns the index of a ISmPartMaster in the collection.

Adding a New ISmPartMaster to the ISmPartMasters Collection

The New ISmPartMaster is added as follows:

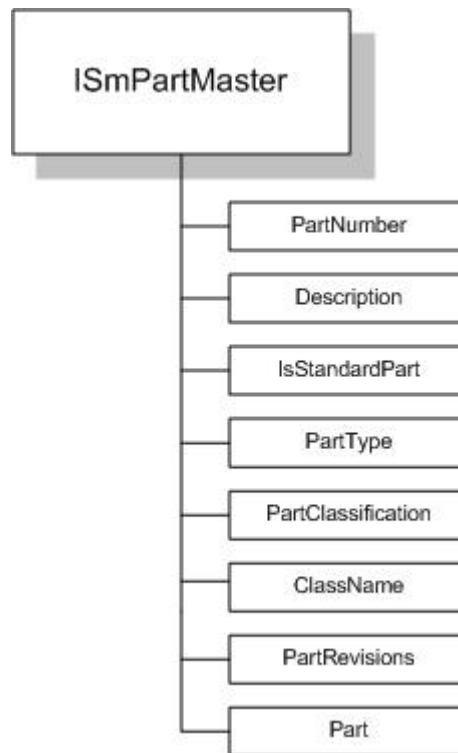
```
Set PartMaster = Boms[0].NewPartMaster(PartNumber, ClassName)
```

Where PartNumber is the part number identifier of all part revisions that belong to the newly created PartMaster and ClassName is the class to which the parts belong.

ISmPartMaster

ISmPartMaster serves as a container for all revisions of a certain part. In addition, it holds the common properties for all part revisions. A ISmPartMaster is characterized by a Part Number and a class name.

The following figure shows the object diagram for the ISmPartMaster Object.



ISmPartMaster Object Diagram

Properties

An ISmPartMaster object has the following properties:

Property	Description
PartNumber	Part number. Unique representation of Part in BOM.
Description	Part Description string
IsStandardPart	Boolean. True if Standard Part
PartType	Part Type. Can be: <ul style="list-style-type: none">• ptAssembly -- a root part• ptDetail -- a regular part
PartClassification	Part classification Not in use.
ClassName	Class Name (from Schema)
PartRevisions	Collection of all Part Revisions. Returns ISmParts Note: In the current release, only one revision of Part Master can be defined.

Part	Last available revision of part. Returns ISmPart. Note: In the current version, only one revision is available.
------	--

Methods

An ISmPartMaster object has the following methods:

Method	Description
Lock	Prevents modification of Part Master
Unlock	Allows modification of Part Master
IsLocked	Boolean. True if locked for modification.

Obtaining the ISmPartMaster Object

To obtain an ISmPartMaster object:

```
Set PartMaster = BomPackage.PartMasters[Index]
```

ISmParts

A ISmParts object is a collection of ISmPart objects.

Properties

An ISmParts object has the following properties:

Property	Description
Item	Accesses a ISmPart from the collection

Methods

An ISmParts object has the following methods:

Method	Description
IndexOf	Returns the index of a ISmPart in the collection.

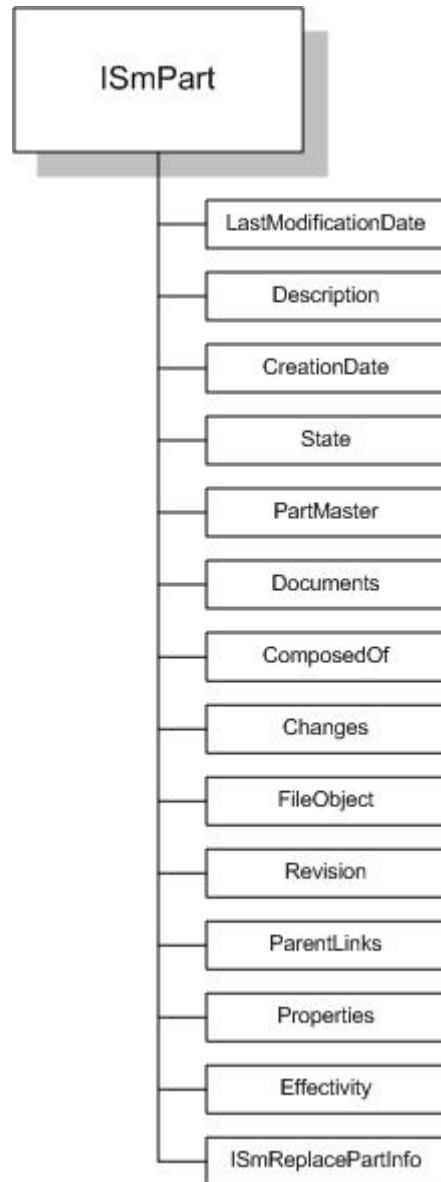
Adding a New ISmPart to the ISmParts Collection

When creating a new PartMaster object, a single ISmPart object is automatically created and associated to the Part property of the ISmPartMaster and added to the PartRevision collection of type ISmParts.

ISmPart

A ISmPart is a single revision of a Part identified by an ISmPartMaster.

The following figure shows the object diagram for the ISmPart Object.



ISmPart Object Diagram

Properties

An ISmPart object has the following properties:

Property	Description
LastModificationDate	Time and date of last modification
Description	Part description
CreationDate	Creation date
State	Current state of Part. Free text field.
PartMaster	Reference to PartMaster container. Returns ISmPartMaster
Documents	Collection of referenced documents. Returns ISmDocuments
ComposedOf	Collection of children of this Part. Returns ISmParts
Changes	Collection of all changes performed on this Part. Returns ISmChanges
FileObject	Reference to ISmFile object, if it exists. Returns ISmFile
Revision	Revision information. Returns ISmRevision
ParentLinks	Collection of ParentLinks associated with this Part. Returns ISmParentLinks
Properties	Collection of custom properties. Returns ISmProperties.
Effectivity	Effectivity data. Returns ISmEffectivity

Methods

An ISmPart object has the following methods:

Method	Description
Lock	Prevents modification of Part
Unlock	Allows modification of Part
IsLocked	Boolean. True if locked for modification.
AddChild	Add child part to part. Returns ISmParentLink
LinkDocument	Establishes a link between a document and the part. Returns one of the constants in LinkDocumentCheckEnum.
RemoveChild	Removes a child part from the part. Returns ISmParentLink
GetParentLink	Retrieves an ISmParentLink object associated with a specific ISmPart parent object.
CanAddChild	Verify the possibility of adding a specific part as a child object for this part. Checks are performed on the composition rules for the part and also for the formation of a hierarchic loop of Parts. Returns

	LinkPartCheckEnum.
ReplaceChild	Replace child Part by another Part. Returns ISmReplacePartInfo.
UnlinkDocument	Remove link to referenced document. Returns one of the constants in LinkDocumentCheckEnum
ReplaceDocument	Replace document link. Returns one of the constants in LinkDocumentCheckEnum.
AddFile	Add ISmFile object to ISmPart
ReplaceFile	Replace attached file
RemoveFile	Delete attached file reference
SetLastModificationDate	Assign file modification date

Obtaining an ISmPart Object

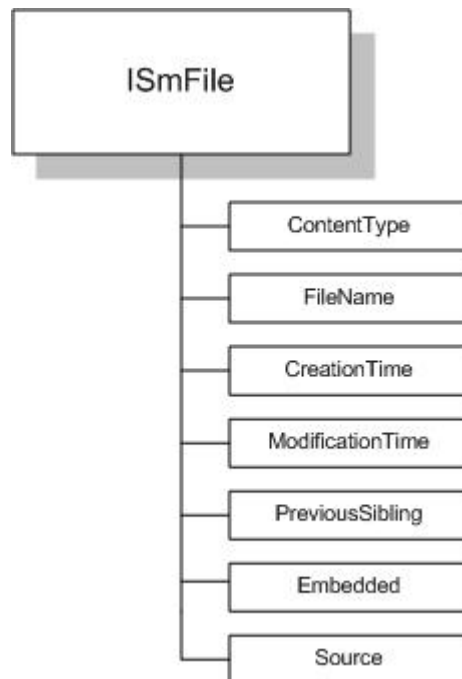
To obtain an ISmPart object:

```
Set Part = BomPackage.PartMasters[Index].PartRevisions[0]  
or  
Set Part = BomPackage.PartMasters[Index].Part
```

ISmFile

A ISmFile is an object that represents a file. ISmFile can be associated with a part or a document.

The following figure shows the object diagram for the ISmFile Object.



ISmFile Object Diagram

Properties

An ISmFile object has the following properties:

Property	Description
ContentType	Content type. Free text field.
FileName	File name. Free text field.
CreationTime	Creation date and time of the file.
ModificationTime	Modification date and time of the file.
PreviousSibling	Not used.
Embedded	If true file is embedded in the BOM package. If false, the file location is specified by a URL. Boolean flag
Source	Information about the actual location of the file. Returns ISmFileSource.

Methods

An ISmFile object has the following methods:

Method	Description
--------	-------------

SetSource	Assigns ISmFileSource to the ISmFile object.
-----------	--

ISmFileSource

ISmFileSource stores the location of an ISmFile object.

Properties

An ISmFileSource object has the following properties:

Property	Description
ComputerName	Computer Name (machine name)
Location	Location (full path)

ISmRevision

A ISmRevision represents the revision number of a Part or a Document. It has the format:

[Major revision number].[Minor revision number]

or

[Major revision number]

Properties

An ISmRevision object has the following properties:

Property	Description
Major	Major revision number
Minor	Minor revision number
Formatted	Formatted composite revision number: [Major revision number].[Minor revision number]

Methods

An ISmRevision object has the following methods:

Method	Description
Next	Generates the next revision, based on the current Major and Minor values

Obtaining the ISmRevision Object

To obtain an ISmRevision object:

```
Set Revision = BomPackage.PartMasters[Index].Parts[Index].Revision
```

ISmParentLinks

A ISmParentLinks is a collection of ISmParentLink objects.

Properties

An ISmParentLinks object has the following properties:

Property	Description
Item	Accesses a ISmParentLink from the collection by variant index.

Methods

An ISmParentLinks object has the following methods:

Method	Description
Remove	Removes an item from the collection. Note: It is not recommended to use this method directly. Instead, use the RemoveChild method in ISmPart
IndexOf	Returns an item by index.

