

Cumulative update 7 for Microsoft  
Dynamics® AX 2012 R2

# USING AND MODIFYING THE SALES TAX HIERARCHY FRAMEWORK FOR INDIA

White Paper

This white paper describes how to use and modify the Sales Tax Hierarchy Framework for India.

September 2013

Send suggestions and comments about this document to [adocs@microsoft.com](mailto:adocs@microsoft.com). Please include the title with your feedback.

---

# Content

<b>Introduction .....</b>	<b>3</b>
What is included in this guide.....	3
What is not included in this guide .....	3
<b>Overview of the Sales Tax Hierarchy Framework.....</b>	<b>3</b>
Components of the Sales Tax Hierarchy Framework.....	5
Physical model for the sales tax hierarchy instance.....	5
Object model for the sales tax hierarchy instance.....	7
Sales tax structure and node definition .....	7
Sales tax hierarchy generation.....	9
Tax transaction posting.....	9
<b>Customization of the Sales Tax Hierarchy Framework .....</b>	<b>9</b>
Define a new sales tax structure.....	10
Update the properties of the sales tax structure and its nodes .....	11
Create a new level or a new node type for the sales tax structure .....	13
Update the tax transaction posting validation rule .....	14

## Introduction

In Microsoft Dynamics AX 2012 and earlier versions, sales tax settlement is supported for the following India tax types: excise, value-added tax (VAT), sales tax, and service tax. The capabilities of the supported tax features include the following:

- Offsetting tax amounts during the settlement process in accordance with the setoff priority defined on the tax component
- Settling the supported tax to the tax authority
- Transferring tax amounts among tax components

In cumulative update 7 for Microsoft Dynamics AX 2012 R2, the tax payment functionality has been redesigned to better support the large volume of customers' daily financial transactions and the changing India regulatory requirements. Changes to the application include the following:

- A more standardized approach for supporting tax payments
- A more visualized and easily configurable tax sales tax payment structure
- Performance improvements to the India sales tax payment process
- An extensible design that lets partners and customers easily customize the functionality to meet their business needs

This guide describes the architectural changes to the India sales tax payment features and explains how to customize the features to meet different business requirements.

## What is included in this guide

This guide introduces the Sales Tax Hierarchy Framework, and describes the major changes to the sales tax payment design from both the object model and the physical model perspectives. It also describes, in detail, how and where to customize the sales tax payment code to meet your business requirements. The flexibility of the Sales Tax Hierarchy Framework design lets customers and partners leverage the original code base, reducing the need for significant customization and ensuring that any customizations align with the original design.

## What is not included in this guide

This guide does not explain the following:

- How India tax is calculated and posted in the application
- How sales tax payment is processed per the legal requirements
- How payments are made to the tax authority
- Any features that do not require customization

## Overview of the Sales Tax Hierarchy Framework

To pay or report a tax amount to a tax authority, a legal entity typically provides certain tax-related information. For example, the tax reporting code is used to describe the key information related to the paid tax, such as the tax type or the tax-related transaction type.

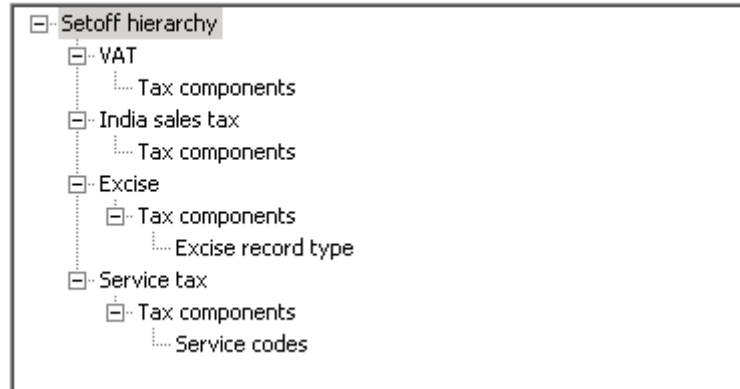
In Microsoft Dynamics AX 2012, the infrastructure for tracking, paying, and recording sales tax settlement changed to a model based on the concept of tax reporting codes. Beginning with Microsoft Dynamics AX 2012 R2 cumulative update 7, India sales tax reporting will adopt the standard Microsoft Dynamics AX approach of using tax reporting codes and will further refine the concept for India sales tax settlements.

