

BHoM Adapter

Workflow diagrams for the *Push* Adapter Action

Notes and Conventions

Each page presents one method, explaining its workflow.

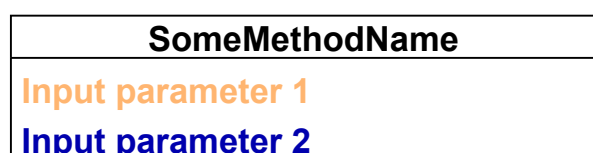
The method name and required parameter are included in a *colored box* (see Legend below) in the top left corner.

Page number is indicated in the top-right corner.

Legend of symbols

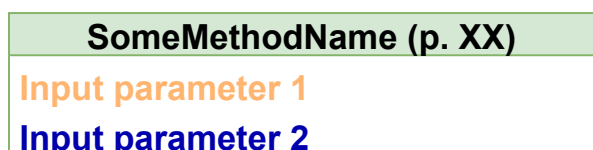


Start/end of a workflow within the currently presented method.

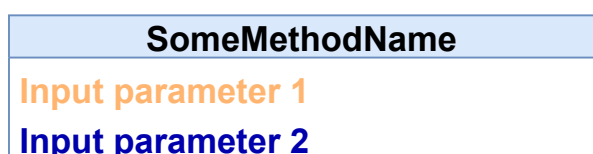


← Indicates *call to a Method* or *the title of the method being explained*.

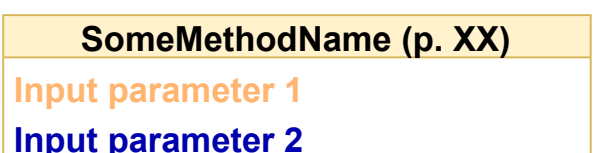
} The parameters are color coded to be identified in the workflow.



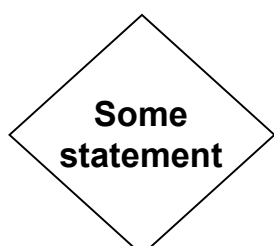
Indicates a Method defined in the Base Adapter.
XX is the document page where the called method is explained.



Indicates a Method *defined/to be defined outside* the Base Adapter
(most commonly, in the specific Adapter implementation in a Toolkit).
As such, their explanation is not included in this document.



Indicates a Method defined in the Base BHoM_Engine.
These are complex method whose understanding is not necessary for the
general learning of the workflow; however, they are included in this document.



Depending on the text, this can be:
- Branching (if-else; switch case; etc.)
- Iteration over a set of items



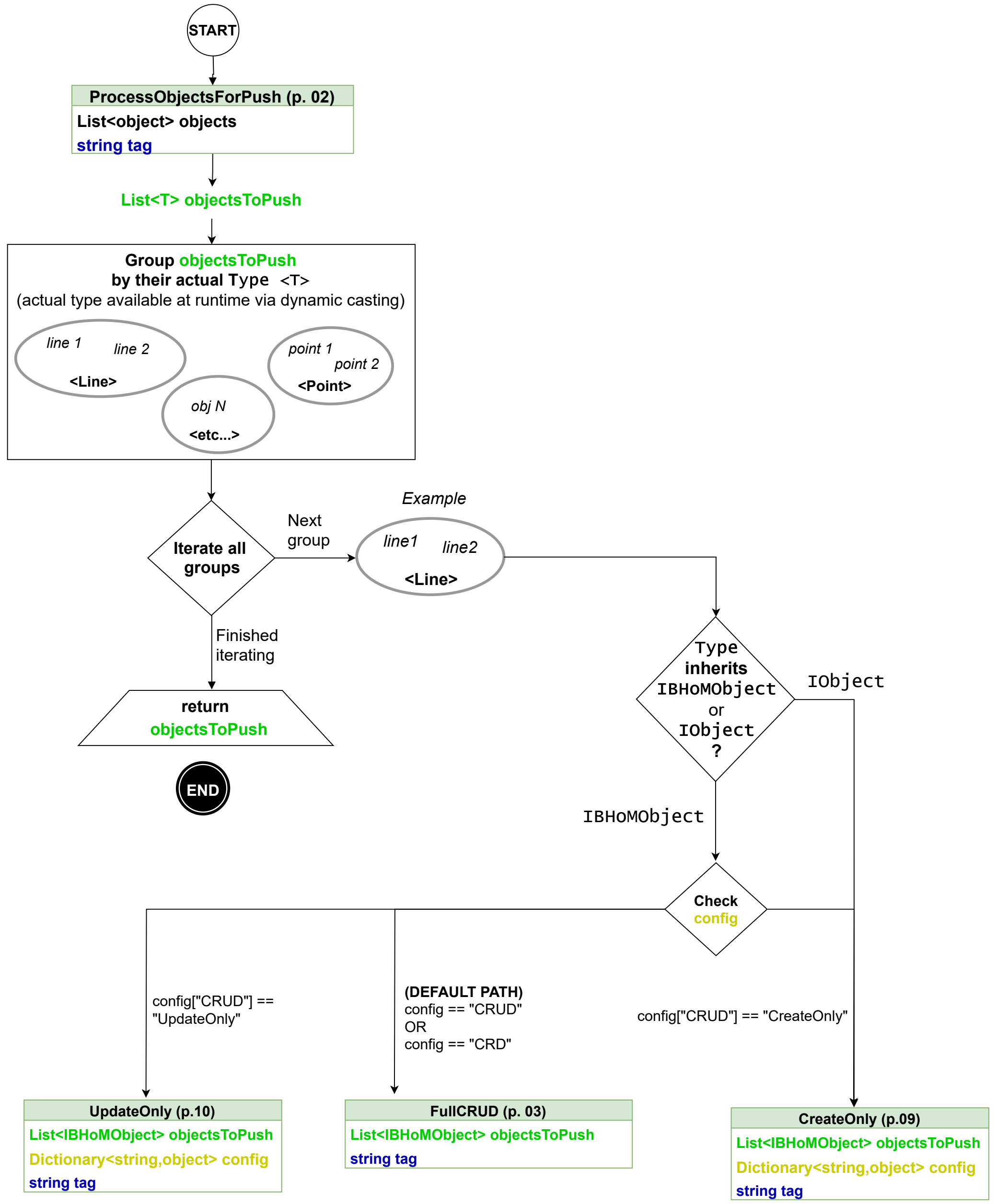
Indicates Property or Method to be defined (implemented) in specific Adapter in a Toolkit.