Adding a New ISmParentLink to the ISmParentLinks Collection

To add or remove a ISmParentLink, use one of the methods AddChild, RemoveChild, or ReplaceChild, of the ISmPart object. The new, changed or removed ISmParentLink is returned by these methods. The BOM module performs checks to ensure the coherency of the data structure, and creates the appropriate changes if the track changes mode is on.

Note: It is not recommended to add or remove ISmParentLink objects from the ISmParentLinks collection directly, since no coherency checks are performed by the system.

ISmParentLink

A ParentLink is a link between an object and its parent object.

Properties

An ISmParentLink object has the following properties:

Property	Description
----------	-------------

ReferenceDesignator	Operation sequence / Reference designator. Returns ISmReferenceDesignator.
Note	Descriptive note
Parent	Reference to the parent part. Returns ISmPart
Quantity	Generic number of child Parts.. Returns ISmQuantity
Effectivity	Structural effectivity data.

Obtaining the ISmParentLink Object

To obtain an ISmParentLink object:

```
Set ParentLink = BomPackage.PartMasters[Index].Part.ParentLinks[Index]
```

Creating the ISmParentLink Object

To create an ISmParentLink object by adding ISmPart as a child to another ISmPart:

```
Set ParentLink = ParentPartMaster.Part.AddChild(ChildPartMaster.Part)  
ParentLink.Quantity = 5
```

ISmQuantity

An ISmQuantity object holds information about the quantity of child Parts associated with this link. Allows you to represent links to multiple children by a single link.

For example, a object of class Wheel is a child of an object of class Van. You can establish a hierarchic link between a single Van object and four Wheel objects using a single ParentLink with Quantity.Value = 4 and Quantity.UnitOfMeasure = Each (default).

Properties

An ISmQuantity object has the following properties:

Property	Description
UnitOfMeasure	Unit of Measure, as a String.

Value	Quantity value. Double
-------	------------------------

Obtaining the ISmQuantity Object

To obtain an ISmQuantity object:

```
Quantity = BomPackage.PartMasters[Index].Part.ParentLinks[Index].Quantity
```

ISmProperties

A ISmProperties object is a collection of ISmProperty objects. This collection is managed automatically by the BOM2, according to the attributes defined in the schema.

Methods

An ISmProperties object has the following methods:

Method	Description
Remove	Removes an item from the collection. Note: It is not recommended to use this method directly. Instead, use the Remove method in ISmBomAttributes collection.
IndexOf	Returns the index of a ISmProperty in the collection.

Obtaining the ISmProperties Object

To obtain an ISmProperties object:

```
Set Properties = BomPackage.PartMasters[Index].Part.Properties
```

ISmProperty

A Property is a single piece of data associated with an object. Properties are the actual instances of the ISmBomAttributes, as defined in the schema.

Properties

An ISmProperty object has the following properties:

Property	Description
Value	Value of the property as Variant type.
Attribute	Reference to the corresponding ISmBomAttribute object in the schema.

Obtaining the ISmProperty Object

To obtain an ISmProperty object:

```
Set Property = BomPackage.PartMasters[0].Parts[Index].Properties[Index]
```

Example

```
Set Properties = BomPackage.PartMasters[Index].Part.Properties
```

```
Set Property = Properties[Properties.IndexOf("Subject")]
```

```
Property.Value = "Hello"
```

ISmReplacePartInfo

ISmReplacePartInfo contains information about the results of the ISmPart.ReplaceChild method.

Properties

An ISmReplacePartInfo object has the following properties:

Property	Description
ReplacedPart	Replaced part. Returns ISmPart
DeletedLink	Deleted link. ISmParentLink
NewPart	New part (replaced old part). Returns ISmPart
NewParentLink	New parent link object. Returns ISmParentLink

ISmReferenceDesignator

ISmReferenceDesignator contains information about Reference Designator values, specially used in the Electronics industry. This array of string values is associated to the link according to the quantity.location information about each Part as an array of string values. Each array item represents an instance of a Part according to the number of items reflected in the Quantity property in BOM.

Properties

An ISmReferenceDesignator object has the following properties:

Property	Description
Original	The original Reference Designator string, as originally typed by the user
Items	Returns ISmReferenceDesignatorItems

ISmReferenceDesignatorItems

ISmReferenceDesignatorItems contains the collection of ISmReferenceDesignator Items.

Properties

An ISmReferenceDesignatorItem object has the following properties:

Property	Description
Item	An individual ISmReferenceDesignator Item

Methods

An ISmReferenceDesignatorItem object has the following methods:

Method	Description
Add	Adds an ISmReferenceDesignator Item to the collection
Clear	Removes an ISmReferenceDesignator Item from the collection

Common Tasks

The following sections describe methods and properties that are used to perform common tasks related to a ISmPart.

ISmPart:

Adding a New Part Master

The following example shows how to create a root part and a regular part and attach them to the BOM.

```
Dim ParentFolder As ISmPartMaster
Dim Folder As ISmPartMaster
Dim Quantity As ISmQuantity

' Create a parent Mailbox folder and populate it with subfolders
' Create PartMaster "Mailbox" with PartNumber (unique identifier)
' "Mailbox" and PartClassName "Folder"
Set ParentFolder = BOM.NewPartMaster("Mailbox", "Folder")
ParentFolder.Description = "Mailbox"
' Define ParentFolder as an Assembly part (root part)
ParentFolder.PartType = ptAssembly
' Define latest (and only) ParentFolder.Part as the BOM assembly
```

```
BOM.SetBomAssembly ParentFolder.Part

' Add an Inbox folder as a regular part
Set Folder = BOM.NewPartMaster("Inbox", "Folder")
Folder.PartType = ptDetail
Folder.Description = "Inbox folder"
' Define Folder "Inbox" as child of ParentFolder "Mailbox" and
' set Quantity
Set Quantity = ParentFolder.Part.AddChild(Folder.Part)
Quantity.Value = 1
```

ISmFile:

Creating a New ISmFile Object and assigning it to a part

```
Set File = BOMPackage.Boms.Item(0).NewFile
File.ContentType = "Text"
Part.AddFile(File)
```

ISmFile:

Obtaining an Existing ISmFile Object

There is no accessible collection for ISmFile objects. Instead, each file can be reached through the Part or Document with which this file is associated.

```
File = Part.FileObject
Or
File = Document.FileObject
```

Retrieve an existing file by specifying its location in the method FindFile of ISmBom:

```
Set File = BomPackage.Boms.Item(0).FindFile(LogicalFileName)
```

ISmDocumentMasters

A ISmDocumentMasters object is a collection of ISmDocumentMaster objects.

Properties

An ISmDocumentMasters object has the following properties:

Property	Description
Item	Accesses a ISmDocumentMaster from the collection by index or unique Document Id

Adding a New ISmDocumentMaster to the ISmDocumentMasters collection

```
Set DocumentMaster = Boms[0].NewDocumentMaster(PartNumber, ClassName).
```

where PartNumber is the part number of all Document revisions that belong to the newly created DocumentMaster and ClassName is the class to which the Documents belong.

Example

```
Set MyDocumentMaster =  
BomPackage.Boms.Item[0].NewDocumentMaster("Specification", "MS Word  
Document")
```

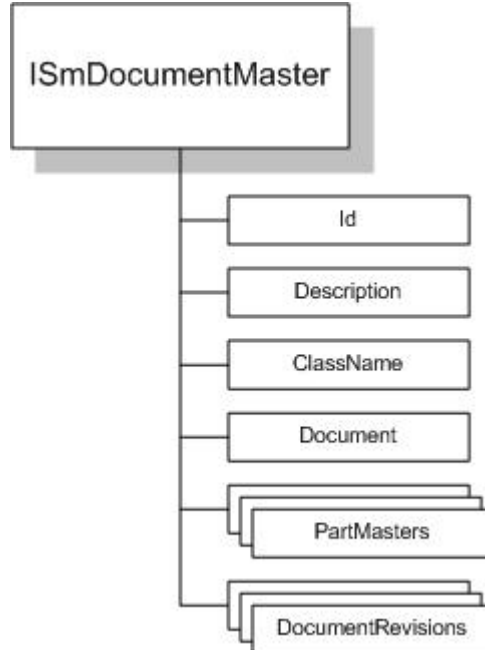
```
MyDocumentMaster.Description = "Full production specification"
```

```
PartMaster.Part.LinkDocument(MyDocumentMaster.Document)
```

ISmDocumentMaster

The ISmDocumentMaster object serves as a container for all revisions of a document. It contains the common properties for all revisions of the document.

The following figure shows the object diagram for the ISmDocumentMaster object.



ISmDocumentMaster Object Diagram

Properties

An ISmDocumentMaster object has the following properties:

Property	Description
Id	Unique Id in BOM
Description	Description of document
ClassName	Class name from schema
PartMasters	Collection of referenced Part Masters. Returns ISmPartMasters.
DocumentRevisions	Collection of all revisions of Document Master. Returns ISmDocuments
Document	Last available revision of the document. Returns ISmDocument.

Methods

An ISmDocumentMaster object has the following methods:

Method	Description
Lock	Prevents modification of Document Master
Unlock	Allows modification of Document Master
IsLocked	Boolean. True if locked for modification.

Obtaining the ISmDocumentMaster Object

To obtain an ISmDocumentMaster object:

```
Set DocumentMaster = BOMPackage.DocumentMasters[Index]
```

ISmDocuments

A ISmDocuments object is a collection of ISmDocument objects.

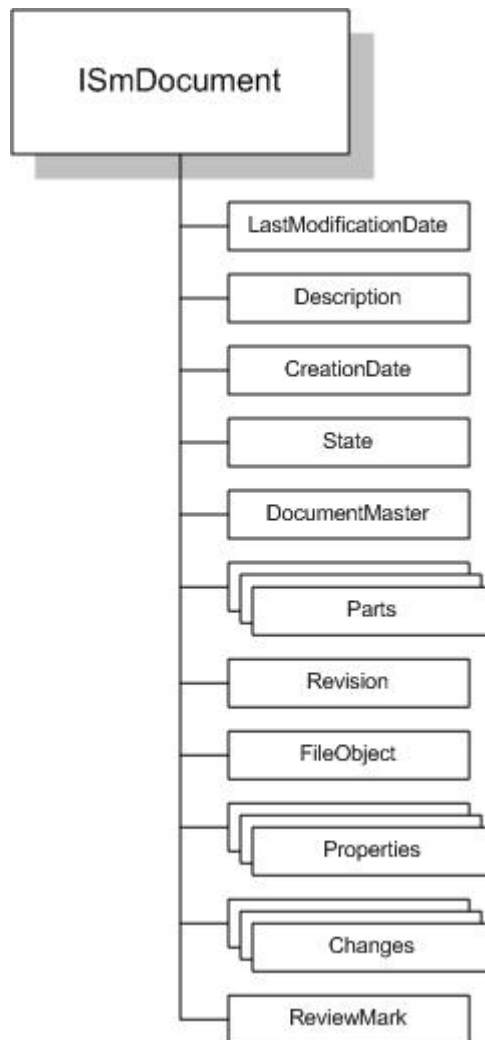
Adding a New ISmDocument to the ISmDocuments collection

When creating a new DocumentMaster object, an ISmDocument object is automatically created and associated to the Document property of the ISmDocumentMaster.