India tax reporting codes are a combination of tax-related information, including the following:

- Tax type
- Tax component
- Excise record type (for excise tax-type transactions)
- Service accounting code (for service tax-type transactions)

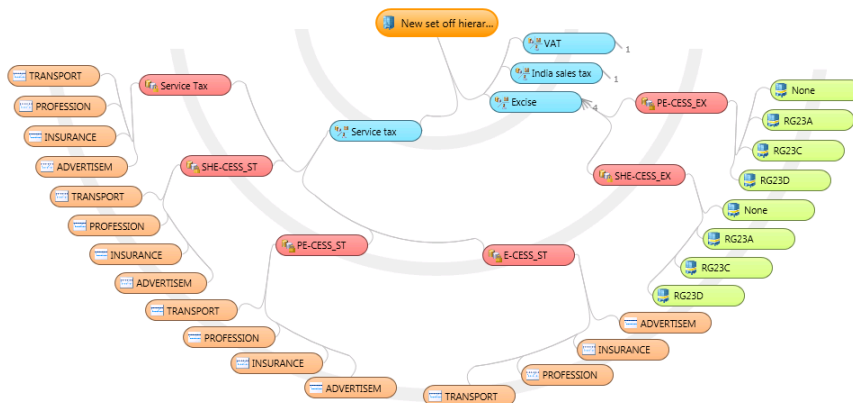
Tax report codes are structured in a setoff hierarchy that you define. You can view the setoff hierarchy in a tree format.

▲ Structure schematic diagram



The setoff hierarchy is an abstract hierarchy that is not populated with data. It can be thought of as a template for a concrete hierarchy. A concrete hierarchy is a tree in which all the nodes are populated from a specified data source.

The Sales Tax Hierarchy view displays an instance of the defined tax report structure, in which each node represents the corresponding tax information, such as the tax component or the excise record type. You can create multiple instances of the tax reporting structure for different purposes.



The sales tax hierarchy instance is generated by the framework based on the definition of the sales tax structure in classes. The data that is used to build the hierarchies is defined either in the application when the user sets up forms, such as the **Service codes** form, or in Microsoft Dynamics AX enums, such as the **Excise record type** enum. Because of this design, the framework is loosely coupled with the data to ensure adaptability and extensibility of the hierarchy.

Based on the sales tax hierarchy, each tax transaction (TaxTrans\_IN) is mapped to one node in the active hierarchy if the transaction needs to be involved in the sales tax settlement. Therefore, all sales tax settlement-related functionality, such as tax transaction posting, sales tax component set off, sales tax settlement, and ad-hoc reporting, uses the same sales tax hierarchy to generate and gather tax settlement information.

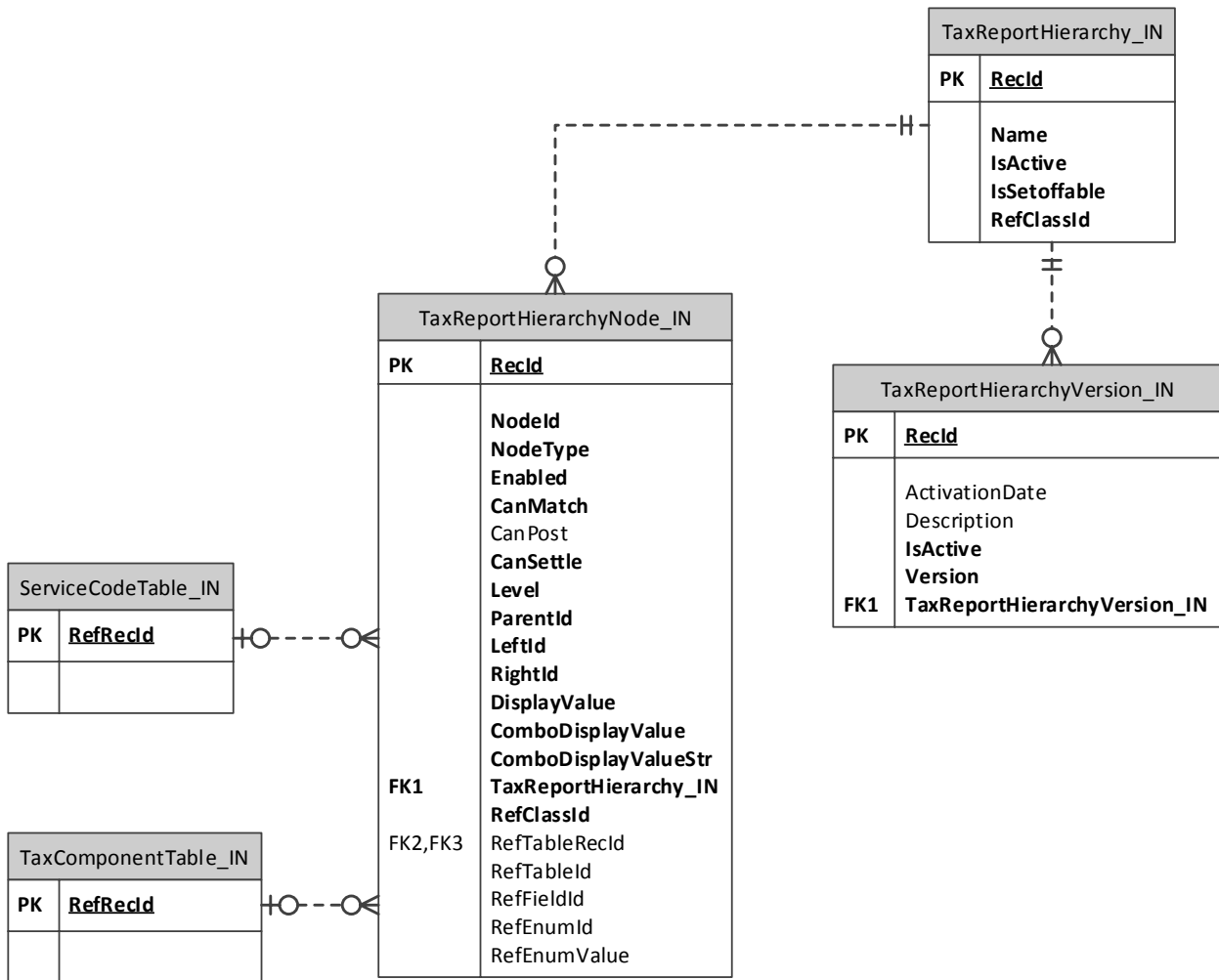
## Components of the Sales Tax Hierarchy Framework