PUSH

The base Push method, defined in the Base BHoM_Adapter.
 This gets invoked by any UI, when a Push component is activated.
 You can override this method completely if this doesn't suit your Toolkit needs.

```

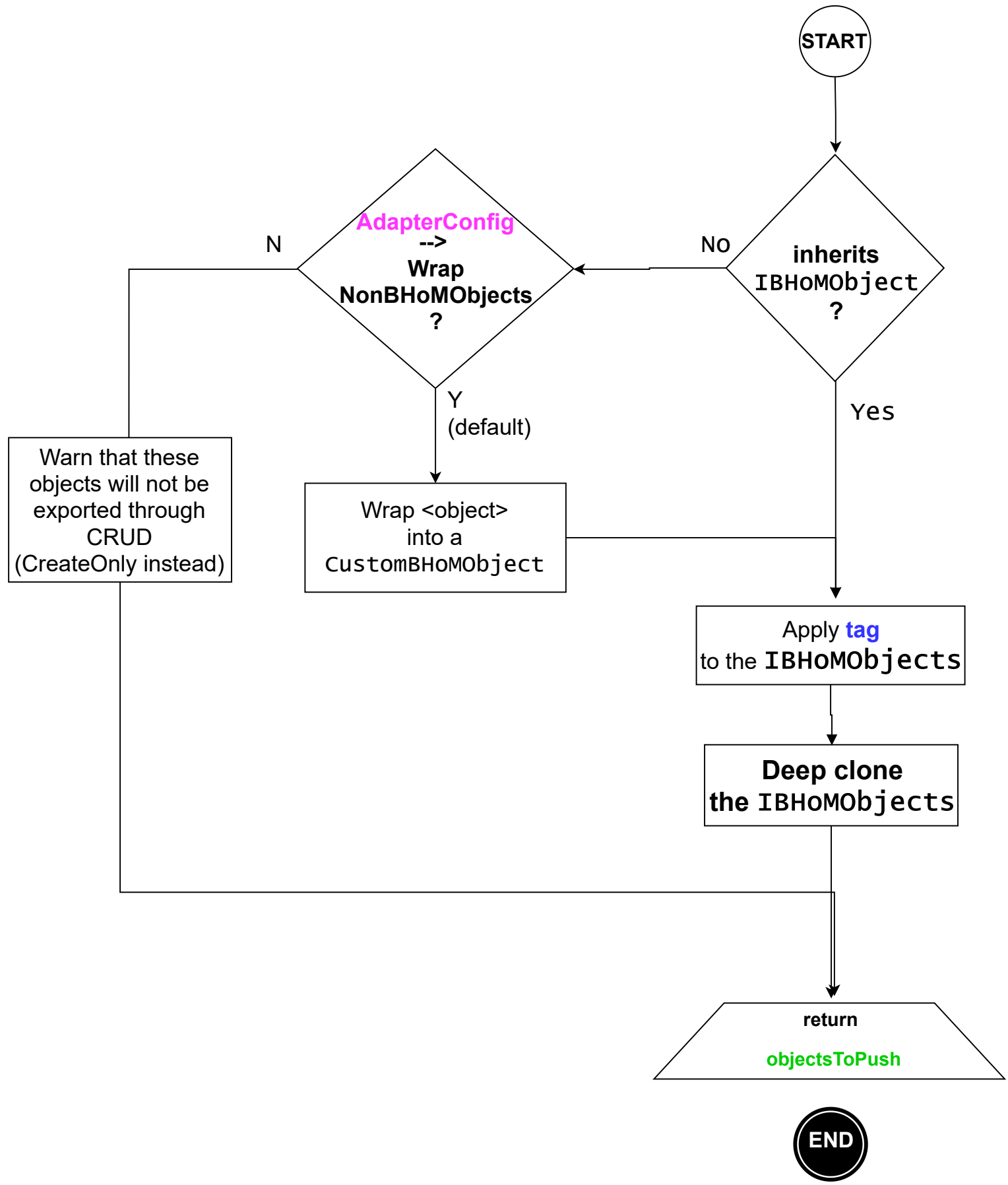
Push
List<object> objects
string tag
Dictionary<string,object> config
  
```



ProcessObjectsForPush

```
ProcessObjectsForPush  
List<object> objects  
string tag
```

Some set up needed on the objects before the Push is performed.