ISmDocument

A ISmDocument is a single revision of an ISmDocumentMaster.

The following figure shows the object diagram for the ISmDocument Object.



ISmDocument Object Diagram

Properties

An ISmDocument object has the following properties:

Property	Description
LastModificationDate	Date of last modification
Description	Document description
CreationDate	Date of creation
State	Current state of Document. Free text field.

DocumentMaster:	Reference to Document Master. Returns ISmDocumentMaster.
Parts	Collection of referenced parts. Returns ISmParts.
Revision	Revision information. Returns ISmRevision
FileObject	Reference to a file object. Returns ISmFile
Properties	Collection of custom properties. Returns ISmProperties
Changes	Collection of changes made on document. Returns ISmChanges
ReviewMark	Information about a possible requirement to review/change the document later on. Returns ISmDocumentReviewMark

Methods

An ISmDocument object has the following methods:

Method	Description
Lock	Prevents modification of Document
Unlock	Allows modification of Document
IsLocked	Boolean. True if locked for modification.
SetLastModificationDate	Assigns a modification date

Obtaining the ISmDocument Object

To obtain an ISmDocument object:

```
Set Document = BOMPackages.DocumentMasters[Index].Document
```

ISmDocumentReviewMark

ISmDocumentReviewMark contains information about a possible requirement to review/change the document later on

Properties

An ISmDocumentReviewMark object has the following properties:

Property	Description
IsMarked	Boolean flag to determine if Document has to be reviewed/changed in the future
UserName	User name to make the actual review/change
Note	Notes about the review/change task, to be done later on
Date	Date of mark

Common Tasks

The following sections describe methods and properties that are used to perform common tasks related to a ISmDocument.

ISmDocument: Adding a New Document Master

The following procedure creates a DocumentMaster for the BOM.

1. Get the BOM property (see Obtaining the ISmBom Object.)
2. Create a DocumentMaster

```
Dim aDocumentMaster As ISmDocumentMaster
' Define a new DocumentMaster of class Document with unique
' name DocumentId
Set aDocumentMaster = Bom.NewDocumentMaster(DocumentId, "Document")
```

ISmDocument: Associating a File with a Document

The following procedure associates a File object with a Document in the BOM.

```
' Define ISmFile object corresponding to attached file AttachedFile
' If computer is local use null, FileName and FileLocation are
AttachedFile
Set File = Bom.NewFile("", DocumentId, AttachedFile)
File.ContentType = FileContent
' Define File as embedded in package
File.Embedded = True
' Associate the file with the Document object
aDocumentMaster.Document.FileObject = File
```

ISmDocument: Linking a Document to a Part

```
' Attach the document to a part by general link
Part.LinkDocument(aDocumentMaster.Document)
```

ISmChangesManager

A ISmChangesManager manages all changes activity and operations.

Tracks all part/documentIt tracks all changes and has the ability to undo/redo change operations.

Change is any change performed on the BOM such as adding a child part, changing a property value, and associating a file. Changes are accumulated automatically by the Change Manager, if the `IsTrackingChanges` flag is set true.

You can “run”“implement” the accumulated changes to reproduce a BOM package.

You can instruct the Changes Manager to accumulate changes in named groups. This facility can be used, for example, to cancel a set of operations if one of them fails.

The Change object is created and modified by the Change Manager; the user can review the Change objects.

Properties

An `ISmChangesManager` object has the following properties:

Property	Description
<code>IsTrackingChanges</code>	Boolean. True if changes are being tracked

Methods

An `ISmChangesManager` object has the following methods:

Method	Description
<code>UndoLastChange</code>	Undo last managed change
<code>ClearChanges</code>	Clear changes
<code>StartTrackingChanges</code>	Enable changes tracking
<code>StopTrackingChanges</code>	Disable changes tracking

Obtaining the `ISmChangesManager` Object

To obtain an `ISmChangesManager` object:

```
Set ChangesManager = BomPackage.ChangesManager
```

Example

```
ChangesManager.StartTrackingChanges
```

```
Call ParentFolder.Part.AddChild(Folder.Part)
```

```
ChangesManager.EndTrackingChanges
```

`ISmChanges`

A `ISmChanges` object is a collection of `ISmChange` objects.

Properties

An ISmChanges object has the following properties:

Property	Description
Item	Accesses a ISmChange object from the collection

Methods

An ISmChanges object has the following methods:

Method	Description
IndexOf	Returns the index of a specific ISmChange object in the collection.

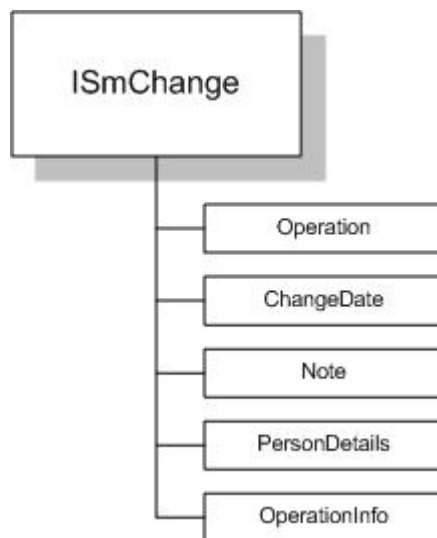
Adding a New ISmChange to the ISmChanges collection

ISmChange objects are created and handled automatically by the BOM2 when the Tracking Changes Mode is turned on. ISmChanges collection can hold a list of changes per object (Part or Document), or for the whole package.

ISmChange

A ISmChange describes an individual change.

The following figure shows the object diagram for the ISmChange Object.



ISmChange Object Diagram

Properties

An ISmChange object has the following properties:

Property	Description
Operation	Operation associated with the change. Returns one of the constants of SmChangeOperationEnum.
ChangeDate	Change Date.
Note	Free text note.
PersonDetails	Information of the user that made the change. Returns ISmBomUserDetails
OperationInfo	Detailed information about the change. Returns ISmChangeOperationInfo

Obtaining the ISmChange Object

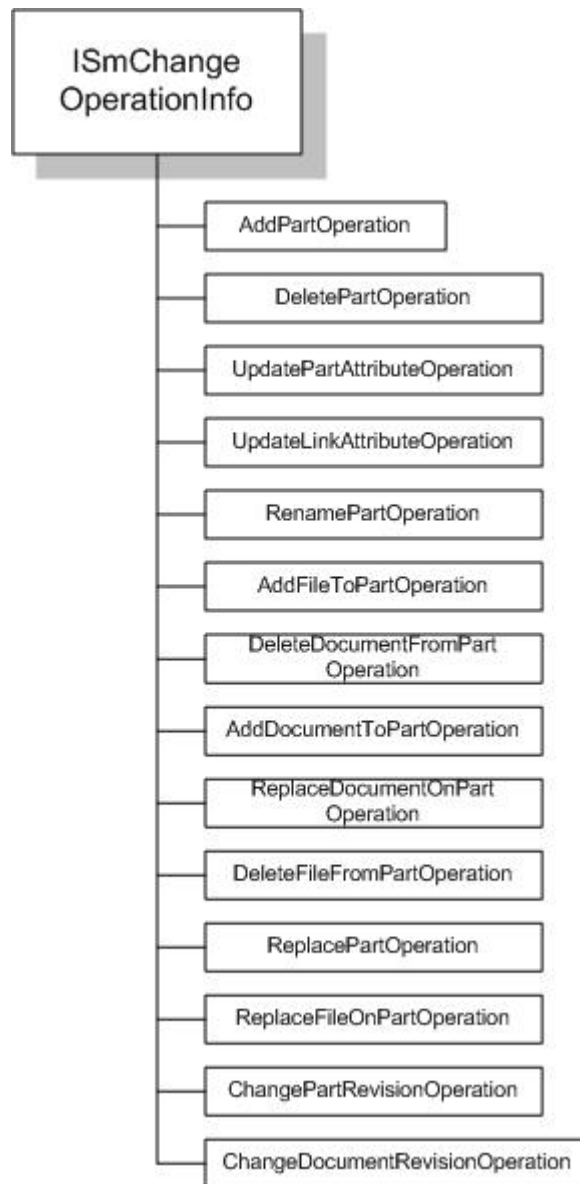
To obtain an ISmChange object:

```
Set Change = BomPackage.Changes[Index]
```

ISmChangeOperationInfo

An ISmChangeOperationInfo describes an individual change. Although ISmChangeOperationInfo has a property for each kind of change operation, only one of them points to a valid operation info object, according to the change that was made. All other properties point to NULL.

The following figure shows the object diagram for the ISmChangeOperationInfo Object.



ISmChangeOperationInfo Object Diagram

Properties

An ISmChangeOperationInfo object has the following properties:

Property	Description
----------	-------------

AddPartOperation	Returns ISmAddPartInfo
DeletePartOperation	Returns ISmDeletePartInfo
RenamePartOperation	Returns ISmRenamePartInfo
ReplacePartOperation	Returns ISmReplacePartInfo
UpdatePartAttributeOperation	Returns ISmUpdatePartAttributeInfo
UpdateLinkAttributeOperation	Returns ISmUpdateLinkAttributeInfo
AddFileToPartOperation	Returns ISmAddFileToPartInfo
DeleteFileFromPartOperation	Returns ISmDeleteFileFromPartInfo
ReplaceFileOnPartOperation	Returns ISmReplaceFileOnPartInfo
AddDocumentToPartOperation	Returns ISmAddDocumentToPartInfo
DeleteDocumentFromPartOperation	Returns ISmDeleteDocumentFromPartInfo
ReplaceDocumentOnPartOperation	Returns ISmReplaceDocumentOnPartInfo
ChangePartRevisionOperation:	Returns ISmChangePartRevisionInfo
ChangeDocumentRevisionOperation	Returns ISmChangeDocumentRevisionInfo

Obtaining the ISmChangeOperationInfo Object

To obtain an ISmChangeOperationInfo object:

```
Set ChangeOperationInfo = BomPackage.Changes[Index].OperationInfo
```

Example

```
Set ChangeOperationInfo = BomPackage.Changes[Index].OperationInfo
Set Change = BomPackage.Changes[Index]
If Change.Operation = coAddPart then
MsgBox ChangeOperationInfo.AddPartOperation.Part.Description + " has been
added as a child part to " +
ChangeOperationInfo.AddPartOperation.ParentLink.Parent.Description
End If
```

ISmAddPartInfo

A ISmAddPartInfo contains information about an Add Part change.

