

Ericsson OSS BSS Education Services

# Notices

Trademark Acknowledgements

Ericsson and the Ericsson logo and product names are registered trademarks of Ericsson Inc. Oracle is a registered trademark of Oracle Corporation. SUN and Solaris are registered trademarks of Sun Microsystems, Inc. UNIX is a registered trademark of The Open Group. Microsoft®, Microsoft Windows®, and Windows NT® are registered trademarks of Microsoft Corporation. Google and Google Web Toolkit are trademarks of Google, Inc. and Weblogic Server is a registered trademark of Oracle Corporation. HP, HP-UX and Hewlett-Packard are trademarks or registered trademarks of Hewlett-Packard Development Company. L.P. Adobe is a registered trademark of Adobe Systems Incorporated.

Ericsson notice of disclaimer and limitation of liability

This document is intended for use solely for training purposes and only by Ericsson customers who have licensed the Ericsson software described herein. The software, this document, and the information contained within this document may be used, copied or communicated only in accordance with the terms of a written license agreement with Ericsson. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording without the prior written permission of Ericsson. While the information contained herein has been prepared from sources deemed reliable, Ericsson reserves the right to revise the information without notice, but has no obligation to do so. Unless the recipient has been expressly granted a license by Ericsson under a separate applicable written agreement with Ericsson, no license, express or implied, is granted under any patents, copyrights or other intellectual property rights. Use of the information contained herein is in your sole determination and shall not be deemed an inducement by Ericsson to infringe any existing or later-issued patent, copyright or other intellectual property rights.

ERICSSON PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS. FURTHER, ERICSSON MAKES NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED WITH RESPECT TO THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN. ERICSSON EXPRESSLY ADVISES THE USER THAT ANY USE OF OR RELIANCE UPON SAID INFORMATION OR OPINION IS AT THE SOLE RISK AND LIABILITY, IF ANY, OF THE USER AND THAT ERICSSON SHALL HAVE NO LIABILITY FOR ANY ERROR OR DAMAGE OF ANY KIND RESULTING FROM THE USE OF THIS DOCUMENT INCLUDING BUT NOT LIMITED TO ANY DAMAGE OR INJURY INCURRED BY ANY PERSON ARISING OUT OF THE SUFFICIENCY, ACCURACY, OR UTILITY OF ANY INFORMATION OR OPINION CONTAINED HEREIN.

Ericsson does not recommend or endorse products and nothing contained herein is intended as a recommendation or endorsement of any product.

This document is a training document and contains simplifications. Therefore, it must not be considered as a specification of the software or system. This document is not intended to replace the technical documentation that was shipped with your software or system. Always refer to that technical documentation during operation and maintenance.

Restricted rights notice

The software, documentation, and/or data delivered herewith are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable.

© Ericsson Inc. 2013

**Contents**

[Notices 2](#_Toc409616974)

[1 About this Document 5](#_Toc409616975)

[1.1 Description 5](#_Toc409616976)

[1.2 Intended audience 5](#_Toc409616977)

[2 Notes About the Content 6](#_Toc409616978)

[2.1 Objectives and content structure 6](#_Toc409616979)

[2.2 Additional help 7](#_Toc409616980)

[3 Module 1: Catalog Manager Overview 8](#_Toc409616981)

[4 Module 2: Catalog Manager Configuration 9](#_Toc409616982)

[4.1 Exercise 1: Catalog configuration 9](#_Toc409616983)

[4.1.1 Create an organizational chart (top level) 9](#_Toc409616984)

[4.1.2 Add subsidiary nodes to the chart 13](#_Toc409616985)

[4.1.3 Assign users to chart positions 14](#_Toc409616986)

[4.1.4 Configure default currency, markup and date 16](#_Toc409616987)

[5 Module 3: Catalog Designer 19](#_Toc409616988)

[5.1 Exercise 2: Introducing Catalog Designer 19](#_Toc409616989)

[5.1.1 Benefits of using Catalog Designer 19](#_Toc409616990)

[5.1.2 Catalog Designer overview 20](#_Toc409616991)

[5.1.3 Navigate the GUI 21](#_Toc409616992)

[5.1.4 Find information in the GUI 25](#_Toc409616993)

[5.2 Exercise 3: Scenario overview and new project 27](#_Toc409616994)

[5.2.1 Scenario overview for Catalog Manager exercises 27](#_Toc409616995)

[5.2.2 Create new project and add detail 28](#_Toc409616996)

[5.2.3 Open project 30](#_Toc409616997)

[6 Module 4: Code Tables and Attribute Types 32](#_Toc409616998)

[6.1 Exercise 4: Code tables and attribute types 32](#_Toc409616999)

[6.1.1 Create code table 32](#_Toc409617000)

[6.1.2 Export and view code tables 34](#_Toc409617001)

[6.1.3 Create bandwidth and string attribute types 37](#_Toc409617002)

[7 Module 5: Component Items and Associations 41](#_Toc409617003)

[7.1 Exercise 5: Component items and bandwidths 41](#_Toc409617004)

[7.1.1 Create component item and add detail 41](#_Toc409617005)

[7.1.2 Create bandwidth attributes, and associate with component item 45](#_Toc409617006)

[7.1.3 Create more components items, and add attribute restrictions 49](#_Toc409617007)

[7.2 Exercise 6: Item relations and groups 56](#_Toc409617008)

[7.2.1 Create association type and relate items 56](#_Toc409617009)

[7.2.2 Create item group and add members 64](#_Toc409617010)

[8 Module 6: Catalog Hierarchy 68](#_Toc409617011)

[8.1 Exercise 7: Catalog hierarchy and testing 68](#_Toc409617012)

[8.1.1 Create catalog hierarchy 68](#_Toc409617013)

[8.1.2 Add item to hierarchy node 72](#_Toc409617014)

[8.1.3 Test catalog hierarchy 76](#_Toc409617015)

[9 Module 7: Pricing 82](#_Toc409617016)

[9.1 Exercise 8: Charge types 82](#_Toc409617017)

[9.1.1 Create charge types 82](#_Toc409617018)

[9.1.2 Associate charge types with items 87](#_Toc409617019)

[9.1.3 Test basket pricing 90](#_Toc409617020)

[10 Module 8: Context Attributes and Rules 93](#_Toc409617021)

[10.1 Exercise 9: Context, context attributes and rules 93](#_Toc409617022)

[10.1.1 Create new context and context attributes 93](#_Toc409617023)

[10.1.2 Create rule and associate with context attribute 97](#_Toc409617024)

[11 Module 9: Conditional Charges 102](#_Toc409617025)

[11.1 Exercise 10: Conditional charges and testing 102](#_Toc409617026)

[11.1.1 Create new rule and associate with charge type 102](#_Toc409617027)

[11.1.2 Test rule logic 107](#_Toc409617028)

[11.2 Extension exercise 11 (OPTIONAL): Introduce setup discount for residential customers 110](#_Toc409617030)

[11.3 Extension exercise 12 (OPTIONAL): Resolve subscription charge issue 111](#_Toc409617031)

# About this Document

The two sections which follow provide a summary description of this ‘*Ericsson Catalog Manager (ECM100)*’ course and the intended audience for whom it was produced. This one-day course is the first component of the five-day *‘Ericsson Basic Catalog Manager/Order Care’* training week, leading to (if required) *‘Ericsson Advanced Catalog Manager/Order Care’* training week.