FullCRUD

NOTE 1

NOTE 2

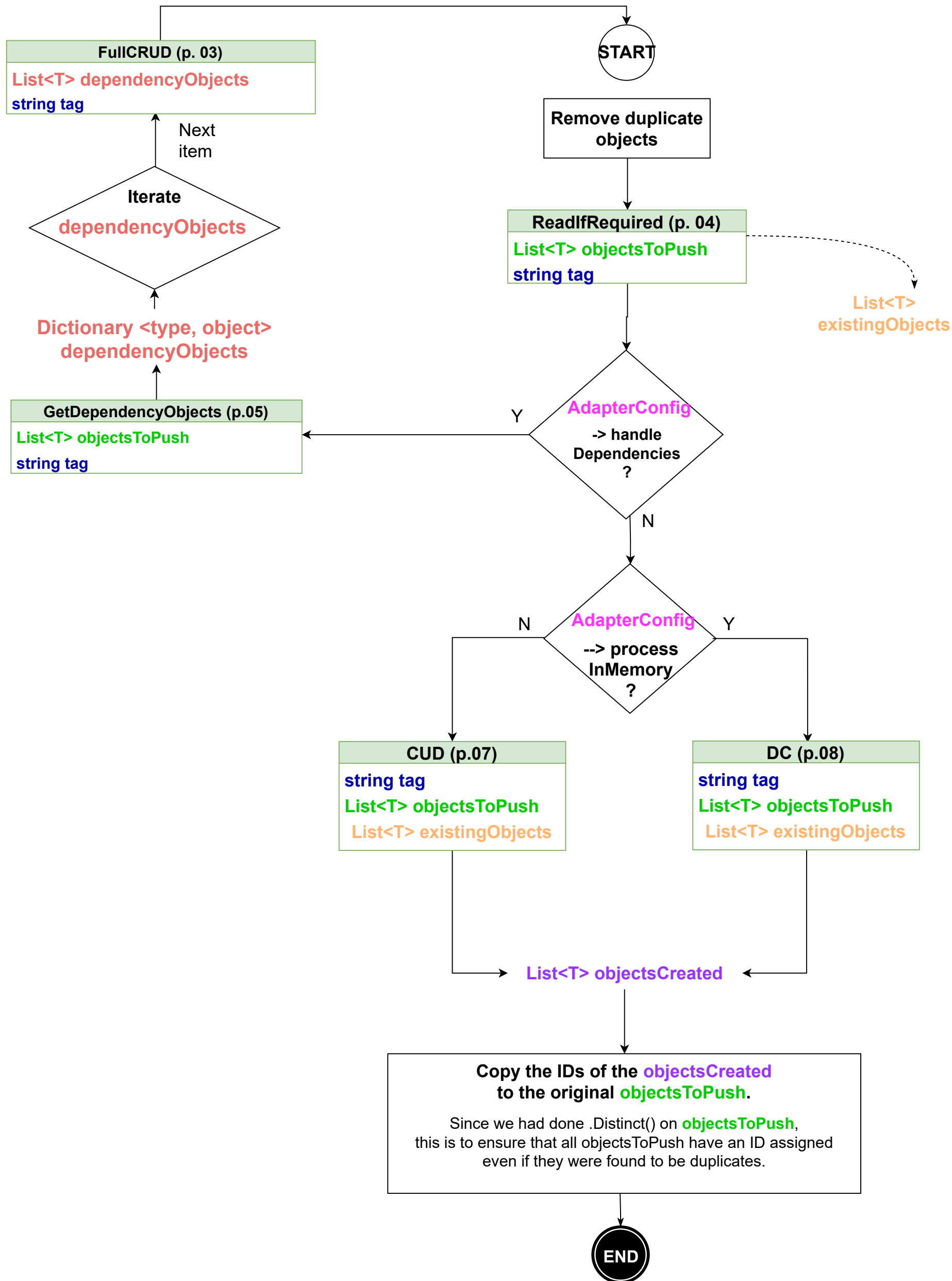
```
FullCRUD
List<T> objectsToPush
string tag
```

Calls all CRUD method as appropriate:
 Reads the external model to understand if there are objects that already exists there that overlap with the ones currently being pushed;
 then it either Creates, Updates or Deletes the objects, depending on what's been implemented in the specific Toolkit.