Properties

An ISmAddPartInfo object has the following properties:

Property	Description
Part	Added part. Returns ISmPart.
ParentLink	Reference to link between new part and parent part. Returns ISmParentLink.

ISmDeletePartInfo

A ISmDeletePartInfo contains information about an Delete Part change.

Properties

An ISmDeletePartInfo object has the following properties:

Property	Description
Part	Deleted part. Returns ISmPart.
DeletedLink	Information about deleted link between parts. Returns ISmParentLink.

ISmUpdatePartAttributeInfo

A ISmUpdatePartAttributeInfo contains information about an Update Part Attribute change.

Properties

An ISmUpdatePartAttributeInfo object has the following properties:

Property	Description
AttributeName	Updated attribute
OldValue	Old value of attribute
NewValue	New value of attribute
Part	Updated part. Returns ISmPart.

ISmRenamePartInfo

A ISmRenamePartInfo contains information about an Rename Part change.

Properties

An ISmRenamePartInfo object has the following properties:

Property	Description
OldPartNumber	Previous part number
Part	Part reference. Returns ISmPart.

ISmReplacePartInfo

A ISmReplacePartInfo contains information about an Replace Part change.

Properties

An ISmReplacePartInfo object has the following properties:

Property	Description
ReplacedPart	Replaced part. Returns ISmPart.
DeletedLink	Deleted link. Returns ISmParentLink.
NewPart	New part (replaced old part) . Returns ISmPart.
NewParentLink	New parent link object. Returns ISmParentLink.

ISmUpdateLinkAttributeInfo

A ISmUpdateLinkAttributeInfo contains information about an Update Link Attribute change.

Properties

An ISmUpdateLinkAttributeInfo object has the following properties:

Property	Description
AttributeName	Link attribute name
OldValue	Old value
NewValue	New value of attribute
Part	Part object. Returns ISmPart.
Link	Link object. Returns ISmParentLink.

ISmAddFileToPartInfo

A ISmAddFileToPartInfo contains information about an Add File To Part change.

Properties

An ISmAddFileToPartInfo object has the following properties:

Property	Description
Part	Part object. Returns ISmPart
AddedFile	Added file object reference. Returns ISmFile

ISmDeleteFileFromPartInfo

A ISmDeleteFileFromPartInfo contains information about an Delete File From Part change.

Properties

An ISmDeleteFileFromPartInfo object has the following properties:

Property	Description
Part	Part object reference. Returns ISmPart .
DeletedFile	Deleted file object reference. Returns ISmFile.

ISmReplaceFileOnPartInfo

A ISmReplaceFileOnPartInfo contains information about an Replace File On Part change.

Properties

An ISmReplaceFileOnPartInfo object has the following properties:

Property	Description
Part	Part object. Returns ISmPart.
OldFile	Old file object reference. Returns ISmFile.
NewFile	New file object reference. Returns ISmFile.

ISmAddDocumentToPartInfo

A ISmAddDocumentToPartInfo contains information about an Add Document To Part change.

Properties

An ISmAddDocumentToPartInfo object has the following properties:

Property	Description
Part	Part object. Returns ISmPart.
Document	Document object. Returns ISmDocument.

ISmDeleteDocumentFromPartInfo

A ISmDeleteDocumentFromPartInfo contains information about a Delete Document From Part change.

Properties

An ISmDeleteDocumentFromPartInfo object has the following properties:

Property	Description
Part	Part object. Returns ISmPart.
Document	Document object. Returns ISmDocument.

ISmReplaceDocumentOnPartInfo

A ISmReplaceDocumentOnPartInfo contains information about an Replace Document On Part change.

Properties

An ISmReplaceDocumentOnPartInfo object has the following properties:

Property	Description
Part	Part object reference. Returns ISmPart.
NewDocument	New document reference. Returns ISmDocument.
OldDocument	Old document reference. Returns ISmDocument.

ISmChangePartRevisionInfo

A ISmChangePartRevisionInfo contains information about an ChangePartRevisionReplace change.

Properties

An ISmReplaceDocumentOnPartInfo object has the following properties:

Property	Description
OldValue	The old revision value of the ISmPart.
Part	Reference to the ISmPart object whose revision has been changed.

ISmChangeDocumentRevisionInfo

A ISmChangeDocumentRevisionInfo contains information about an ChangeDocumentRevision change.

Properties

An ISmReplaceDocumentOnPartInfo object has the following properties:

Property	Description
OldValue	The old revision value of the ISmDocument.
Document	Reference to the ISmDocument object whose revision has been changed.

Common Tasks

The following sections describe methods and properties that are used to perform common tasks related to a ISmChangeManager and ISmChanges.

ISmChangeManager: Controlling Changes Recording

```
' Start tracking changes
Package.ChangesManager.StartTrackingChanges

' Attach message to the folder by general link
Call Part.LinkDocument(aDocumentMaster.Document)

' Stop tracking changes
Package.ChangesManager.StopTrackingChanges
```

**ISmChangeManager:
Controlling Changes Recording**

Displaying Changes on BOM Package

See DisplayChanges method in Chapter 6, An Automotive Application

2 SmartBomIxf Library

Introduction

The SmartBomIxf Library enables you to perform the following functions:

- Reading BOM Packages from IXF files.

- Writing BOM Packages into IXF files.

For additional information, see SmartIxf documentation.

Overview of Objects – ISmBomIxfHelper

This section presents an overview of the main SmartBomIxf objects including a description of the associated objects that are useful for the programmer:

The ISmBomIxfHelper is the highest level object; its main purpose is to contain the other objects. This is the only creatable object in this library.

Properties

The ISmBomIxfHelper object has the following properties:

Property	Description
Progress	Reference to the ISmProgress interface.

Methods

The ISmBomIxfHelper object has the following methods:

Method	Description
BomPackageFromIxf	Reads an Ixf file into a BOM package
BomPackageFromIxfStream	Reads an Ixf stream into a BOM package – N/A
IxfFromBomPackage	Writes a BOM package to an Ixf file

Obtaining the ISmBomIxfHelper Object

To create an ISmBomIxfHelper object:

Example

```
Set BomIxf = CreateObject("SmartBomIxf.SmartBomIxfHelper")  
(Saving a package)  
BomIxf.IxfFromBomPackage Package, "C:\MyDocuments\MyBomPackage.stbom"  
(Loading a package)  
BomIxf.BomPackageFromIxf "C:\ MyDocuments\MyBomPackage.stbom", "c:\temp,  
Package
```

ISmProgress

An ISmProgress interface is an abstract interface based on a callback function for establishing a progress bar for saving and loading a BOM package. See also ISmBriefcaseProgress.

This interface is optional. To use the interface, you need to define a class that implements the interface and pass the class to the progress property of the ISmBOMIxfHelper.

Methods

An ISmProgress object has the following methods:

Method	Description
SetValue (Value)	Called by SmartBomIxf to report on significant progress. Value is an integer between 0 and 100, which represents the percentage of progress reported relative to the entire task.

3 SmartBriefcase Library

Introduction

The **SmartBriefcase Library** enables you to perform the following functions:

- Creating and managing a self-executable zip file (Briefcase)
- Altering an existing Briefcase
- Setting configuration variables for a Briefcase

SMARTBriefcase

SMARTBriefcase allows you to create a self-executable zip file containing applications and compressed or uncompressed data files together with a Briefcase stub (automatically embedded in the Briefcase) for executing the applications.

Overview of Objects – ISmBriefcase

This section presents an overview of the main SmartBriefcase objects including a description of the associated objects that are useful for the programmer.

The ISmBriefcase is the highest-level object; its main purpose is to contain the other objects. This is the only creatable object in this library.

The major ISmBriefcase components are shown in the following object diagram:

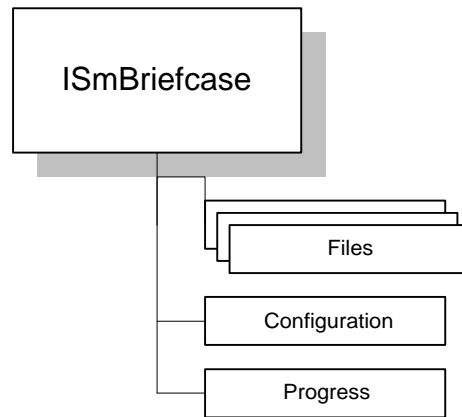


Figure 3-1 ISmBriefcase Object Diagram

Properties

The ISmBriefcase object has the following properties:

Property	Description
Files	Collection of Briefcase files. Returns ISmBriefcaseFiles.
Configuration	Stores the configuration strings. Returns ISmConfiguration.
Progress	Reference to a class that implements the ISmProgress interface. You would use this property only if you want to create and activate a progress bar; the default is null. Returns ISmBriefcaseProgress.

Methods

The ISmBriefcase object has the following methods:

Method	Description
New	Creates a new Briefcase object. The user must call the New method before calling any other Briefcase method. You must call Save to save the new Briefcase to a file.
Open	Open an existing Briefcase file
Save	Save the Briefcase object into a Briefcase file

Obtaining the ISmBriefcase Object

To create an ISmBriefcase object from the coclass CoSmBriefcase:

```
Set Briefcase = CoSmBriefcase.Create
```

ISmBriefcaseFiles

An ISmBriefcaseFiles object is a collection of ISmBriefcaseFile objects. It specifies the files whose objects can exist in the ISmBriefcase.