The Sales Tax Hierarchy Framework consists of the following:

- A set of tables that contain the information related to the sales tax hierarchy instance and its node properties
- A set of classes that define the sales tax structure
- A set of classes that are used to generate the sales tax hierarchy instances based on the definition of the sales tax structure
- A set of classes that validate and control the tax transaction posting

## Physical model for the sales tax hierarchy instance

The following physical model defines the sales tax hierarchy instance and its node properties.



- The TaxReportHierarchy\_IN and TaxReportHierarchyVersion\_IN tables contain the basic information about the sales tax hierarchy instance.
- The TaxReportHierarchyNode\_IN table contains the information for each node in the sales tax hierarchy instance.
- The ServiceCodeTable\_IN and TaxComponentTable\_IN tables are existing tables that are referred to by the sales tax hierarchy nodes.

Although most instance and node properties are set automatically by the framework or defined sales tax structure, three properties determine the major logic of the tax transaction posting and the tax transaction settlement:

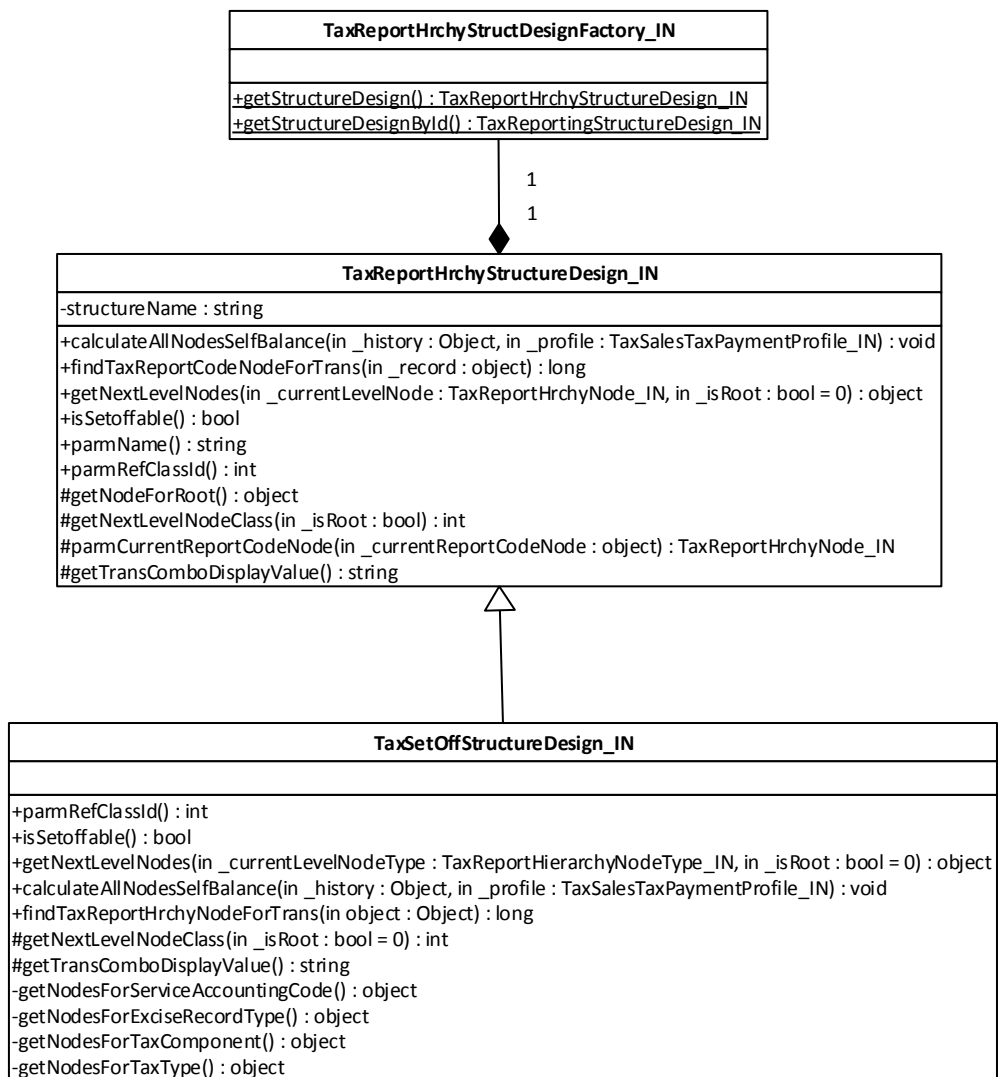
- **CanMatch** – This property indicates whether the tax transaction, with the tax-related information, can reference the node.
- **CanSettle** – This property indicates whether the node will be involved in the tax settlement.
- **CanPost** – This property indicates whether the node will be used to summarize the tax amount and generate the vendor payment journal to the tax authority or the carry forward journal.