=
 to be defined
 in specific Toolkit
 Adapter

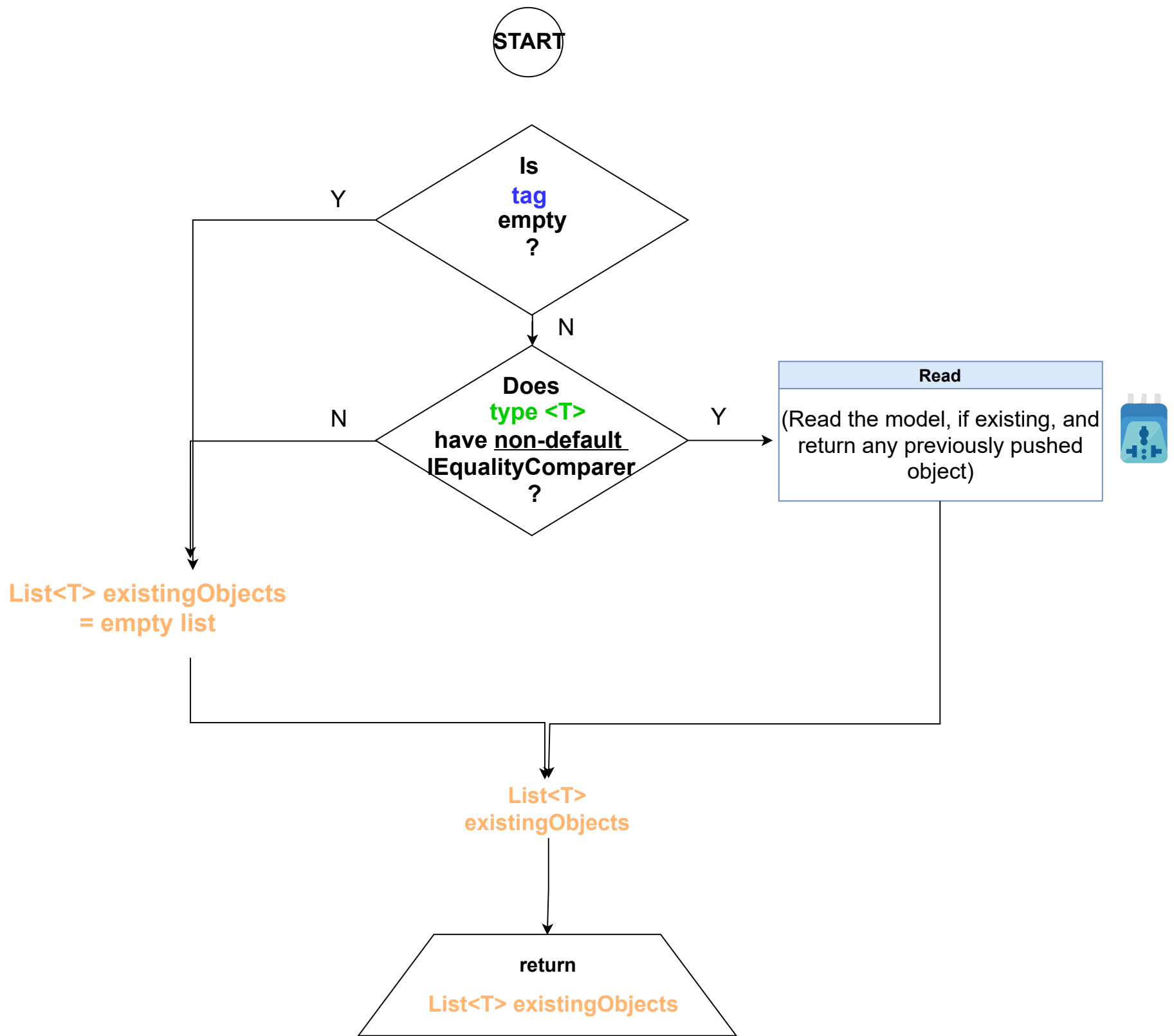
```
Adapter Fields
AdapterConfig
ErrorLog
Adapter GUID
AdapterId
DependencyTypes()
```



ReadIfRequired

```
ReadIfRequired  
List<T> objectsToPush  
string tag
```

Calls Read if tags and comparers are present and meaningful to the rest of the CRUD. In order for the Read to be meaningful, you need to be able to compare the objects currently being Pushed with the ones that will be Read from the External model.



GetDependencyObjects

GetDependencyObjects

```

List<T> objectsToPush
string tag
Dictionary<string,object> config

```

BHoM does not define inheritance chain between objects. The dependency between two types (Bar -> Nodes) has to be defined in the DependencyTypes field in the specific Toolkit; using that, the dependencies are here collected.