## Description

This document provides detailed information about the topics discussed in this course, and instructions for how to carry out the prescribed activities related to those topics. The course content will be managed by the instructor, who will ask students to follow demos provided by the instructor and work on predefined activities in this exercise document, sometimes with detailed document support, and sometimes with only high-level guidance. The entire document will also be available to students after the course, along with a copy of the presentation slides used by the instructor.

The information contained in this document is the sole property of Ericsson and should not be reproduced or distributed in any form without prior written permission.

## Intended audience

The ‘*Ericsson Catalog Manager (ECM100)*’ training course is intended for anyone wishing to have an introduction and overview of the ‘Ericsson Catalog Manager’ product, and is suitable for people in many different roles such as Managers, End Users, Administrators, Project Managers, Solution Architects, Developers, Database Administrators, etc.

A basic knowledge of BSS applications and processes (such as billing and charging) is desirable but not essential.

Notes About the Content

The following notes summarize the content structure, describing its main objectives. They also provide a few guidelines on how best to use this document.

## Objectives and content structure

The main objectives of this course are to:

* Do some basic catalog configurations
* Learn how to navigate the Catalog Designer GUI
* Work with design projects
* Work with code tables and attribute types
* Create bandwidths and component items
* Work with item relations and groups
* Create and test a new catalog hierarchy
* Create product items and groups
* Work with contexts and context attributes
* Understand and use pricing
* Work with conditional charges

The document’s contents break down into the following logical structure:

* Catalog configuration
  + Creating an organizational chart
  + Assigning users to chart positions
  + Configuring default settings
* Getting to know the Catalog Designer GUI
  + Benefits of using Catalog Designer
  + Navigating the GUI
  + Finding information in the GUI
* Exercise scenario and project build
  + Overview of exercise scenario
  + Creating a project and adding detail
  + Opening a project
* Code table and attribute types
  + Creating and exporting code tables
  + Creating attribute types
* Bandwidths and component items
  + Creating bandwidths
  + Relating bandwidths to attribute types
  + Creating component items with detail
  + Restricting attribute values
  + Viewing hierarchy
* Item relations and groups
  + Creating association type
  + Relating items to association type
  + Creating item group with detail
* Catalog hierarchy
  + Creating nodes
  + Adding items to nodes
  + Testing
* Pricing
  + Creating new charge types
  + Associating charge types with items
  + Testing basket pricing
* Contexts
  + Creating contexts and context attributes
  + Creating catalog rules
  + Associating rules with context attributes
* Conditional charges
  + Creating new rule
  + Associating rule with charge type
  + Testing

## Additional help

The following guidelines aim to show how and when this document should be used, and tries to help students be efficient and productive in their work:

* The exercises will be done at the discretion of the instructor, who will decide at which points of the course presentation they will be addressed
* The additional information presented in this document should be studied in support of the exercise activity. Students should be given time during the course to read this information, and practice the described processes in the GUI
* Some of the exercises are intended to be done by students entirely unsupported by the instructor. The instructor will be available to assist, but students should try to work these without help
* Once the exercises in this document are completed, the knowledge gained can be used in assisting with real-life projects, including customized products and processes
* For help with any of the material included in this document, email oss.bss.education@ericsson.com

Module 1: Catalog Manager Overview

There are no exercises associated with Module 1.

Module 2: Catalog Manager Configuration

In this exercise only a small amount of configuration is done in the **Administration** GUI as a prerequisite for our user-based exercise activity. Exercise 1 covers this configuration activity, whilst all the remaining exercises focus on user issues in the **Catalog Designer** GUI.

## Exercise 1: Catalog configuration

In this exercise we perform some simple configurations required before starting work on a new project:

* Creating an organizational chart (top level)
* Adding subsidiary nodes
* Assigning users to different chart positions
* Configuring default currency, markup and date

*Note: In order for us to be able to log in and use the Catalog Designer, project metadata must already exist. Two Designer Projects have therefore been set up in the system to facilitate this training.*

### Create an organizational chart (top level)

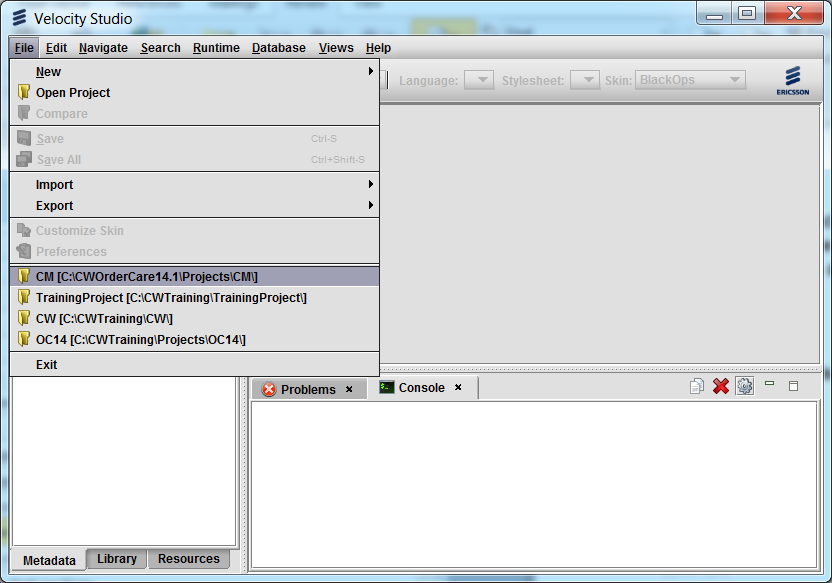
The **organizational chart** is a collection of different teams, departments or roles, structured in a hierarchical tree to indicate the different positions within the organization.

An organizational chart is created in the **User Profile Manager** of the **Administration** GUI.

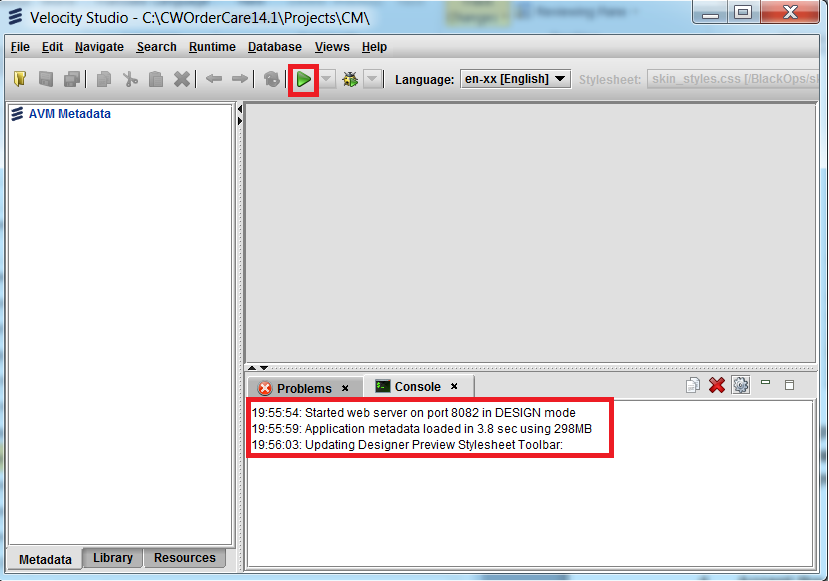
1. Log in to the **Velocity** **Studio** application from the shortcut icon on the desktop, or from the **Programs** menu: *ConceptWave > ConceptWave Designer* (no username/password required).