## Object model for the sales tax hierarchy instance

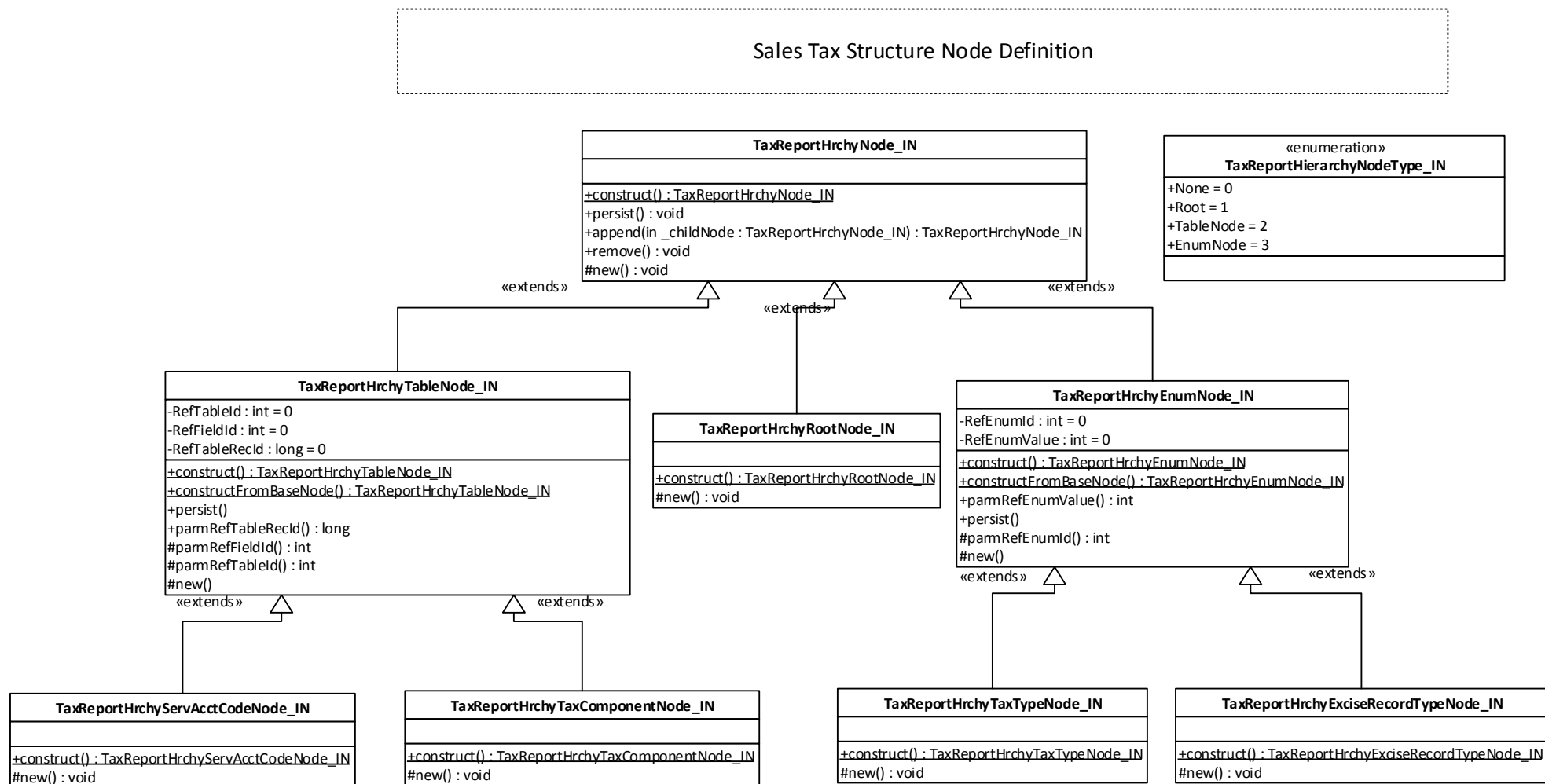
### Sales tax structure and node definition

A set of sales tax structure design classes contain the definitions of the structures that will be used to generate the sales tax hierarchy instances. The **TaxSetoffStructureDesign\_IN** structured class is predefined. It conforms to India regulatory requirements and also supports common business scenarios. This structured class can be used to build the sales tax hierarchy instance by using the provided form UI.