NOTE

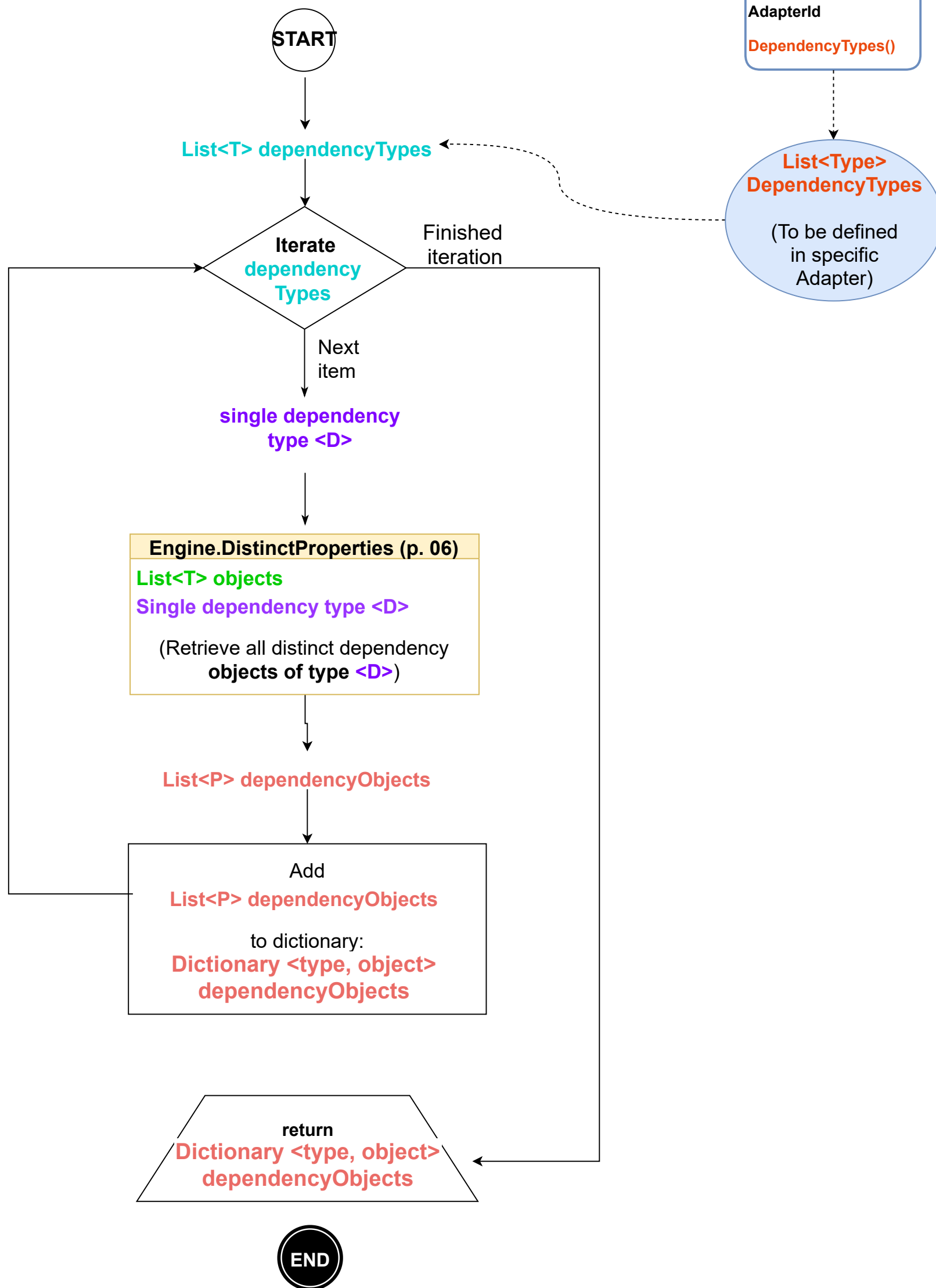
```

Adapter Fields
AdapterConfig
ErrorLog
Adapter GUID
AdapterId
DependencyTypes()

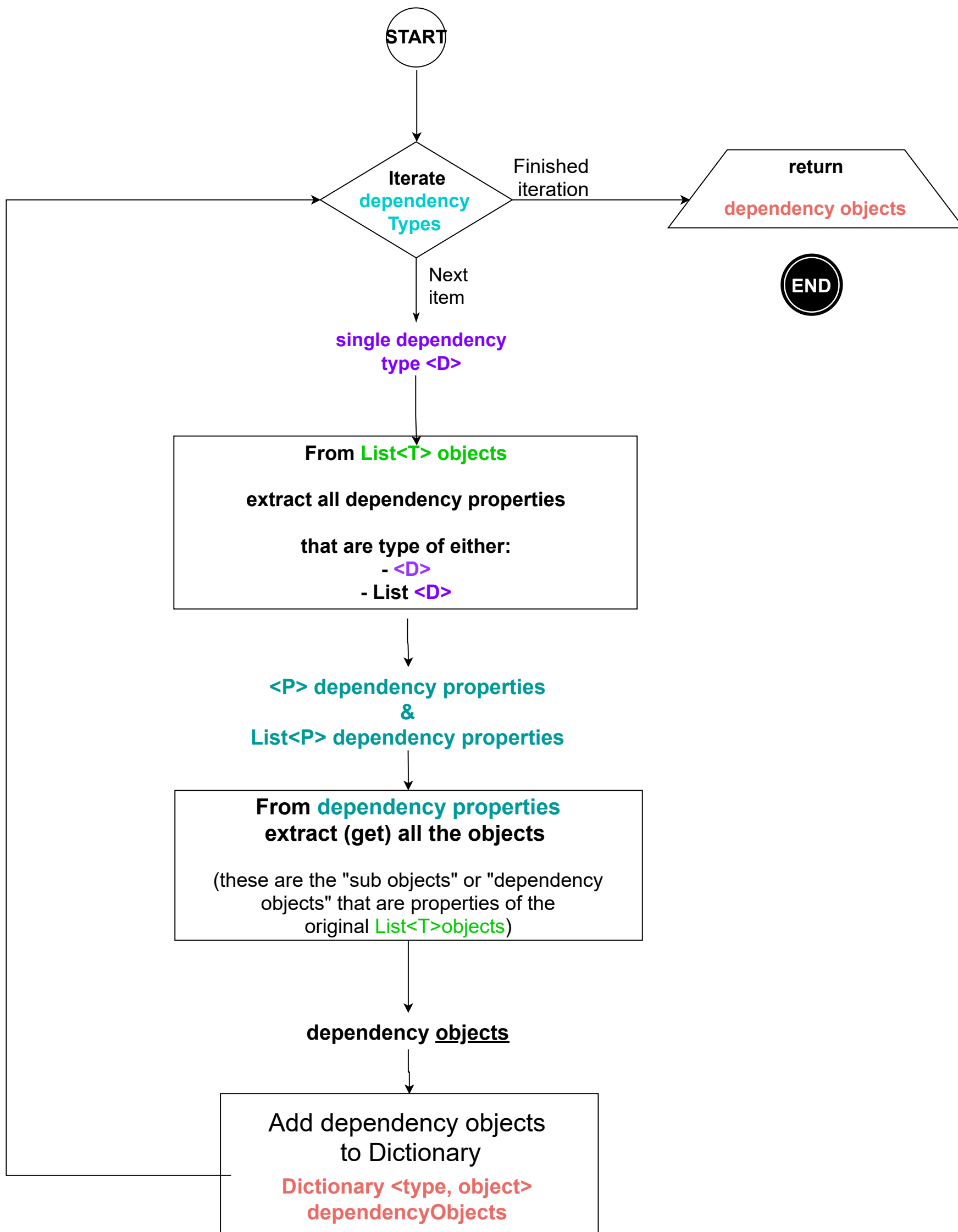
```