*Note: Velocity Studio is an advanced user-interface design tool for rapidly creating and configuring applications, processes, business logic, order and data models.*

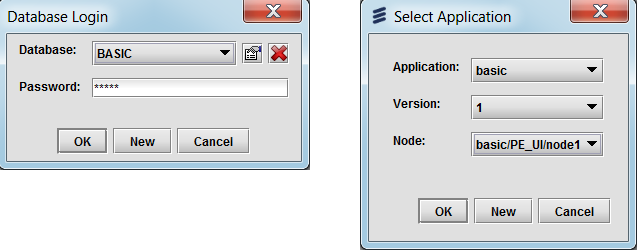
1. Open the existing (and previously opened) designer project called ‘**CM’** from the **File** menu or use the *File > Open Project* menu option, navigating to the project folder (*C:\CWOrderCare14.1\Projects\CM*):

**

1. Once the project is open, note that the log entries in the **Console** tab at the bottom of the screen indicate that the project is in **DESIGN** mode. Start the web server (Framework) by clicking the **Start runtime** green arrow in the toolbar:

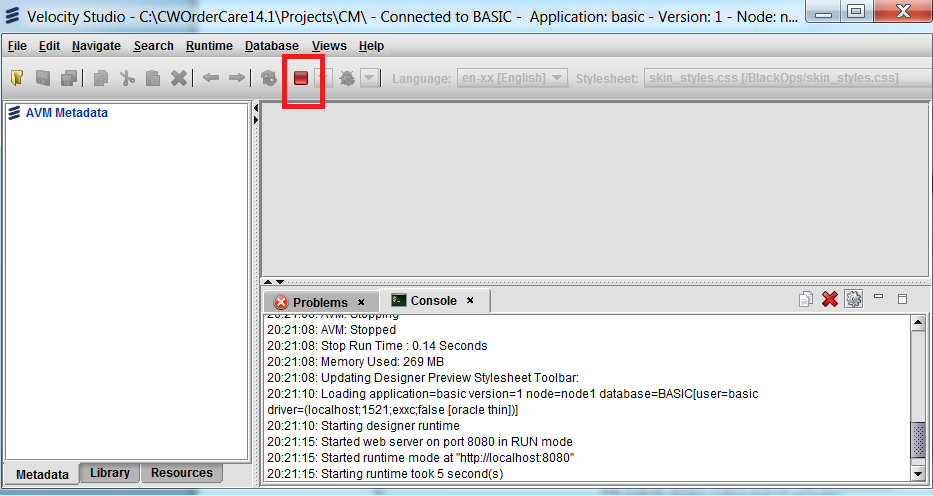


1. Accept the default login credentials for the database which supports this project (click **OK**):



(If this process fails, it means that the configuration is missing or incomplete).

Once the web server is started, the **Console** tab should indicate that the project is in **RUN** mode, and the **Start runtime** green arrow is now a **Stop runtime** red square:



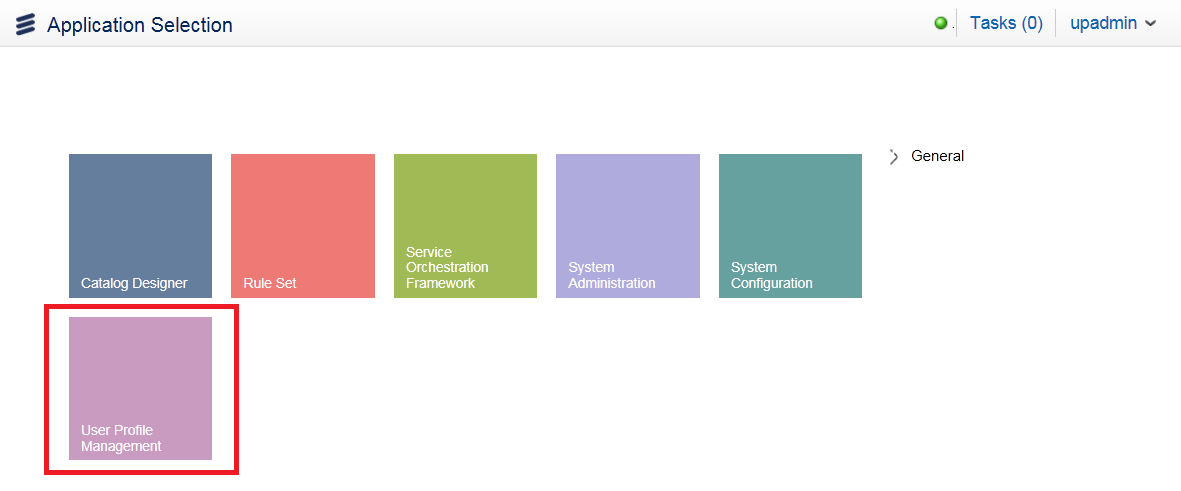
1. Log into Catalog Designer using the following URL and login credentials:
   * **URL** = ‘http://localhost:8081/cwf’
   * **Username** = ‘upadmin’
   * **Password** = ‘upadmin’

(The **upadmin** profile is the sole user defined in the out-of-the-box’ software release).

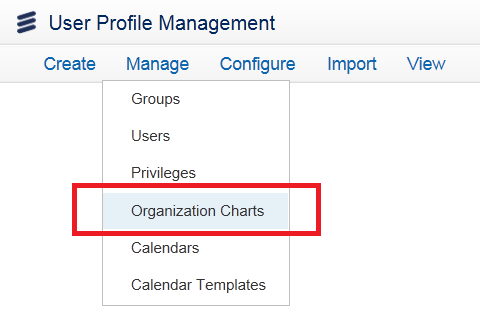
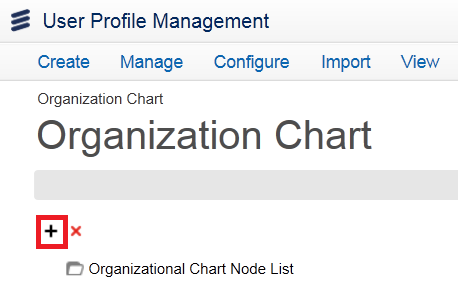


*Note: There are two* ***GUIs*** *used in this exercise:* ***Administration*** *and* ***Catalog Designer****. We will be making a few configuration changes in the* ***Administration*** *GUI now, but thereafter will be focusing on user activity in the* ***Catalog Designer****.*

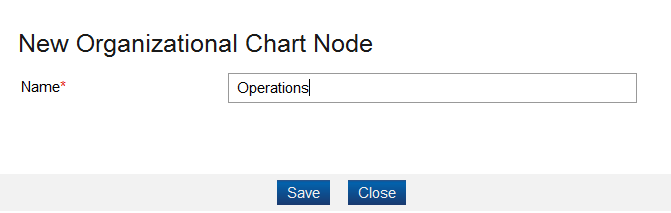
1. Select **User Profile Management app** by clicking on the **User Profile Management button** on the **Application** Selection screen:

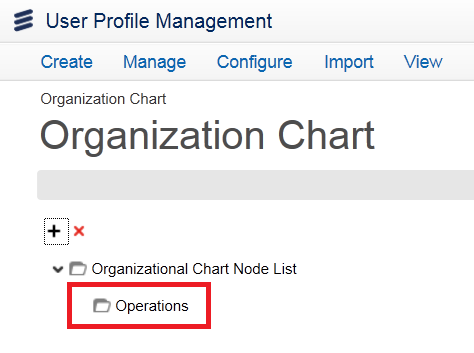


When the **User Profile Manager** app opens, click on **Manage/Organizational Charts**, and finally on the **+** button which appears at the top of the left navigation bar.