Properties

An ISmBriefcaseFiles object has the following properties:

Property	Description
IndexOf	Returns the index of a ISmBriefcaseFile object in the collection
Item	Accesses an ISmBriefcaseFile object from the collection by index. Returns ISmBriefcaseFile.

Methods

An ISmBriefcaseFiles object has the following methods:

Method	Description
Add	Adds a new ISmBriefcaseFile object to the collection. Returns ISmBriefcaseFile
Remove	Removes a ISmBriefcaseFile object from the collection
Clear	Delete all ISmBriefcaseFile objects from the collection
ItemByIndex	Returns ISmBriefcaseFile by its index.

Adding a BriefcaseFile to the BriefcaseFiles Collection

Use the Add method of the ISmBriefcaseFiles object to add a BriefcaseFile to the collection. The Add method returns an object of type ISmBriefcaseFile.

The Add method is called as follows:

```
Set BriefcaseFile = Briefcase.BriefcaseFiles.Add
```

Example

For an example of how to add a BriefcaseFile to the Briefcase, see Common Tasks.

ISmBriefcaseFile

A BriefcaseFile represents an individual file in the briefcase.

Properties

An ISmBriefcaseFile object has the following properties:

Property	Description
SourceFileName	Original location of File
DestinationFileName	File location in destination target
IsCompressed	Whether to compress the ISmBriefcaseFile, or not.

Methods

An ISmBriefcaseFile object has the following methods:

Method	Description
Extract (DestinationFileName)	Extract the file to its destination location. If the DestinationFileName argument is empty, the file is extracted to the location denoted by the property DestinationFileName.

Obtaining the ISmBriefcaseFile Object

To obtain an ISmBriefcaseFile object:

```
Set BriefcaseFile = Briefcase.BriefcaseFiles[Index]
```

ISmConfiguration

The ISmConfiguration object is a collection of configuration properties of the current Briefcase. These configurable parameters provide ISmBriefcase with the flexibility to operate in typical scenarios, such as installing and uninstalling the content of the Briefcase and opening the Briefcase.

Properties

The ISmConfiguration object has the following properties:

Property	Description
InstallCommand	The Install command string that is passed to BriefcaseStub
UnInstallCommand	The uninstall command string that is passed to BriefcaseStub
ExecuteCommand	The execute command string that is passed to BriefcaseStub
PackageId	The package identifier. A unique string for identifying the package. Can be a URI namespace, GUID, or any other unique string.
Description	The package description

ISmBriefcaseProgress

The ISmBriefcaseProgress object represents information about the current user. This interface is not implemented in this library; the user can implement a class that supports this interface and assign the ISmBriefcase property Progress to the class. You would use this property only if you want to create and activate a progress bar; the default is null.

Methods

The ISmBriefcaseProgress object has the following methods:

Method	Description
SetValue (NewValue)	Called by SmartBriefcase to report on significant progress. NewValue is an integer between 0 and 100, which represents the percentage of progress reported relative to the entire task.

Common Tasks

The following sections describe methods and properties that are used to perform common tasks related to an ISmBriefcase.

ISmBriefcase: Creating and Saving a Briefcase

The following example shows how to create and save a briefcase.

```
Const DATA_FILE = BASE_DIR + "temp.dat"

Dim Briefcase As SmartBriefcase.SmBriefcase
Dim File As ISmBriefcaseFile

' Save mailbox as Ixf data file
SaveAsIxf BomPackage, IXF_FILENAME

' create briefcase
Set Briefcase = CreateObject("SmartBriefcase.smBriefcase")
Briefcase.New

' Add data file
Set File = Briefcase.Files.Add
File.SourceFileName = IXF_FILENAME
' Add application executable file that runs the data file
Set File = Briefcase.Files.Add
File.SourceFileName = BASE_DIR + APPLICATION

' Define configuration
Briefcase.Configuration.PackageId = "My Unique Id"
Briefcase.Configuration.Description = "This briefcase contains mailbox
application and data"
'Command that executes the data file, for example open it and
'display the data
Briefcase.Configuration.ExecuteCommand = APPLICATION + " " + IXF_FILENAME
' Save the briefcase to a file
Briefcase.Save FileName
```

4 SmartBomServices Library

Introduction

The **SmartBomServices Library** provides the following functionality:

- Access the BOM preferences as stored in a SMARTEAM database.
- Access the BOM Configuration settings in a SMARTEAM database.
- Generate a BOM representation from a SMARTEAM assembly.
- Synchronize BOM changes with a SMARTEAM assembly.

Overview of Objects – SmartBomServices

This section presents an overview of the main SmartBomServices objects including a description of the associated objects that are useful for the programmer.

The ISmBomServices is the highest-level object; its main purpose is to contain the other objects. SmartBomServices is a singleton object, executed as a service in a SMARTEAM session.

The major ISmBomServices components are shown in the following object diagram:

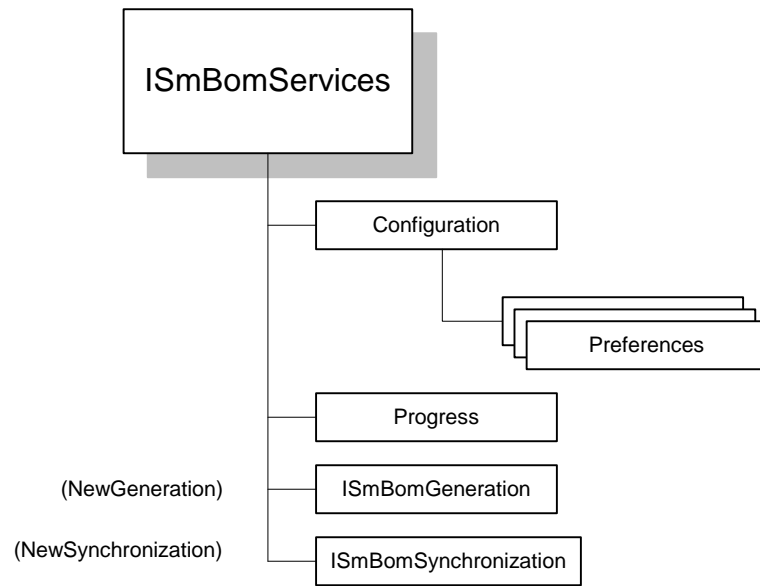


Figure 4-1 ISmBomServices Object Diagram

Properties

The ISmBomServices object has the following properties:

Property	Description
Configuration	Reference to configuration data. Returns ISmBomConfiguration.
Progress	Reference to a ISmProgress interface. To be served as a callback function.

Methods

The ISmBomServices object has the following methods:

Method	Description
NewGeneration	Creates and returns a ISmBomGeneration interface for generating BOM representation based on a SMARTEAM assembly.
NewSynchronization	Creates and returns ISmBomSynchronization interface for synchronizing BOM representation back to SMARTEAM
RetrieveBomPackage	Retrieves a BOM representation from a SmartBOM Package object
SaveBomPackage	Saves a BOM Representation into a SmartBOM Package object
IsCopyTreeEnabled	Returns True if an object of a specific class can take place in the CopyTree process, False otherwise
IsBomEnabled	Returns True if a specific class as a BOM enabled class, False otherwise

Obtaining the ISmBomServices Object

To create an ISmBomServices object from a SMARTEAM SmSession object:

```
Set BomServices = SmSession.GetService("SmartBomServices.SmBomServices")
```

ISmBomConfiguration

The ISmBomConfiguration object is a collection of properties about the current BOM configuration.

Properties

The ISmBomConfiguration object has the following properties:

Property	Description
BomWhiteBoardProcessClassId	Process Class Id to be automatically associated for new BOM WhiteBoards
ConflictResolution	Resolving generation conflicts of more than one Item with the same Part Number
PropagateRevision	Determines in which cases revision should be propagated
PartEffectivityOverlapping	Determines whether part effectivity overlapping is allowed
Preferences	Reference to a collection of BOM preferences. Returns ISmBomPreferences
DefaultPreference	Reference to the default / current BOM preference. Returns ISmBomPreference.

Obtaining the ISmBomConfiguration Object

```
Set BomConfiguration = BomServices.Configuration
```

Example

```
Set BomServices = SmSession.GetService("SmartBomServices.SmBomServices")
Set BomConfiguration = BomServices.Configuration
BomConfiguration.DefaultPreference = BomConfiguration.Preferences.Item(0)
```

ISmBomPreferences

The ISmBomPreferences object provides a reference to BOM preferences.

Methods

The ISmBomPreferences object has the following methods:

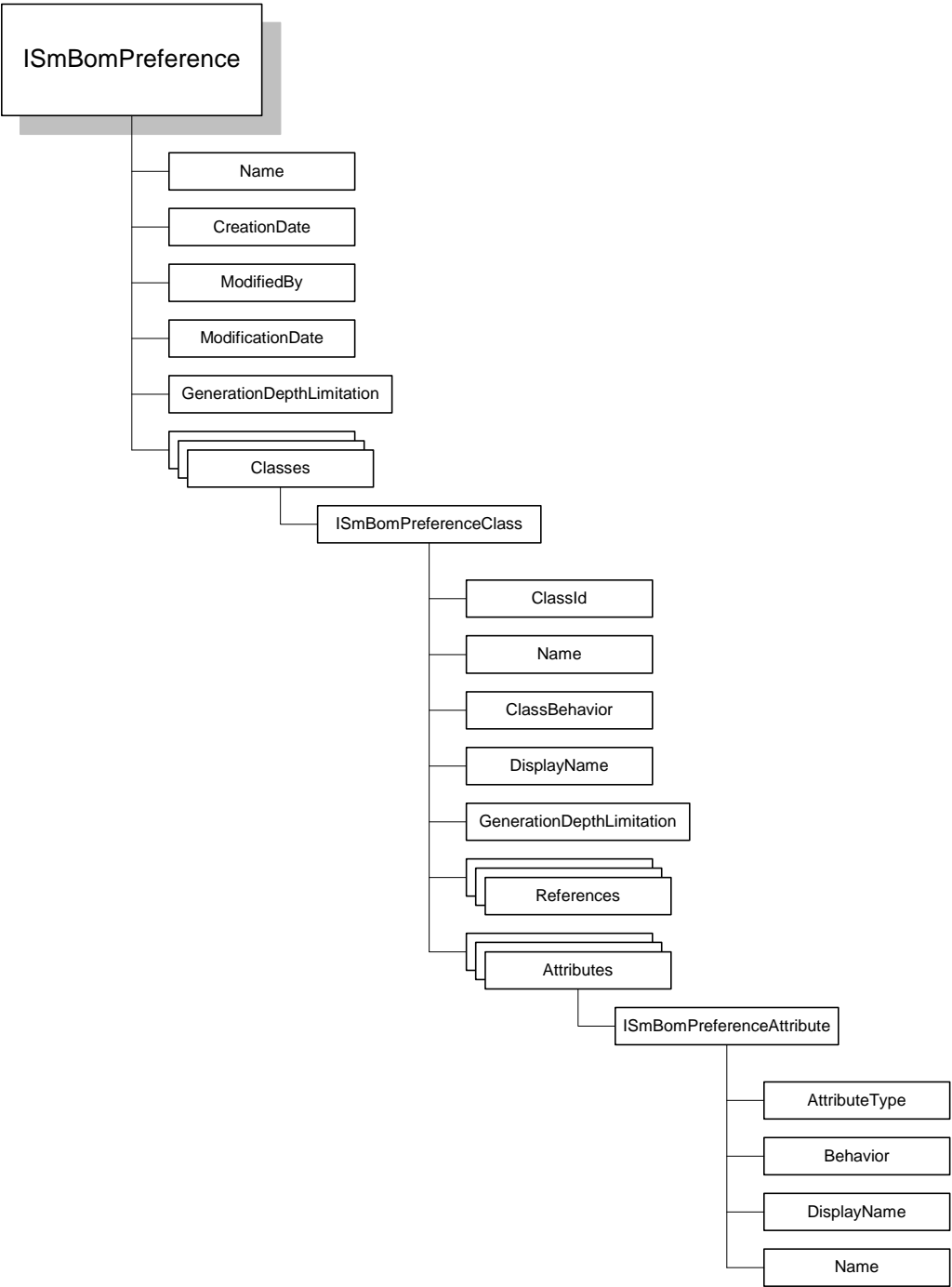
Property	Description
Item	Get ISmPreference by Id or by Index