#### Sales Tax Structure Definition



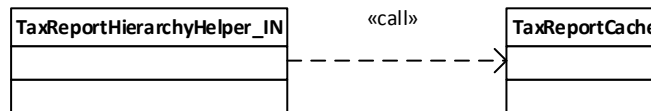
A set of tax report hierarchy node classes contain the definitions of the nodes that compose the sales tax hierarchy instances. The **TaxReportHrchyServAcctCodeNode\_IN**, **TaxReportHrchyTaxComponentNode\_IN**, **TaxReportHrchyTaxTypeNode\_IN**, and **TaxReportHrchyExciseRecordTypeNode\_IN** node classes are predefined. These node classes generate the hierarchy instances in accordance with the sales tax structure definition.





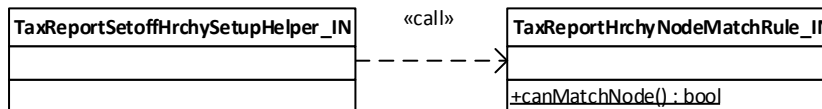
## Sales tax hierarchy generation

A set of classes perform the major operations of the sales tax hierarchy, such as creating, reviewing, updating, and deleting the hierarchy and its nodes in accordance with the sales tax structure definition. Certain operations interact with the cached data to improve performance. This is primarily framework-driven and does not require customization.



## Tax transaction posting

The tax transaction will not be posted if it does not provide enough required tax-related information. For example, if an excise tax transaction does not provide information about the excise record type, the framework will block the transaction and provide an error message. However, certain types of tax transactions do not require an association with a hierarchy node.



## Customization of the Sales Tax Hierarchy Framework

The Sales Tax Hierarchy Framework lets users define structures for setoff or reporting purposes. To completely define a structure, you need to determine the following:

- Whether the structure will be used for set-off. It is important to know whether the structure can be used for setoff. India has complex requirements concerning how input tax can be used to set off output tax during tax settlement.
- The number of levels of the structure.
- The properties of each node in the structure. These properties, the data source, can be matched, can be set off, and so on, are used by default for the corresponding nodes in concrete hierarchies.
- How the structures will be presented to end users. The schematic diagram of the structure is used for this purpose.

If the provided sales tax functionality meets your business needs, it is not necessary to customize the Sales Tax Hierarchy Framework. However, if customizations are required, the following sections explain how to customize the application. Use the following list of checkpoints to evaluate whether customizations are necessary. Customizations are required if you need to do the following:

- Permanently change the node properties in the sales tax hierarchy for tax posting and settlement purposes
- Change the sales tax structure, such as by adding, deleting, or updating a level in the structure
- Update the validation rule for tax transactions, so that some transactions do not need to be associated with a sales tax hierarchy node

**Important:** We strongly recommend that you create your own sales tax structure definition class instead of updating the existing class.