1. In the resulting **Org Chart Detail** box, type the name ‘**Operations**’. This represents the top level (node) of our organizational structure.





*Note: If you need to delete any node in the chart, simply select the item and use the* **X***.*

1. Expand the tree to view and validate the new entry.

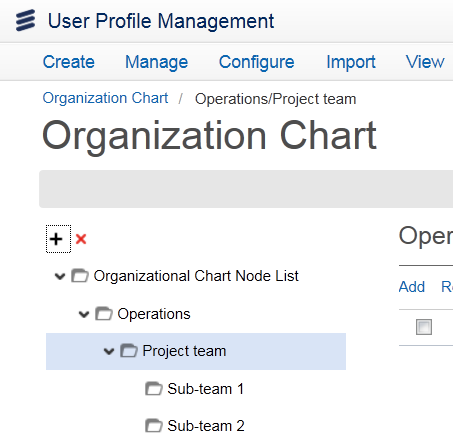
### Add subsidiary nodes to the chart

1. Click on **Organizational Chart** underneath **Operations**, and use the **Add** button once more to add a *secondary* level to **Operations**, and then two more nodes at a *tertiary* level, thus producing a typical example of an organizational structure which can be used in projects.

Use the following definition to complete your organizational chart build:



1. When done, expand your tree and check that your chart matches the screenshot below.

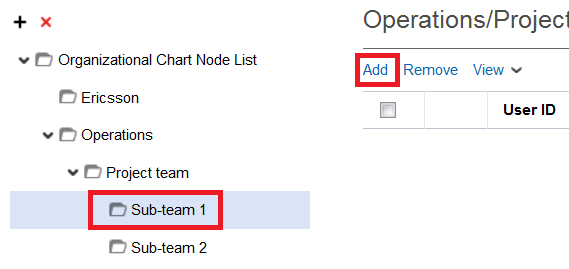


### Assign users to chart positions

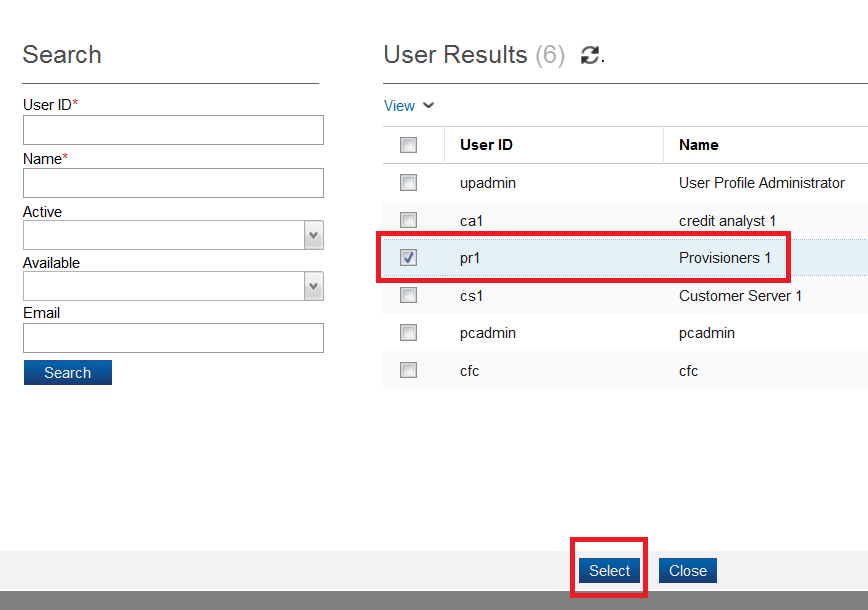
Users can be assigned to any one of the positions defined in the organizational chart defined above. In the catalog you can also associate any catalog object with such a position. This means that only the users assigned to a *particular position* (and their managers) will be able to maintain that catalog object.

Follow the instructions below to add users to the organizational chart positions you created above.

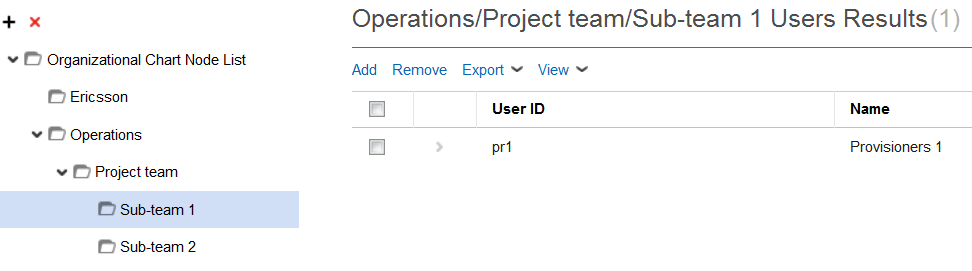
1. Highlight Sub-team 1 and click on Add in center pane:



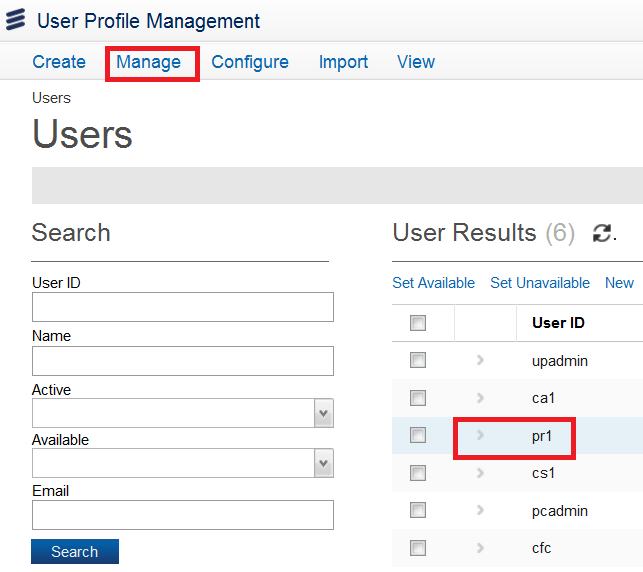
1. In the User pop-up window, select the **Provisioner 1** user and click Select.