to be defined in specific Toolkit Adapter



Engine.DistinctProperties
List<T> objects
List<type> dependencyTypes

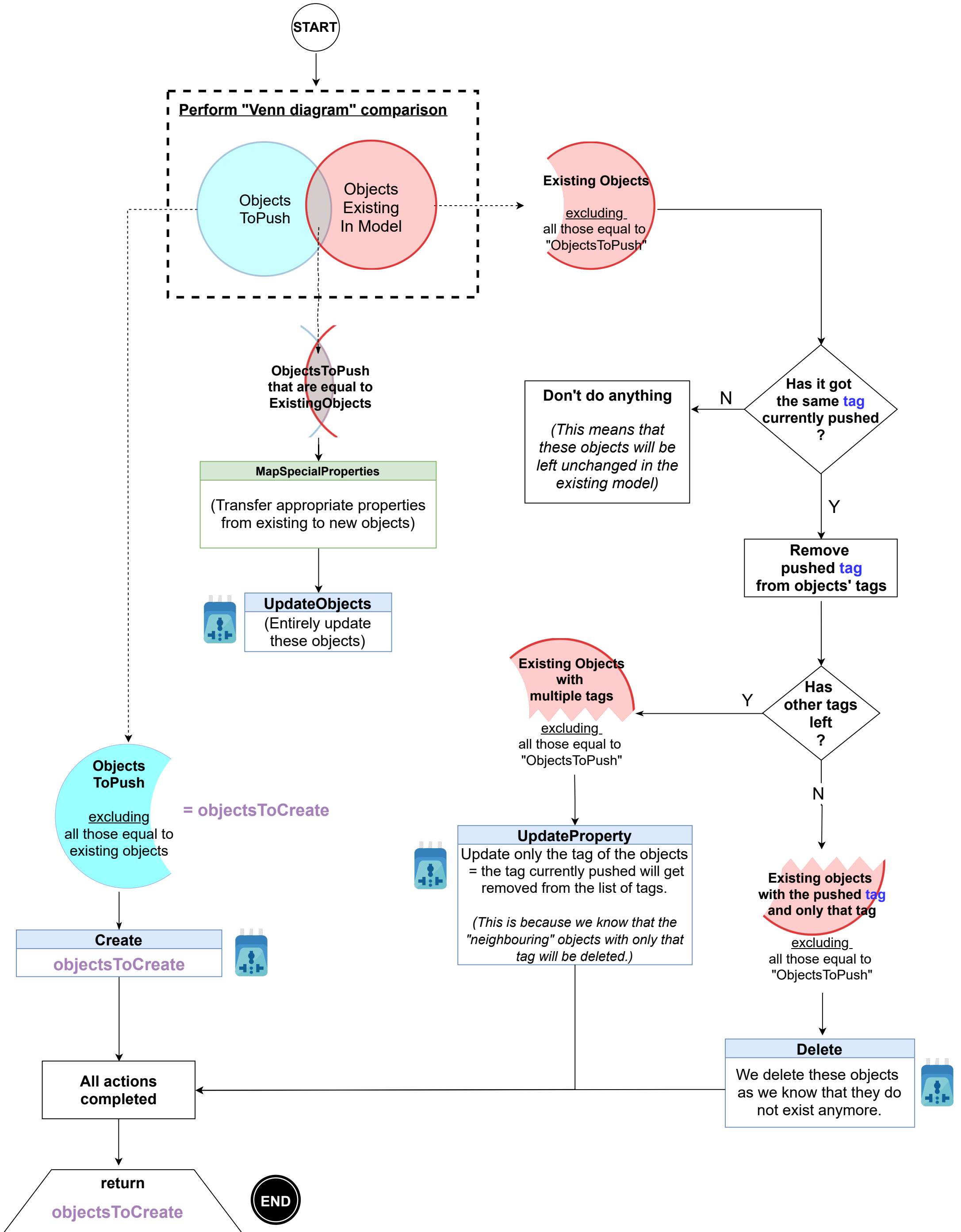


CUD (ReplaceThroughAPI)

```

ReplaceThroughAPI
List<T> objectsToBePushed
string tag