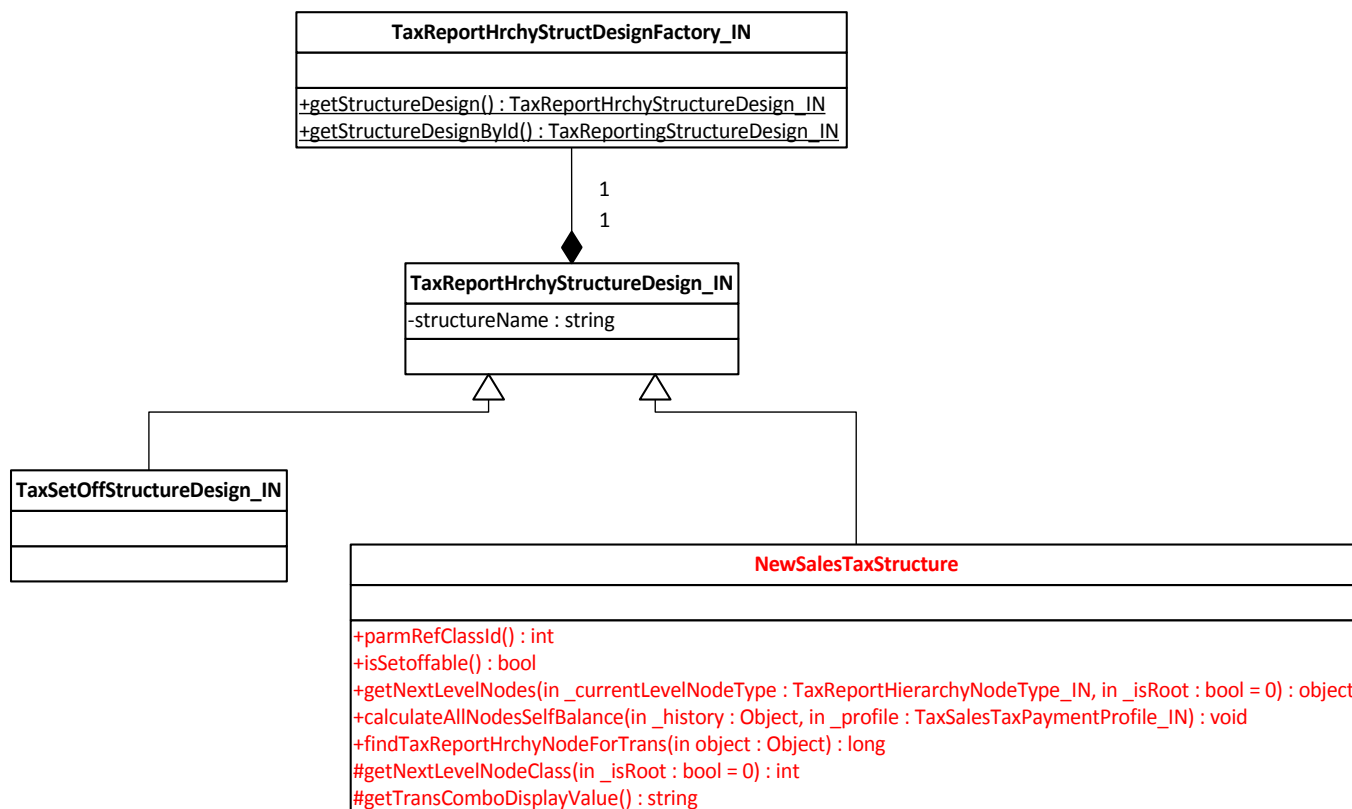
## Define a new sales tax structure

To create a new sales tax structure, define a new enum value in the **TaxReportHrchyStructDesignType\_IN** enum for the new class. The new enum value will be used as a class attribute to enable the framework to recognize the new structure.

```
[TaxReportHrchyStructureAttribute_IN(TaxReportHrchyStructDesignType_IN::NewDesign)]  
public class NewSalesTaxStructure extends TaxReportHrchyStructureDesign_IN  
{  
}
```

The new class should extend the **TaxReportHrchyStructureDesign\_IN** base class and implement the following class methods:

- **parmRefClassId()** – Returns the classNum of the new class
- **isSetoffable()** – Indicates whether the hierarchy is used for tax settlement purposes
- **getNextLevelNodes()** – Populates the concrete child nodes for the specified parent node in the hierarchy
- **calculateAllNodesSelfBalance()** – Defines how to get and calculate the tax amount from a set of valid tax transactions that are associated with the node
- **findTaxReportHrchyNodeForTrans()** – Defines the tax-related information that is required, and how the information is combined to find the hierarchy node for the tax transaction
- **getNextLdevNodeClass()** – Defines the levels of the structure; for example, the tax component type node follows the tax type node in the structure
- **getTransComboDisplayValue()** – Defines how to determine the combination display value for the tax transaction based on its tax-related information