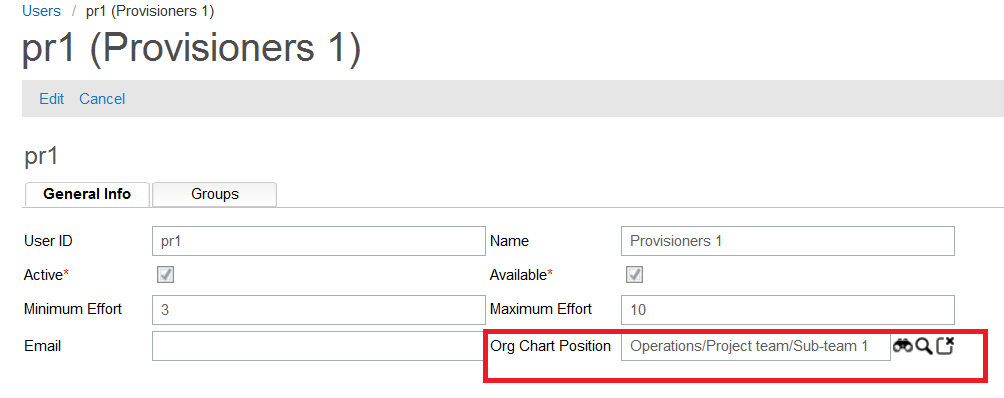
**

1. The **Provisioner 1** user is now associated with **Sub-team 1** in your chart. To view this association at any time, you can click on the team and the user appears in the center pane:



1. You can also see a particular user’s org position by clicking on Manage/Users and select particular user by clicking on >. You will see:





1. Using the same process, add further users to your chart positions, as follows:

* Customer Server 1 🡪 Sub-team 2
* Credit Analyst 1 🡪 Project team
* User Profile Administrator 🡪 Operations

### Configure default currency, markup and date

Three universal attributes will be configured in this section, and these values can be applied to all projects created in the Catalog Designer GUI:

* **Default Currency**: In the case of no currency being specified at the *element* level, an element will

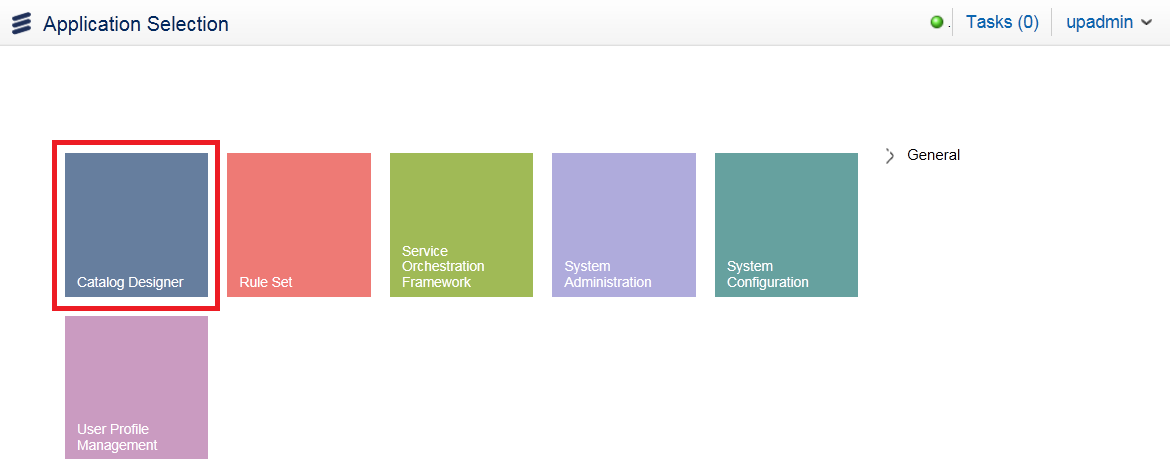
default to this currency. (To be discussed later).

* **Default Markup**: In the case of no values being given at the element level, an element will default

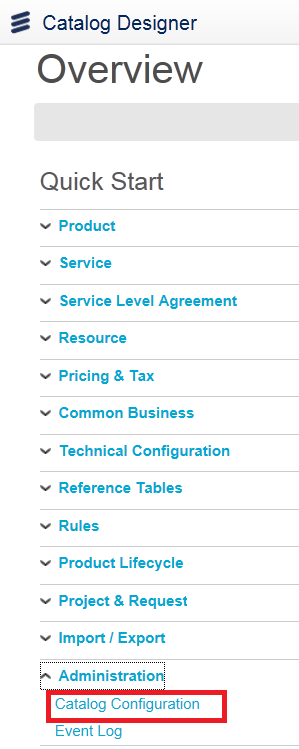
to this value. (To be discussed later).

* **Default Catalog Date**: Select a default date when the **Currency** and **Markup** values take effect.

1. Click on **upadmin/Switch** menu at the top left. Select **Catalog Designer**:



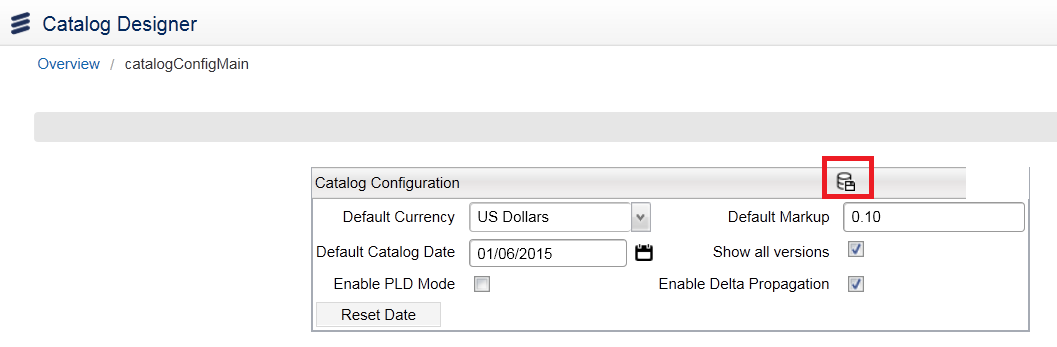
1. When the Catalog Designer window opens, go to the **Administration** menu and choose **Catalog Configuration**:



1. In the Catalog Configuration window you can enter the **Default Currency**, **Default Markup** and **Default Catalog Date** values, as required by your project. Enter the following values (leaving other values unchanged):

* **Default Currency**: ‘US Dollars’
* **Default Markup**: ‘0.10’
* **Default Catalog Date**: ‘12/05/2014’ (the instructor will advise)

**Click Save icon at top right**



1. Click on upadmin *> Logout* to log out of the Catalog Designer application:



Module 3: Catalog Designer

Having briefly dipped into the Catalog Designer GUI in Module 2, you are now given the opportunity in this Module to take a more considered view of what this application is, what it does, and what tools it provides to enable you to efficiently manage the service catalog. Some simple exercise activity is also provided to facilitate your exploration of the Catalog Designer GUI.

## Exercise 2: Introducing Catalog Designer

In this exercise we spend some time learning about the basic functions and navigational tools associated with the **Catalog Designer** GUI, followed by some ‘search and find’ questions to give you some practice in looking for specific objects and information in the GUI.

Before this, however, the benefits of using Catalog Designer are briefly summarized, and a quick overview of its *functions* and *elements* is presented.

### Benefits of using Catalog Designer

The information in this section is for background reading and for your interest. The instructor may prefer that you read this information later if time is at a premium.

There are many benefits of using Catalog Designer. Here are the main ones.

**Faster time to market**

Tight integration with the telecom's *Order Entry* and *Order Management* systems creates a catalog environment in which new products can be quickly added or changed with minimal IT expertise, resulting in reduced development costs and a faster time to market.

**Resource-based view**

Each product in the Catalog consists of a set of resources that are required to fulfill that product. Resources (components) may reflect a *physical* resource such as a set-top box, or a *logical* resource such as an SVC.

**Defining eligibility**

The catalog provides a ‘one-stop’ solution for defining both the general market availability of each product, (e.g. by market segment or by geographic location), as well as the product's specific ‘eligibility’ rule, such as a subscription to another product.

**Linking product components and workflow**

Traditionally, driving a product's fulfillment consists of defining a set of product workflows, and selecting the appropriate workflows to instantiate based on the product, market, rollout date, availability of underlying resources, supplier selection, and more. These two components have been traditionally hardwired into the logic of order management systems, resulting in an overly-complex and increasingly inefficient environment after several years of operation and change. ConceptWave's catalog-driven fulfillment approach relies on the power of a catalog to perform these functions, including provisioning a product for a target market, with a specific due date and supplier.