Obtaining the ISmBomPreferences Object

```
Set BomServices = SmSession.GetService( "SmartBomServices.SmBomServices" )  
Set BomPreferences = BomServices.Configuration.Preferences
```

ISmBomPreference

The ISmBomPreference object represents information about a BOM preference. The object diagram of ISmBomPreference is shown below.



Properties

The ISmBomPreference object has the following properties:

Property	Description
Name	Preference name
CreationDate	Preference creation date
ModifiedBy	User name of last modifier
ModificationDate	Last modification date
GenerationDepthLimitation	Hierarchical depth limitation for generation of BOM representation
Classes	Reference to a collection of BOM-Enabled classes in this preference. Returns ISmBomPreferenceClasses

Methods

The ISmBomPreference object has the following methods:

Property	Description
Clone	Clone the preference

ISmBomPreferenceClasses

The ISmBomPreferenceClasses object represents a set of Preference Classes.

Methods

The ISmBomPreferences object has the following methods:

Property	Description
Item	Get ISmBomPreferenceClass
ItemByClassId	Returns ISmBomPreferenceClass

ISmBomPreferenceClass

The ISmBomPreference object represents information about a BOM preference.

Properties

The ISmBomPreference object has the following properties:

Property	Description
ClassId	Unique Class Id (the BOM-enabled Class ID in the SMARTEAM database)
Name	Unique class name
ClassBehavior	Class behavior for a class in the BOM representation – whether it behaves as a Part class or a Document class.
DisplayName	Class name as displayed to the user
References	Reference to a collection of reference classes for this class. Returns ISmBomPreferenceClasses
Attributes	Reference to a collection of attributes in this class. Returns ISmBomPreferenceAttributes

Methods

The ISmBomPreference object has the following methods:

Property	Description
Clone	Clone the preference class

ISmBomPreferenceAttributes

The ISmBomPreferenceAttributes object represents a set of Preference Attributes.

Methods

The ISmBomPreferences object has the following methods:

Method	Description
Item	Get ISmBomPreferenceAttribute

ISmBomPreferenceAttribute

The ISmBomPreferenceAttribute object represents information about a BOM preference attribute.

Properties

The ISmBomPreference object has the following properties:

Property	Description
AttributeType	Attribute data type
Behavior	Attribute behavior as reflected in BOM representation. For example, as a Part Number or PartDescription, or Revision.
DisplayName	Attribute name as displayed to the user
Name	Attribute unique name

ISmBomGeneration

The ISmBomGeneration object is a collection of properties about the current BOM generation. ISmBomGeneration serves SMARTEAM BOM by generating a BOM representation from SMARTEAM object storage.

Properties

The ISmBomGeneration object has the following properties:

Property	Description
ClassId	Class id of the root assembly to be generated
ObjectId	Object id of the root assembly to be generated
Effectivity	Reference to an effectivity range to filter the BOM representation according to Part and structural effectivity.
Content	BOM representation content. Can be Parts only, Parts and references, or Parts, references and files.
ReplaceRevision	Whether or not to take the latest / last public revision of each Item to the BOM representation
Preference	One of the ISmBOMPreferences to be used when generating the representation (NULL means default preference)

Methods

The ISmBomGeneration object has the following methods:

Method	Description
Execute	Generates the representation based on the relevant ISmBomGeneration properties

Example

```
Set BomServices = SmSession.GetService("SmartBomServices.SmBomServices")
Set BomGeneration = BomServices.NewGeneration
BomGeneration.ClassId = 15
BomGeneration.ObjectId = 40
BomGeneration.Content = gcPartsReferences
Set BomPackage = BomGeneration.Execute
```

ISmBomSynchronization

The ISmBomSynchronization object is a collection of properties about the current BOM synchronization. ISmBomSynchronization serves SMARTTEAM BOM by synchronizing BOM representation changes to SMARTTEAM object storage.

Properties

The ISmBomSynchronization object has the following properties:

Property	Description
Package	The BOM representation to be synchronized
TargetClassId	Class id of root assembly parent / host (Non-zero only for newly created assemblies, otherwise set to zero.)
TargetObjectId	Object id of root assembly parent / host (Non-zero only for newly created assemblies, otherwise set to zero.)
Preference	One of the ISmBOMPreferences to be used when synchronizing the representation (NULL means default preference)

Methods

The ISmBomGeneration object has the following methods:

Method	Description
Execute	Synchronizes the representation based on the relevant ISmBomSynchronization properties

Example

```
Set BomServices = SmSession.GetService("SmartBomServices.SmBomServices")
Set BomSynchronization = BomServices.NewSynchronization
BomSynchronization.Package = BomPackage
BomSynchronization.Execute
```

5 SmartBomGuiServices Library

Introduction

The **SmartBomGuiServices Library** enables you to perform the following functions:

- Supports creation of GUI wizards and dialog boxes relevant to SMARTEAM – BOM

Overview of Objects – SmartBomGuiServices

This section presents an overview of the main SmartBomGuiServices objects including a description of the associated objects that are useful for the programmer. Because of its similarity to SmartBomGuiServices, the ISmBomConsolidationWizard is also included.

The ISmBomGuiServices is the highest-level object; its main purpose is to contain the other objects.

The major ISmBomGuiServices components are shown in the following object diagram:

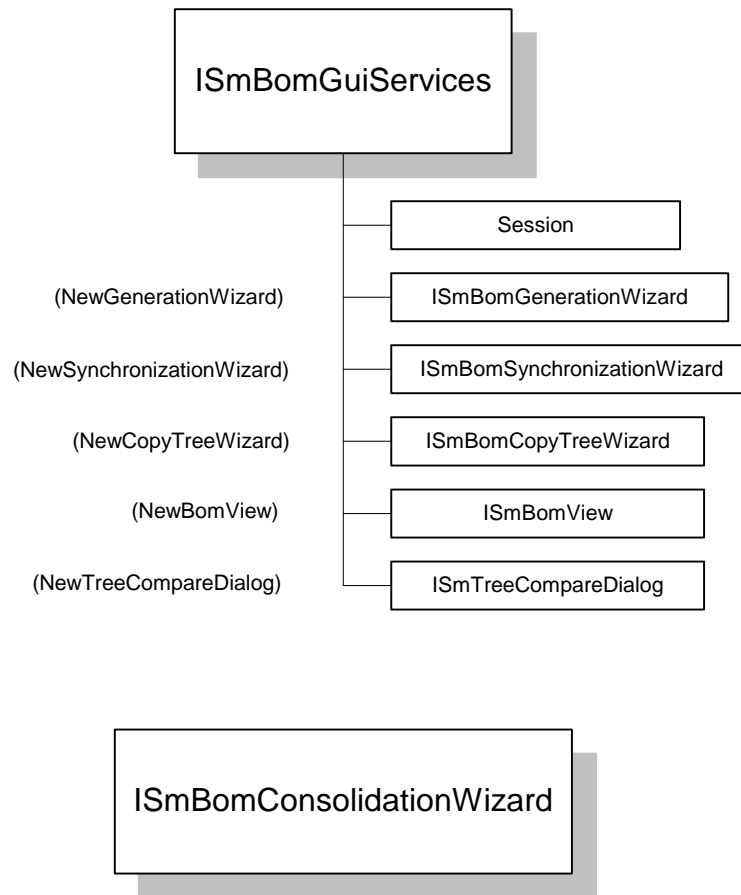


Figure 5-1 *ISmBomGuiServices* Object Diagram (including *ISmBomConsolidationWizard*)

Properties

The ISmBomGUIServices object has the following properties:

Property	Description
Session	Reference to the active SMARTEAM session. This session serves all wizards and dialog boxes in this library.

Methods

The ISmBomGUIServices object has the following methods:

Method	Description
NewGenerationWizard	Creates and returns the ISmBomGenerationWizard interface for opening a Generation Wizard. Returns ISmBomGenerationWizard
NewSynchronizationWizard	Creates and returns the ISmBomSynchronizationWizard interface for opening a Synchronization Wizard. Returns ISmBomSynchronizationWizard
NewCopyTreeWizard	Creates and returns the ISmBomCopyTreeWizard interface for opening a CopyTree Wizard. Returns ISmBomCopyTreeWizard
NewBomView	Creates and returns the ISmBomView interface for opening a BOM View. Returns ISmBomView
NewTreeCompareDialog	Creates and returns the ISmTreeCompareDialog interface for opening a Tree Compare dialog. Returns ISmTreeCompareDialog

Obtaining the SmartBomGuiServices Object

To create an SmartBomGuiServices object:

```
Set BomGuiServices = CreateObject("SmartBomGuiServices.SmBomGuiServices")
```

Example

```
Set BomGuiServices = CreateObject("SmartBomGuiServices.SmBomGuiServices")  
Set BomGuiServices.Session = SmSession
```

ISmBomGenerationWizard

The ISmBomGenerationWizard object is an interface through which the Generation Wizard is displayed. The Generation Wizard generates the BOM as a file (BOM Active Briefcase or BOM Briefcase) from the root part of the assembly.