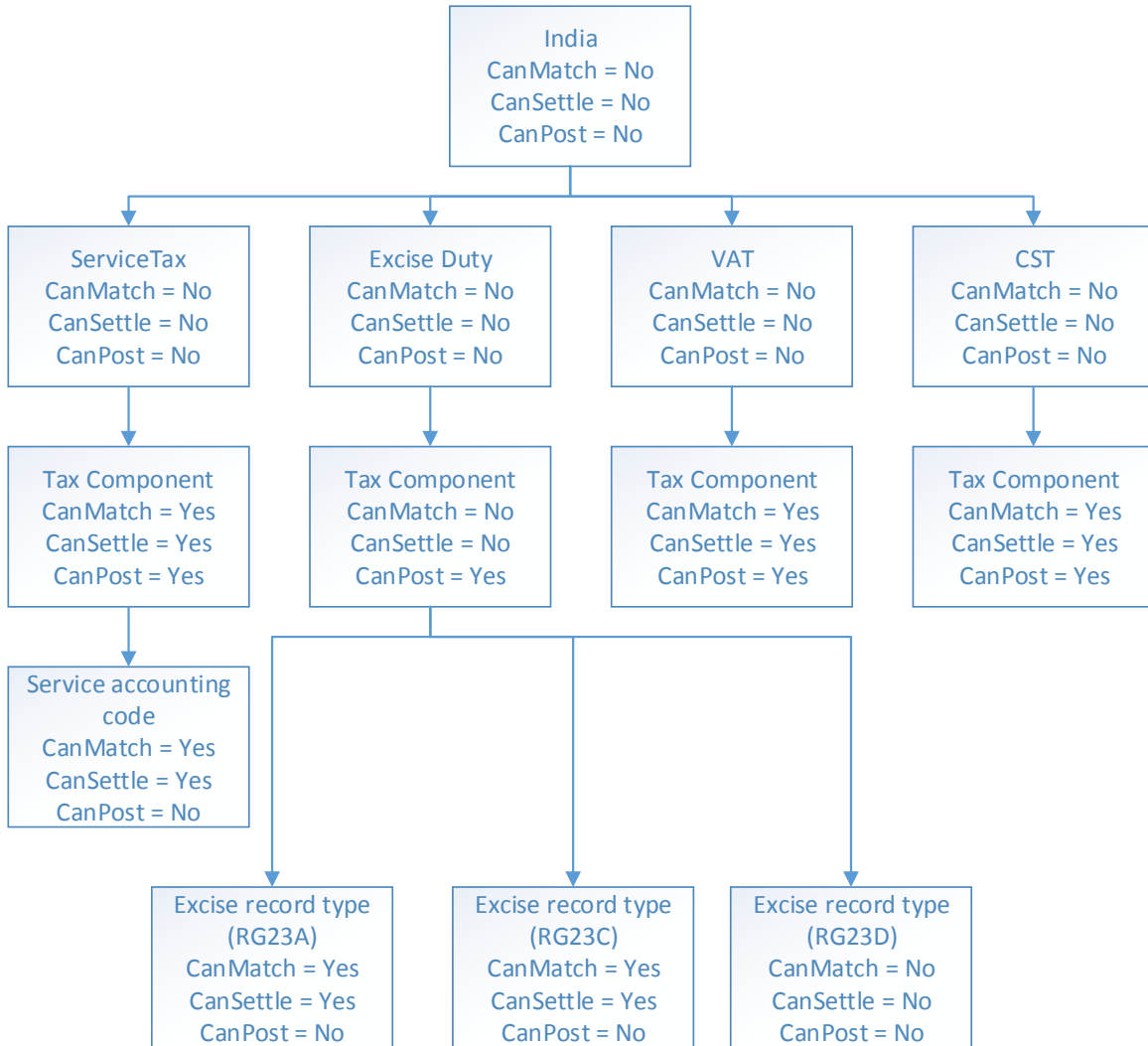


## Update the properties of the sales tax structure and its nodes

Certain properties of the sales tax structure and its nodes determine how a tax transaction will be validated during posting. For example, the properties determine whether a transaction will be included in the tax settlement and how the payment journal is generated during the settlement process. The following properties are the most common properties that you might need to update to suit your business requirements:

Table	Field	Purpose
TaxReportHierarchy_IN	IsSetoffable	Indicates whether the sales tax hierarchy can be used for tax settlement purposes
TaxReportHierarchyNode_IN	canMatch	Indicates whether the tax transaction, with the tax-related information, can have a reference to this node
TaxReportHierarchyNode_IN	canSettle	Indicates whether this node will be involved in the tax settlement process
TaxReportHierarchyNode_IN	canPost	Indicates whether this node will be used to summarize the tax amount, and generate the vendor payment journal to the tax authority or the carry forward journal

These properties are predefined for the **TaxSetoffStructureDesign\_IN** sales tax hierarchy class. For **TaxSetoffStructureDesign\_IN**, the **isSetoffable** property is set to **True**, because this sales tax hierarchy will be used for tax settlement purposes. The default values for the **Can** properties are shown in the following illustration. The default **Can** property values differ, depending on the node type.