**Telecom context**

The main restriction in all catalogs is the rules engine's ability to implement business rules. In the telecommunications domain the key constraint is the rules engine's ability to access and interpret the context in which these rules must be evaluated. In other words, the complexity of the order and services, and the underlying commercial customer hierarchies, are beyond that of most catalogs to implement. The Catalog Designer is built specifically for such an environment and has taken these complexities into consideration when designing the overall catalog and its rules engine.

**Support for external catalogs and systems**

Catalog data is spread throughout most organizations, and budgets and legacy technology may preclude the wholesale migration of this data into a central catalog. Instead of synchronizing this data, ConceptWave offers the ability to supplement its own information with that contained in external catalogs. This data may be accessed in real-time through web services or JDBC access, or through the information model where data is periodically pushed into ConceptWave in ‘native’ format.

### Catalog Designer overview

This section also provides some essential reading, which will help you understand the nature and structure of the Catalog Manager application. It is recommended that you take a few moments to read and consider this content before proceeding with the exercises proper which follow.

**Catalog elements**

Conceptually, the Catalog can be presented in an entity-relationship model in which the entities represent the Catalog objects and relationships represent the Catalog associations. The entities and relationships of the Catalog are called ‘**elements**’.

Catalog elements are *versioned* or *non-versioned*. Non-versioned elements exist in one instance only, while the versioned elements may have many instances.

All elements have a *life span*, with start and end dates. The elements in a *production* state have a *mandatory* start date, but are not required to have an end date, meaning that their life span is not limited in the future.

**Items, properties and attributes**

With the exception of **Projects**, the Catalog Designer application maintains the Catalog objects, but it also maintains the [**Info Tables**](http://InfoTable.html/) object, which contains the content of tabular data structures defined by the **Info Models**.

The Catalog **items** (or just ‘items’) occupy central place among the Catalog objects. They usually account for 90% of the Catalog elements, and represent the products and services that a company offers. The other Catalog objects serve to define aspects of the items' structure, price and business rules.

Catalog objects **properties** consist of data that is defined, entered and maintained by the Catalog maintenance application or the API on the Catalog Server. The properties in Catalog client applications are ‘immutable’ data. **Attributes** consist of data that is also defined by the Catalog maintenance application or the API on the Catalog Server, but this is entered and maintained by the Catalog client applications. The values of the object attributes stored in the **shopping basket**and **service inventory** are used in the Catalog rules and Catalog client applications.

### Navigate the GUI

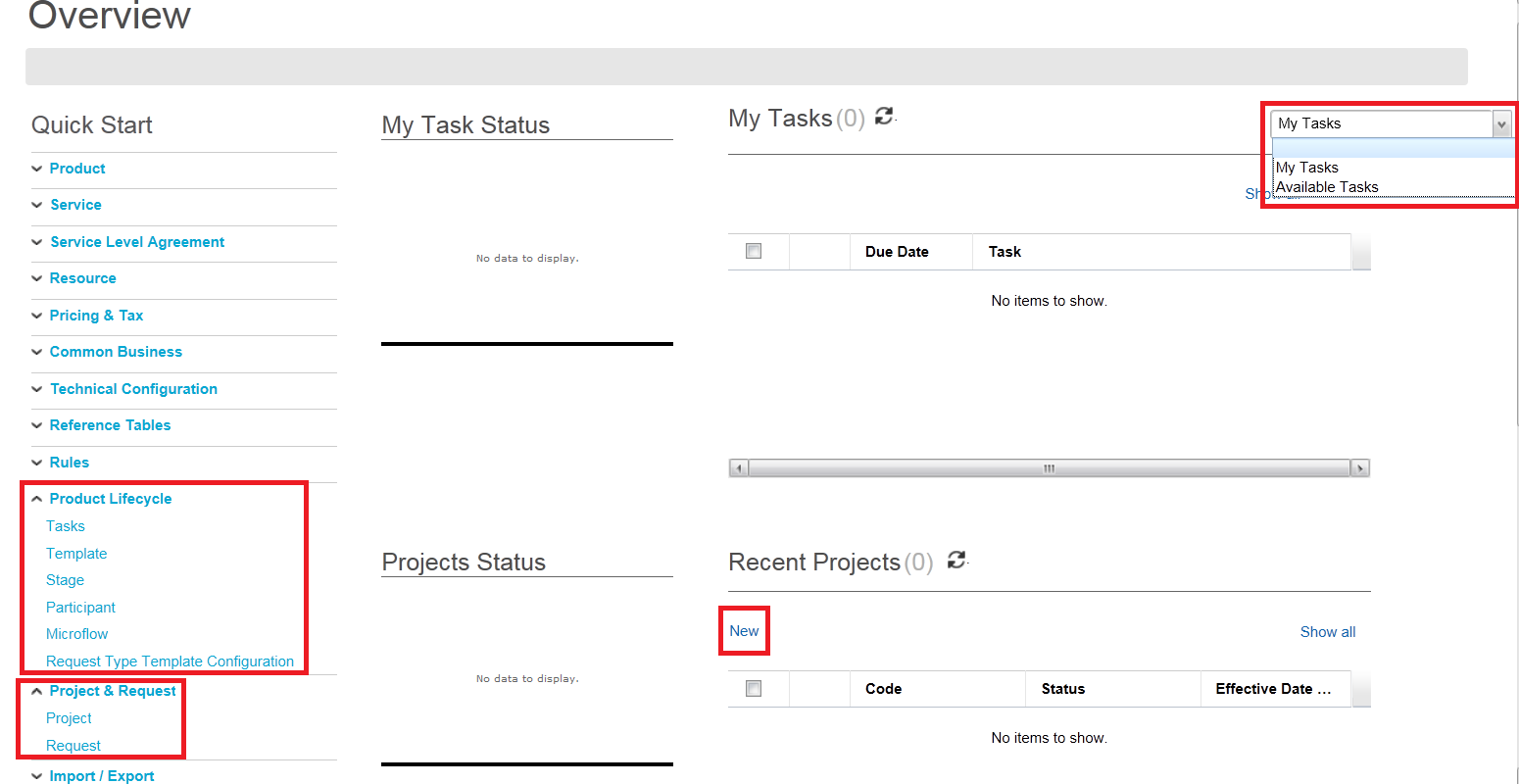
In this section the instructor will take you on a brief tour of the Catalog Designer GUI, providing a summary of the main functions, and a quick review of the toolbar and menu options. At the end of this activity you will then have a better understanding of what is in the GUI, and it will be easier manage the exercises which follow.

The Catalog Designer GUI provides a user-friendly environment for efficiently managing all the elements discussed in the previous section. It allows you to define, control and maintain all the objects, as well as all the properties and attributes associated with them.

There now follows a summary of the different menus and contained options which define the required functions, and which provide the tools needed to manage those functions.