```

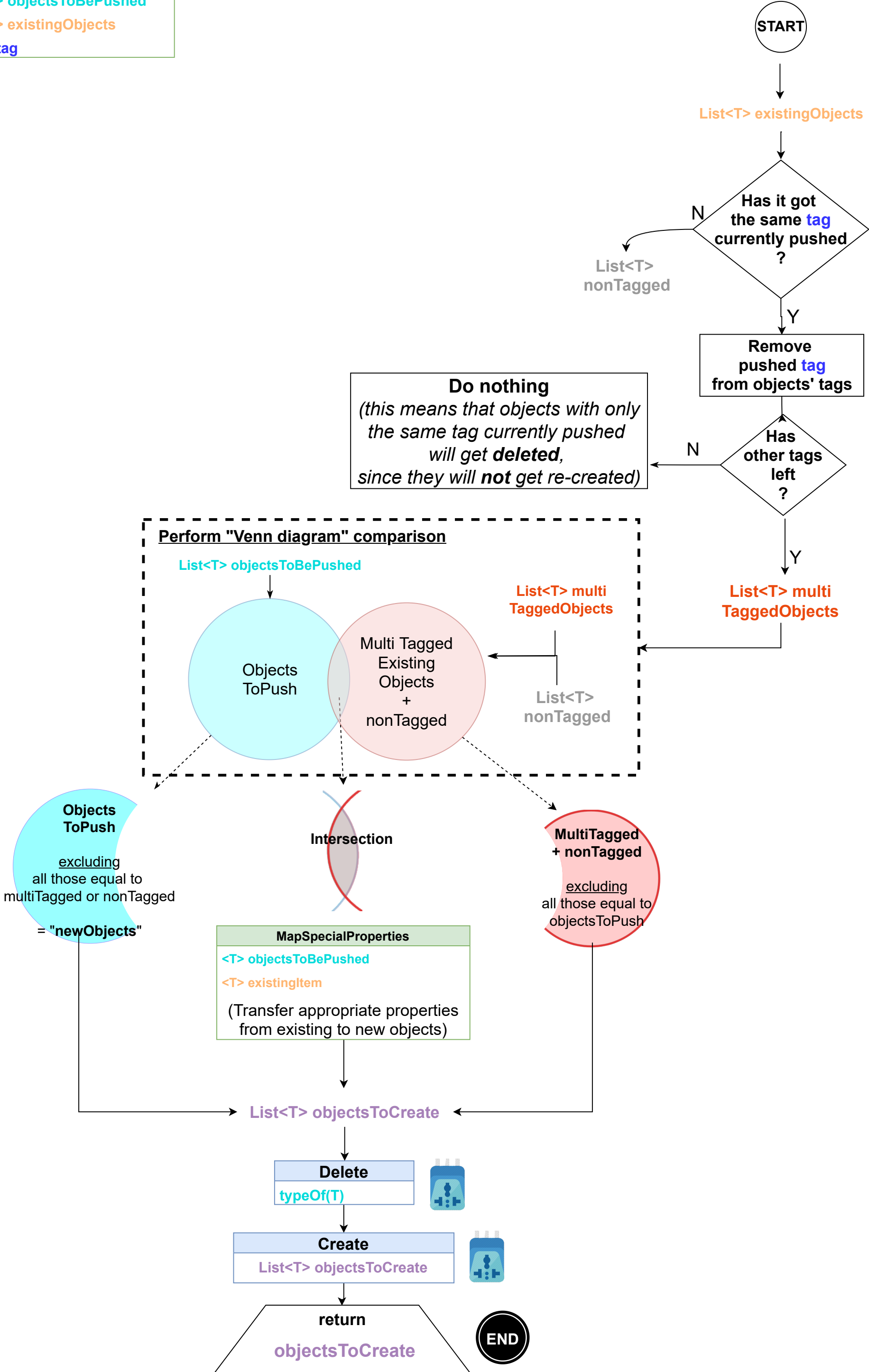


DC (ReplaceInMemory)

```

CUDInMemory
List<T> objectsToBePushed
List<T> existingObjects
string tag

```



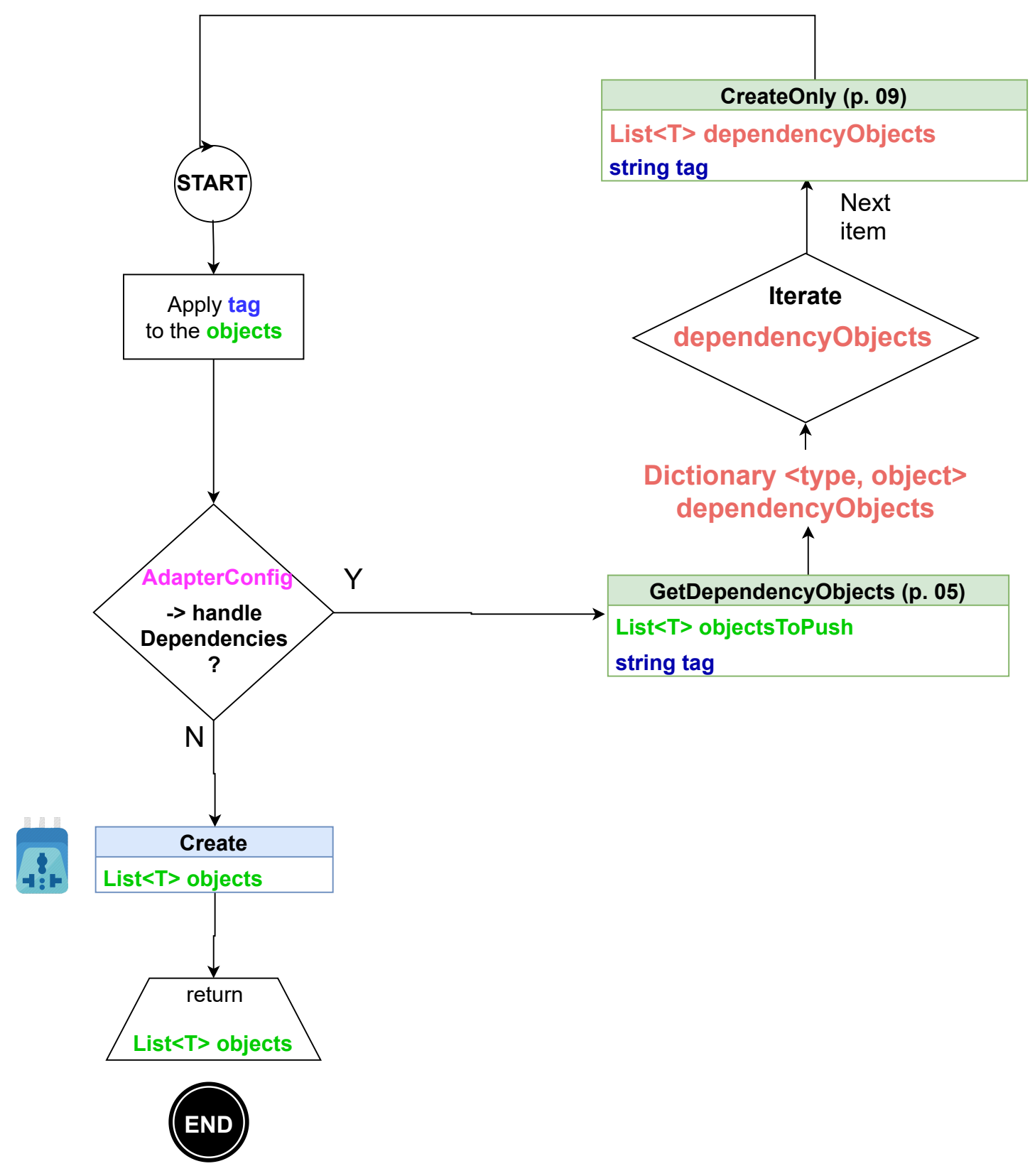
CreateOnly

```

CreateOnly
List<T> objects
string tag
Dictionary<string,object> config