Properties

The ISmBomGenerationWizard object has the following properties:

Property	Description
ClassId	Class id of the root part of the assembly
ObjectId	Object id of the root part of the assembly
FileName	File name with full path to the generated BOM Briefcase file

Methods

The ISmBomGenerationWizard object has the following methods:

Method	Description
ShowModal	Opens the Generation Wizard

ISmBomSynchronizationWizard

The ISmBomSynchronizationWizard object is an interface through which the Synchronization Wizard is displayed. The Synchronization Wizard implements a BOM package into a SMARTEAM database. The source can be a BOM Briefcase file or a reference to a BOM package.

Properties

The ISmBomSynchronizationWizard object has the following properties:

Note: You can use only one of the following properties as the source for the Synchronization Wizard.

Property	Description
BomPackage	Reference to the BOM package
FileName	File name with full path to a BOM Briefcase

Methods

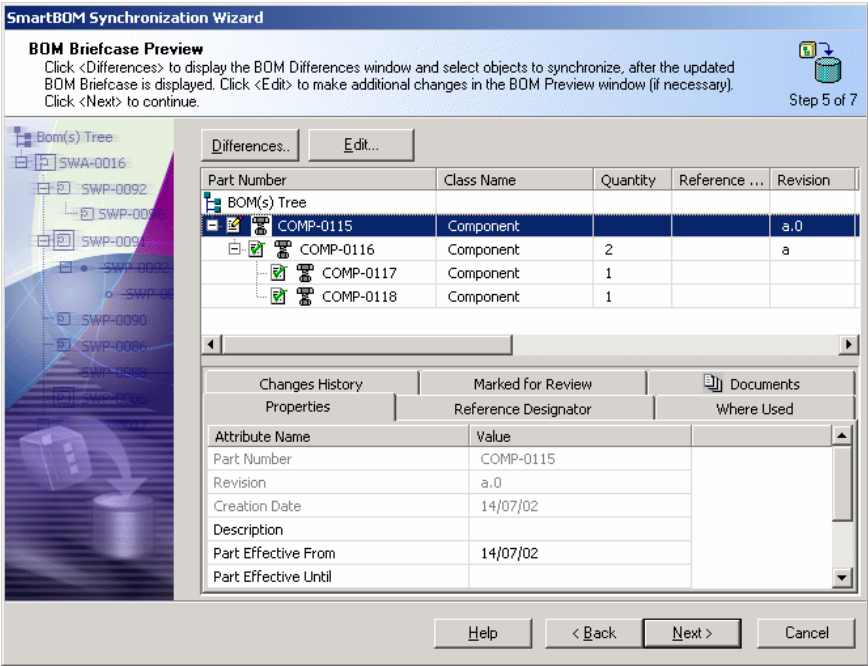
The ISmBomSynchronizationWizard object has the following methods:

Method	Description
ShowModal	Opens the Synchronization Wizard

Example

```
Set BomGuiServices = CreateObject("SmartBomGuiServices.SmBomGuiServices")
Set BomSynchronizationWizard = BomGuiServices.NewSynchronizationWizard
Set BomSynchronizationWizard.BomPackage = Package
BomSynchronizationWizard.ShowModal
```

The resulting window appears as follows:



The ISmTreeCompareDialog object is an interface through which the TreeCompare dialog is displayed. The TreeCompare dialog lets you compare two trees in a BOM package and displays the results in a GUI window. The ISmTreeCompareDialog object is generic and can compare two trees of any type in any SMARTEAM-based product.. You specify the roots of both trees to be compared.

The ISmTreeCompareDialog object has the following properties:

Property	Description
ClassId1	Class id of first object to be compared
ClassId2	Class id of second object to be compared
ObjectId1	Object id of first object to be compared
ObjectId2	Object id of second object to be compared

Methods

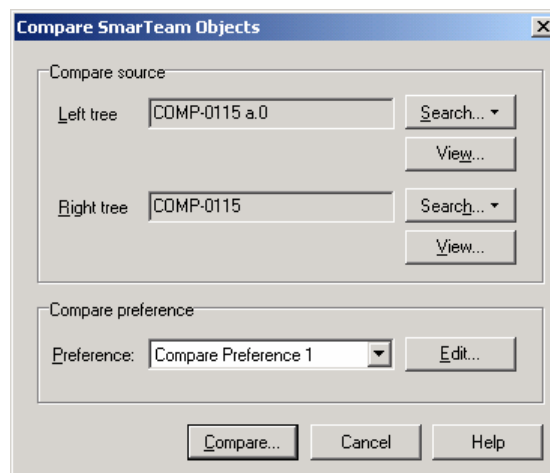
The ISmTreeCompareDialog object has the following methods:

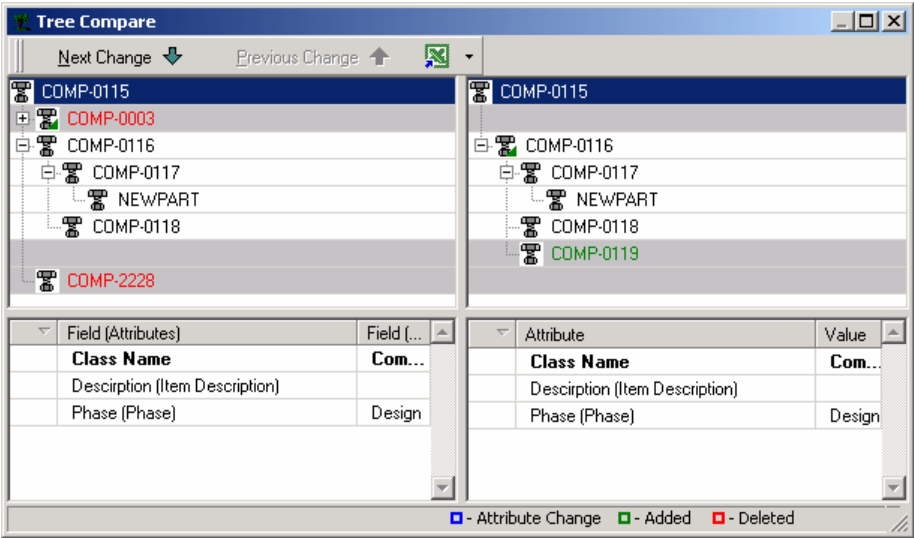
Method	Description
ShowModal	Opens the TreeCompare dialog

Example

```
Set BomGuiServices = CreateObject("SmartBomGuiServices.SmBomGuiServices")
Set TreeCompareDialog = BomGuiServices.NewTreeCompareDialog
TreeCompareDialog.ClassId1 = 15
TreeCompareDialog.ObjectId1 = 40
TreeCompareDialog.ClassId1 = 15
TreeCompareDialog.ObjectId1 = 47
TreeCompareDialog.ShowModal
```

The resulting windows appear as follows:





ISmBomConsolidationWizard

The ISmBomConsolidationWizard object is an interface through which the BomConsolidation dialog is displayed. The BomConsolidation dialog lets you consolidate Boms from a primary and secondary BOM package into a single BOM package and displays the results in a GUI window. You specify the primary and secondary BOM packages by package reference or BOM Briefcase file.

Properties

The ISmBomConsolidationWizard object has the following properties:

Note: You can use either the package reference or the BOM Briefcase file and not both.

Property	Description
UserName	User name
UserEmail	User email
PrimaryPackage	Reference to the primary BOM Package
PrimaryFileName	File name with full path to the primary BOM Briefcase file
SecondaryPackage	Reference to the secondary BOM Package

SecondaryFileName	File name with full path to the secondary BOM Briefcase file
Destination	Full path to the consolidated BOM Briefcase file
ResultPackage	Reference to the consolidated BOM Package
ParentWindow	Handle of the parent window

Methods

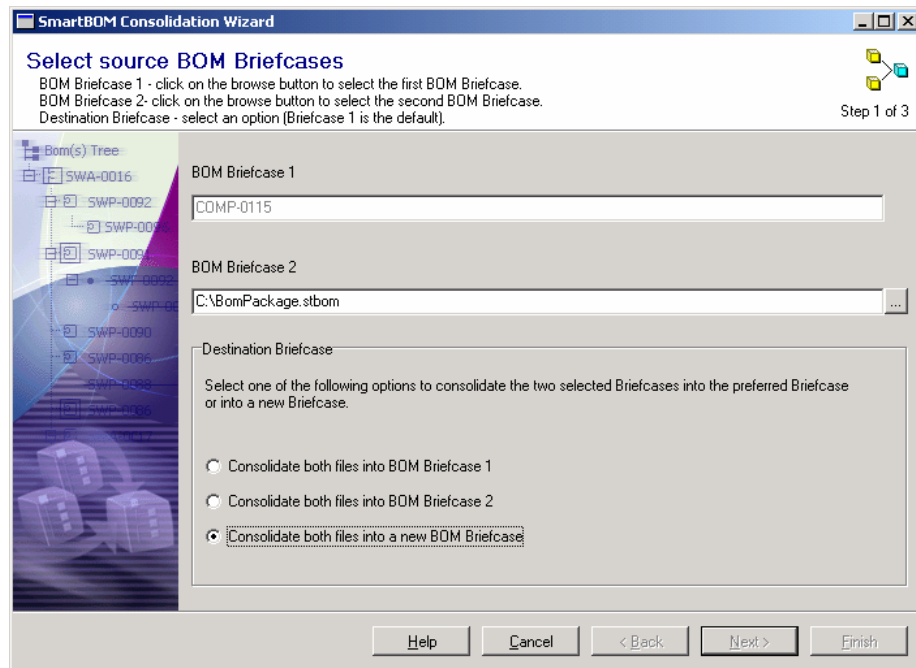
The ISmBomConsolidationWizard object has the following methods:

Method	Description
ShowModal	Opens the Consolidation Wizard

Example

```
Set BomGuiServices = CreateObject("SmartBomGuiServices.SmBomGuiServices")
Set ConsolidationWizard = BomGuiServices.New ConsolidationWizard
ConsolidationWizard.UserName = 'Myself'
ConsolidationWizard.UserEmail = 'Myself@me.com'
Set ConsolidationWizard.PrimaryPackage = BomPackage
ConsolidationWizard.SecondaryFileName = 'c:\BomPackage.stbom'
ConsolidationWizard.Destination = ConsolidationDestinationEnum
.cdNewPackage
ConsolidationWizard.ShowModal
```

The resulting window appears as follows:



6 An Automotive Application

The following application illustrates the use of the libraries:

- ISmSmartBOM
- ISmBOMIxf

The application creates a structure of automotive parts, each of class Part and a specification document of class Document where the document is linked to the root assembly part. The resulting BOM Package is saved in Ixf format and packed into an active BOM Briefcase.