**Overview**

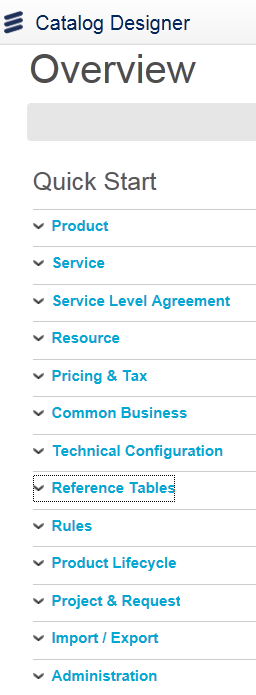
The **Overview** hyperlink returns you to the start screen. Selecting **Project and Request** allows you to see a list of all **Projects** and their status. **Tasks** and **Available Tasks** can be viewed by setting menu value in upper right corner of screen (see screenshot below). The Projectslistwill include those owned by the user, as well as those that don’t have any owner assigned.



This screen also presents an option under **Recent Projects** called ‘**[New](http://PLM/PLMUserProcess.html)**’ (see screenshot above). This provides the ability to use the **Product Lifecycle Manager (PLM)** feature to enable change requests for Catalog items to be processed into the live production environment.

**Overview Quick Start menu**

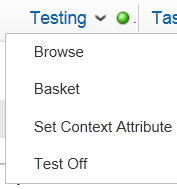
The **Quick Start** menu provides the main user tools for element management:



* **Product**
  + Product Specification
  + Offering Specification
  + Characteristic Specification
  + Items – details of product component
* **Service**
  + Options to define Service specifications
* **SLA**
  + Definition of Service Level Agreements
* **Resource**
  + Resource Specifications
* **Pricing & Tax**
  + Definition of Tax Information and Charge Types (prices & discounts)
* **Common Business**
  + Definition of Recurring Specifications, Algorithms and Business Interactions
* **Technical Configuration**
  + Association Definitions
  + Attribute Types - Definition of the data types used as attributes of different objects
* **Reference Tables**
  + Code Tables
  + Info Models - Definition of tabular data structures
* **Rules**
  + Characteristic rules – object rules
  + Context rules - attribute types that hold request specific data at runtime
  + Catalog rules - Catalog business rules
* **Product Lifecycle**
  + Specifications to govern the lifecycle of a product
* **Project & Request**
  + Definition and Viewing of Projects
  + Within a project the following features are provided:
    - Mechanism to control changes to the catalog definition, including releasing changes to a live production environment in a controlled manner
    - Domains - ability to limit data access
* **Import/Export**
  + Importing and exporting of Catalog information
* **Administration**
  + Configuration of Catalog options such as default currency, catalog privileges
  + Event log - A report that displays a list of messages logged by Catalog Administration

**Test Mode menu**

The **Testing** menu (top right corner of screen) allows a runtime view without integrating your metadata into the Designer, and is useful for testing offer or product selection and charging:

* **Test On**
* Initializes all session variables
* **Set Context Attributes**
* Allows you to provide a value for each of the defined context attributes
* **Browse**
* Opens the catalog browser
* **Basket**
* Allows you to test item and pricing details
* **Test Off**
* Clears the session of the context variables and the current order

**Current ID menu(display will show current ID)**

* **Preferences**
* Specifies Rows per page
* **Switch**
* Allows you to switch between any defined applications
* **Logout**
* Allows you to end your Catalog Designer session in an appropriate fashion

**Login User information**

The **Login User** information button gives the name of the current logged-in user:



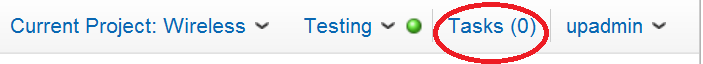
**Projects information**

The **Projects** information button gives the name of the current open project, if one is open:



**Tasks**

The **Tasks** button gives the number of tasks on the current user’s worklist:



**Item types**

The items listed in the **Item** tree node can be identified graphically by distinguishing icons. These icons are stored in the following metadata resources folder:

*<cwproject>\templates\catalog.jar\resources\images\tree\_icons\*

Additional icons can be added to this folder. The script that associates these icons to the item type can be overwritten and modified, and is located in the following library file**:**

***cwt\_pc*** *>* ***Business Processes*** *>* ***Scripts*** *>* ***getItemTreeNodeIcon***

### Find information in the GUI

There now follows a few ‘search and find’ questions, inviting you to look for specific information in the Catalog Designer GUI. Having learnt some things about the functions and navigational options in the previous exercise, you should now be able to quickly and efficiently find specific objects and related information. This activity should also provide some GUI familiarity before embarking on building any items in the new project.

Using the **Catalog Designer** GUI menu options, search for and find the information requested in each question below, and write down your answers in the spaces provided.

1. How many **Attribute Types** are there currently in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is the full **Name** of the ‘stepped’ rule?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is the name of the only **Charge Type** in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Are there any **Contexts** currently in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. How many **Association Types** are there currently in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Describe the structure of any **Catalog Hierarchies** you can find.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is the name of the only **Project** in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. How many **Items** (at the ‘base’ level) are listed in Catalog Designer?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Which **Charge** is associated with the **Silver** item?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Which **Group** is associated with the **Bronze** item?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. How many **Code Tables** are there currently in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Review the **Event Log** and say how many log entries start with the number ‘6’.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. How many **Worklist** tasks are currently in the system?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What does the **Select Application** menu option do (specifically)?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Is it possible to change your own login password?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Open the online **Help** facility and give the names of the three main chapters you find there.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Are there any tasks in the default **WorkList**?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Exercise 3: Scenario overview and new project

In order to be able to map the details we want into Catalog Designer, we must first create and open a new *project*. We are then well-positioned to start building items that can be associated with that project. These items can be structured in a hierarchy, where top-level items will inherit lower-level ‘base’ items.

The sections that follow give a brief overview of the scenario to be built, and then take you through the process of creating and opening a new project.

### Scenario overview for Catalog Manager exercises

The scenario we will build in these exercises is a simple example of a typical project, which includes consideration of required attributes and a demonstration of item inheritance.

Consider the customer request. The customer would like a new **HSI** (High Speed Internet) service, and we, the company, are able to provide different levels of service for this type of offering: **Gold**, **Silver** or **Bronze**. Each service level is defined by a choice of bandwidths and associated charges. The Catalog Designer system therefore needs all the items and attributes for these services mapped against a specific **project**, and with a structure that clearly defines each service level. A user can then simply select the required service and relate it to the customer making the service request.

The following diagram provides a service structure overview, the components of which will form the basis of which items we create in the Catalog Designer GUI. The different levels provide an idea of how *association* and *inheritance* will be worked, leading to the pricing requirements at the bottom of the tree.



Your instructor will discuss the logic of this scenario with you in class.

*Note: Throughout the exercises which follow, all new items and relations are created in a state of ‘****Definition****’ (development mode). Typically, at the end of the development, the project is then ‘activated’, which sets the status of all that was created to ‘****Active****’. (The system allows you to do this regardless of* ***Start Date*** *and* ***End Date*** *information logged against the different items).The mechanics of the ‘activation’ operation are not covered on this course.*

Before the structure demonstrated above can be configured in Catalog Designer, a new project must first be created and opened.

### Create new project and add detail

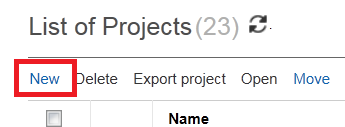
In Catalog Designer you cannot create or maintain any catalog object unless there is an open project. Multiple projects can be created, but you can only have one project open at any one time.