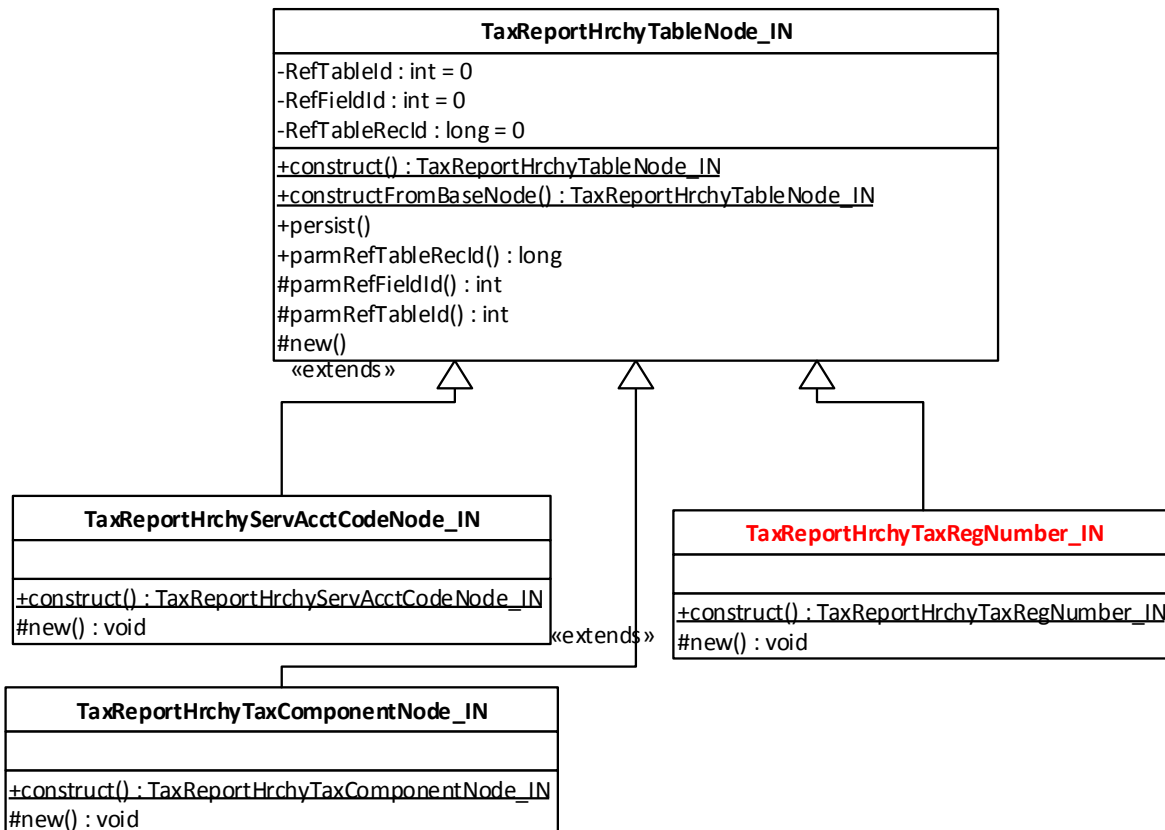


To update the property values, extend the **TaxSetoffStructureDesign\_IN** base class, and, in the derived class, override the default values to meet your business needs.

## Create a new level or a new node type for the sales tax structure

If the sales tax structure needs to have different levels of node types or needs to include more node types that are not yet defined, you must create a new node type class and define the structure definition class of the level structure.

For example, if you want to add one node type, **TaxRegistrationNumber**, between the **TaxType** node type and the **Tax component** node type, you need to create a new node type class as shown in the following illustration. In this example, **TaxReportHrchyTaxRegNumber\_IN** extends the **TaxReportHrchyTableNode\_IN** base class.



In the new tax node class, only certain default values need to be specified in the new method, to indicate the table from which the data should be obtained.

```

protected void new()
{
    super();

    //Init the default value
    this.parmClassId(classNum(TaxReportHrchyTaxRegNumNode_IN));
    this.parmRefTableId(tableNum(TaxRegistrationNumbers_IN));
    this.parmRefFieldId(fieldNum(TaxRegistrationNumbers_IN, RegistrationNumber));
}
  
```

The next step is to create a new class for the new sales tax structure, and to define the levels and node properties of the structure.

```
protected ClassId getNextLevelNodeClass(boolean _isRoot = false)
{
    ClassId nextLevelClass;
    TaxReportHrchyTaxComponentNode_IN componentNode;
    TaxComponentTable_IN taxComponentTable;

    if(_isRoot)
        return classNum(TaxReportHrchyRootNode_IN);

    switch(this.parmCurrentNode().parmClassId())
    {
        case classNum(TaxReportHrchyRootNode_IN):
            nextLevelClass = classNum(TaxReportHrchyTaxTypeNode_IN);
            break;

        case classNum(TaxReportHrchyTaxTypeNode_IN):
            nextLevelClass = classNum(TaxReportHrchyTaxRegNumNode_IN);
            break;

        case classNum(TaxReportHrchyTaxRegNumNode_IN):
            nextLevelClass = classNum(TaxReportHrchyTaxComponentNode_IN);
            break;
    }
}
```

After the node type and structure class are correctly defined, the framework will generate a new sales tax hierarchy and populate the data automatically.

## Update the tax transaction posting validation rule

Not all the tax transactions need to be associated with a sales tax hierarchy node for settlement purposes. For example, a tax transaction that records a tax amount to expense does not need to be included in the settlement process.

By default, the setoff able sales tax hierarchy only summarizes the tax transactions that pertain to recoverable tax and payable tax. A matching rule is then defined in the **TaxReportHrchyNodeMatchRule\_IN::canMatchNode** class method.

You can update this static method if a specific validation step needs to be included or excluded during the tax transaction posting process.

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989

Worldwide +1-701-281-6500

[www.microsoft.com/dynamics](http://www.microsoft.com/dynamics)

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. The information contained herein is not intended to constitute legal, tax, accounting or financial advice or to be used as a substitute for specific advice from a channel partner or other professional. You should not act (or refrain from acting) based on information in this document without obtaining professional advice about your particular facts and circumstances.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Although Microsoft may refer to its channel partners as "partners", they are independent entities. There is no partnership, joint venture, agency or franchise relationship or fiduciary duty between Microsoft and its channel partners. Channel partners are solely responsible for any configurations, customizations, localizations or translations they create or implement on behalf of their customers, including any support or other service they provide to customers for such solutions. Microsoft does not warrant or guarantee partner-created solutions and disclaims any and all liability arising out of any partner-created solution or service.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2013 Microsoft Corporation. All rights reserved.