```

Adapter wrapper of the toolkit method Create.
Assumes that these objects are BHoMObjects (they must have CustomData and IDs stored) and if their ID has been set up

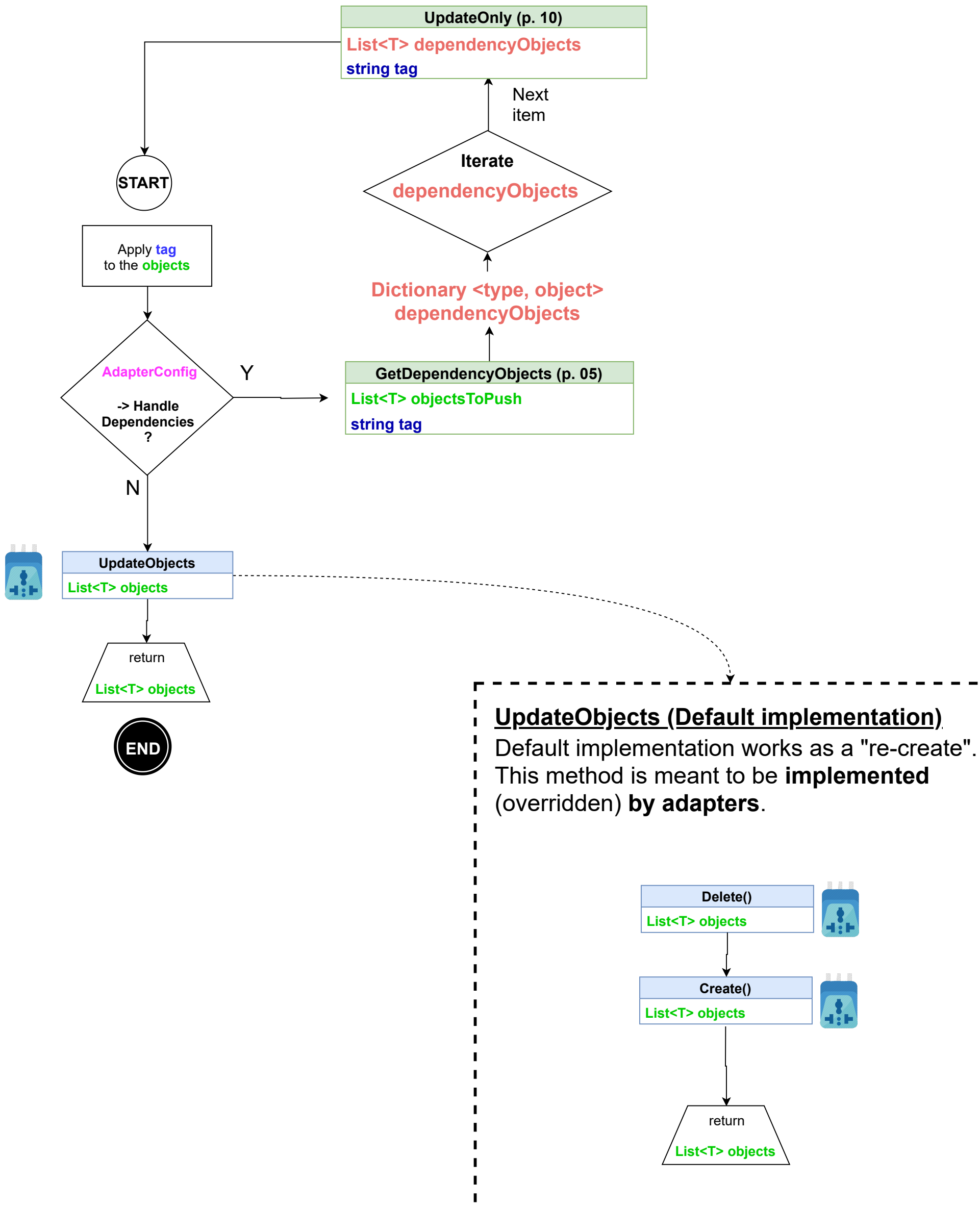


UpdateOnly

```

UpdateOnly
List<T> objects
string tag
Dictionary<string,object> config
    
```

Adapter wrapper of the toolkit method UpdateObjects.
 Assumes that these objects are IBHoMObjects (they must have CustomData and IDs stored) and if their ID has been set up



UpdateProperty

```

UpdateProperty
IRequest
string propertyName
object newValue
Dictionary<string,object> config

```

Default implementation of the method commonly used to update a single property of specified objects (e.g. only the tag).

This method can be overridden by adapters. Not to be confused with the toolkit method UpdateProperty.