The constants in the example are directed to a local folder. If you wish to execute the example, you should make sure the folder exists. In addition, the file Document.doc must exist. Alternatively, you can set a different value for BASE_DIR.

```
' Constants
Const BASE_DIR = "C:\Tmp\Demo\"
Const USER_NAME = "User"
Const USER_EMAIL = "user@smarteam.com"
Const APPLICATION = "SmartBomEditor.exe"
Const IXF_FILENAME = BASE_DIR + "Package.stbom"
Const BRIEFCASE_FILE = BASE_DIR + "Package.exe"
Const DOCUMENT_FILENAME = "Document.doc"
Const DOCUMENT_FULL_FILENAME = BASE_DIR + DOCUMENT_FILENAME

' declare global variable
Dim BomPackage As SmartBom2.SmBomPackage

Private Sub CreateBomBut_Click()

    Dim aPart As SmartBom2.ISmPart

    ' create and init package
    Set BomPackage = CreatePackage(USER_NAME, USER_EMAIL)

    ' Define schema for BOM
    SetSchema BomPackage.Schemas.Item(0)
```

```
' Add Items to BOM
Set aPart = AddItem(BomPackage, Nothing, 1, "VOLVO 930", "Family car",
"Volvo", 13, 4)
Call AddItem(BomPackage, aPart, 4, "AX-300", "Wheel", "Goodyear", 1, 1)
Call AddItem(BomPackage, aPart, 1, "EG-2001", "Engine", "Volvo", 3, 2)

' Add technical specification document to root assembly
AddDocument BomPackage, aPart, "DOC-0001", "Technical Specifications",
DOCUMENT_FILENAME, DOCUMENT_FULL_FILENAME

' display changes in BOM
DisplayChanges BomPackage

' Save the BOM package as an Active BOM Briefcase
SaveAsBriefcase BomPackage

' Display Message to user
MsgBox "BOM Package created successfully"
End Sub

Function CreatePackage(UserName As String, UserEmail As String) As
ISmBomPackage
    Dim BomPackage As Object

    ' Create the package and set its initial attributes
    Set BomPackage = CreateObject("SmartBOM2.SmBomPackage")
    BomPackage.PackageInfo.UserDetails.Name = UserName
    BomPackage.PackageInfo.UserDetails.Email = UserEmail

    'return BomPackage
    Set CreatePackage = BomPackage
End Function

Sub SetSchema(Schema As ISmSchema)
    Dim AClass As ISmBomClass
    Dim AAttribute As ISmBomAttribute

    ' add the "Item" class to the Schema
    Set AClass = Schema.Classes.Add("Item")
    'Set its attributes
```



```
AClass.ClassType = ctPart
AClass.IsFileManaged = False
'Add composition rule that allows general links between objects of Item
class and objects of Document class
AClass.CompositionRules.GeneralLinks.Add ("Document")
'Add composition rule that allows hierarchical links between objects in
Item class
AClass.CompositionRules.AssemblyLinks.Add ("Item")

' Add custom attributes to the "Item" class
Set AAttribute = AClass.Attributes.Add("Manufacturer")
AAttribute.AttributeType = atString
Set AAttribute = AClass.Attributes.Add("Height")
AAttribute.AttributeType = atInt
Set AAttribute = AClass.Attributes.Add("Weight")
AAttribute.AttributeType = atInt

' Add the "Document" class to the Schema
Set AClass = Schema.Classes.Add("Document")
AClass.ClassType = ctDocument
AClass.IsFileManaged = True
'Add composition rule that allows general links between objects of Item
class and objects of Document class
AClass.CompositionRules.GeneralLinks.Add ("Item")
End Sub

Function AddItem(BomPackage As ISmBomPackage, ParentItem As ISmPart,
Quantity As Integer, PartNumber As String, Description As String,
Manufacturer As String, Height As Integer, Weight As Integer) As ISmPart

Dim aParentLink As ISmParentLink
Dim aPartMaster As ISmPartMaster

'Create new PartMaster
Set aPartMaster = BomPackage.Boms.Item(0).NewPartMaster(PartNumber,
"Item")
aPartMaster.Description = Description

'Define item as root (ptAssembly) if the item has no parent (if the
parameter ParentItem is not assigned)
If (ParentItem Is Nothing) Then
```

```
aPartMaster.PartType = ptAssembly
' Assign the item to the BOM assembly as a root item
BomPackage.Boms.Item(0).Assembly = aPartMaster.Part
Else
' Start tracking changes
BomPackage.ChangesManager.StartTrackingChanges

' Add this part as a child of the ParentItem (passed as a parameter)
aPartMaster.PartType = ptDetail
Set aParentLink = ParentItem.AddChild(aPartMaster.Part)
aParentLink.Quantity.Value = Quantity
End If

' Set properties of Part
aPartMaster.Part.Description = Description
aPartMaster.Part.Properties.Item(aPartMaster.Part.Properties.
    IndexOf("Manufacturer")).Value = Manufacturer
aPartMaster.Part.Properties.Item(aPartMaster.Part.Properties.
    IndexOf("Height")).Value = Height
aPartMaster.Part.Properties.Item(aPartMaster.Part.Properties.
    IndexOf("Weight")).Value = Weight

' Stop tracking changes
BomPackage.ChangesManager.StopTrackingChanges

'Return the added item
Set AddItem = aPartMaster.Part

End Function

Sub AddDocument(Package As ISmBomPackage, Part As ISmPart, DocumentId As
String, Description As String, AttachedFile As String,
AttachedFileFullPath As String)

Dim aDocumentMaster As ISmDocumentMaster
Dim File As ISmFile

' Define a new DocumentMaster of class Document with unique name
'(DocumentId)
```

```
Set aDocumentMaster = Package.Boms.Item(0).NewDocumentMaster(DocumentId,
"Document")

' Define an ISmFile object corresponding to the attached file,
' if present
If AttachedFile <> "" Then
    Set File = Package.Boms.Item(0).NewFile("", AttachedFile,
AttachedFileFullPath)
    'Define File as embedded in BOM package
    File.Embedded = True
    'Associate the file with the document object
    aDocumentMaster.Document.FileObject = File
End If

' Start tracking changes
Package.ChangesManager.StartTrackingChanges

' Attach document to the part by general link
Call Part.LinkDocument(aDocumentMaster.Document)

' Stop tracking changes
Package.ChangesManager.StopTrackingChanges
End Sub

Sub DisplayChanges(Package As ISmBomPackage)
    Dim n As Integer
    Dim MessageText As String
    Dim aChange As ISmChange

    ' Run over all changes in BOM package
    For n = 0 To Package.Changes.Count - 1
        Set aChange = Package.Changes.Item(n)

        ' Set message text according the Operation
        Select Case aChange.Operation
            Case coAddPart
                MessageText =
aChange.OperationInfo.AddPartOperation.Part.PartMaster.PartNumber + " was
added."
            Case coUpdatePartAttribute
```

```
        MessageText = "Attribute " +  
aChange.OperationInfo.UpdatePartAttributeOperation.AttributeName + " has  
been updated in part " +  
aChange.OperationInfo.UpdatePartAttributeOperation.Part.PartMaster.  
PartNumber  
    Case coUpdateLinkAttribute  
        MessageText = "Link attribute " +  
aChange.OperationInfo.UpdateLinkAttributeOperation.AttributeName + " has  
been updated in part " +  
aChange.OperationInfo.UpdateLinkAttributeOperation.Part.PartMaster.  
PartNumber  
    Case coAddDocumentToPart  
        MessageText = "Document " +  
aChange.OperationInfo.AddDocumentToPartOperation.Document.DocumentMaster.  
Id + " has been added to part " +  
aChange.OperationInfo.AddDocumentToPartOperation.Part.PartMaster.  
PartNumber  
    ' ...  
    ' ...  
    Case Else  
        MessageText = "Unknown change"  
    End Select  
  
    ` Display changes to user  
    MsgBox MessageText  
Next n  
End Sub  
  
Sub SaveAsIxf(Package As ISmBomPackage, FileName As String)  
    Dim BomIxf As ISmBomIxfHelper  
    'Create SmartBomIxfHelper for converting the BOM package to an Ixf file  
    Set BomIxf = CreateObject("SmartBomIxf.SmartBomIxfHelper")  
  
    ' save it as an IXF file  
    BomIxf.IxfFromBomPackage Package, FileName  
  
End Sub  
  
Function LoadFromIxf(FileName As String) As ISmBomPackage  
    Dim BomIxf As SmartBomIxf.SmartBomIxfHelper
```

```
'Create SmartBomIxfHelper for converting an Ixf file to a BOM package
Set BomIxf = CreateObject("SmartBomIxf.SmartBomIxfHelper")
' Load a BOM package from an IXF file. BASE_DIR serves as the temporary
' folder to which the files in the Package are extracted
Set Package = BomIxf.BomPackageFromIxf(FileName, BASE_DIR)

' return the package
Set LoadFromIxf = Package

' Display Message
MsgBox "BOM Package loaded successfully"
End Function

Private Sub SaveAsBriefcase(Package as ISmBomPackage)
    Dim BomIxf As SmartBomIxf.SmartBomIxfHelper

    ' Create SmartBomIxfHelper for saving the BOM package as a active BOM
    ' Briefcase
    Set BomIxf = CreateObject("SmartBomIxf.SmartBomIxfHelper")
    BomIxf.ActiveBriefcaseFromBomPackage Package, BRIEFCASE_FILE, ""

    ' Display Message
    MsgBox "Active BOM Briefcase has been saved successfully"
End Sub
```