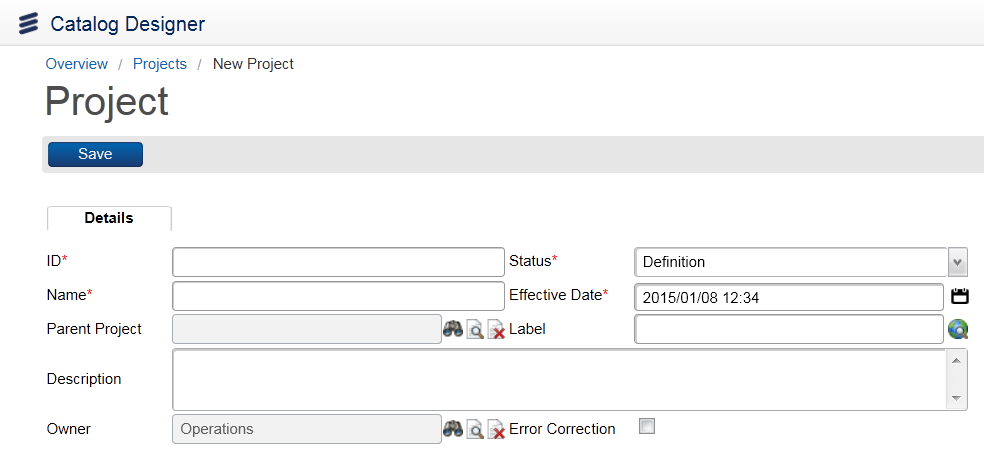
1. If required, log back in to **Catalog Designer** GUI as **upadmin** (as you did in Exercise 1).
2. Select **Project & Request/Project** via Quick Start Menu*.*:



In the resulting list of projects you may see project(s) there already. You can ignore these for the purposes of this training.

1. Click **New** in the center of the screen. The **Project Detail** panel then appears:





In the next few exercise steps you will put in attribute details that will define the new project and impact on its future behavior. Some of these details are mandatory (indicated in the GUI by an asterisk), whilst others are optional.

1. Enter an **ID** of ‘**ProjHSI**’. This is used for internal business purposes in the organization.

*Note: All ‘Code’ values associated with items throughout the exercise document should not contain spaces, so that code names are compatible with other systems and GUIs associated with the Catalog Manager application.*

1. Enter a **Status** of ‘**Definition**’ (is the default). The lifecycle of statuses starts with **Definition**, and can go all the way through to **Active**:



Whilst the project has a status of **Definition**, it can be modified. However once the project is transitioned into **Active** status, the project cannot be modified thereafter. Any objects in the catalog with a status other than **Active** will not be visible to the outside world.

*Note: This behavior is not just applicable to projects, but also to any other catalog objects, which have similar statuses applied. When a project’s status is changed, all objects that have been created or modified within that project also have their statuses changed automatically.*

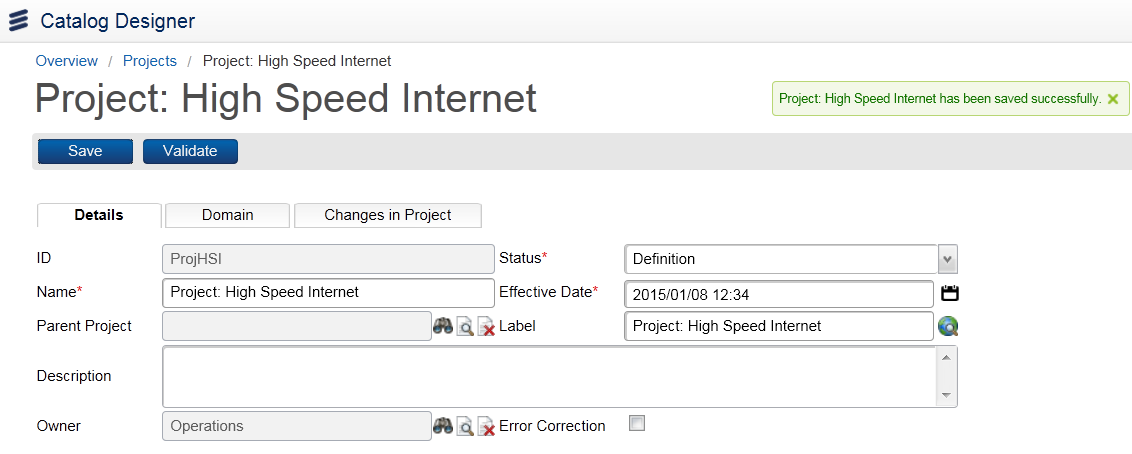
1. Enter a **Name** of ‘**Project: High Speed Internet**’.
2. Enter an **Effective Date**. Today’s date is automatically populated in this field, but you could select any future date here. This date refers to when the changes configured in the project are intended to take effect. Leave today’s date in the field, otherwise you won’t be able to see the effect of your changes during this training period!
3. Leave the **Parent Project** field blank. It is possible to have your projects configured in a hierarchy, but this is not a requirement in this exercise.
4. Enter a **Label** value of ‘**Project: High Speed Internet**’. This is the same as the **Name** value on this occasion. Use *Copy/Paste* to save time.

*Note: It is possible to provide translations to this Label text by clicking on the globe icon next to the field, selecting the required language, and entering the translation text. This is not required today.*

1. Leave the **Description** field empty. This is an optional, free text field.
2. Leave the default value of ‘**Operations**’ in the **Owner** field (the top level of our **organizational chart** defined earlier).

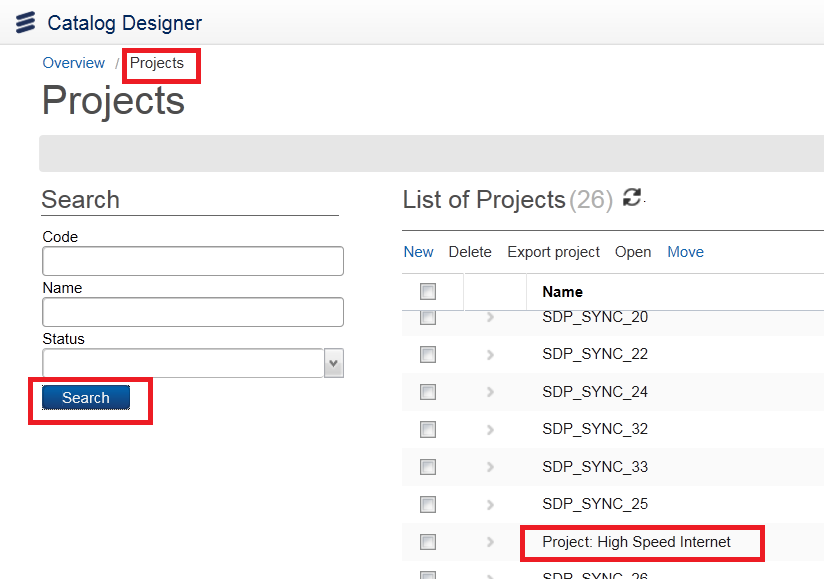
It is possible to associate the project with a location (position) in the organizational chart. This means that only those users at a selected position in the chart (and their managers) will be able to maintain this project.

1. When you have entered all these values and saved your new project, your **Project Detail** window should look like the screenshot below.



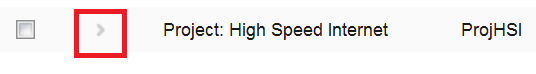
*Note: The information associated with the new project will not come into effect until the* ***Effective Date*** *has been reached, and the* ***Status*** *has been set to* ***Active****.*

1. Click on Projects at top left and then Search. Your new project now appears in the list. You may have to scroll down the list of projects to view your new project.

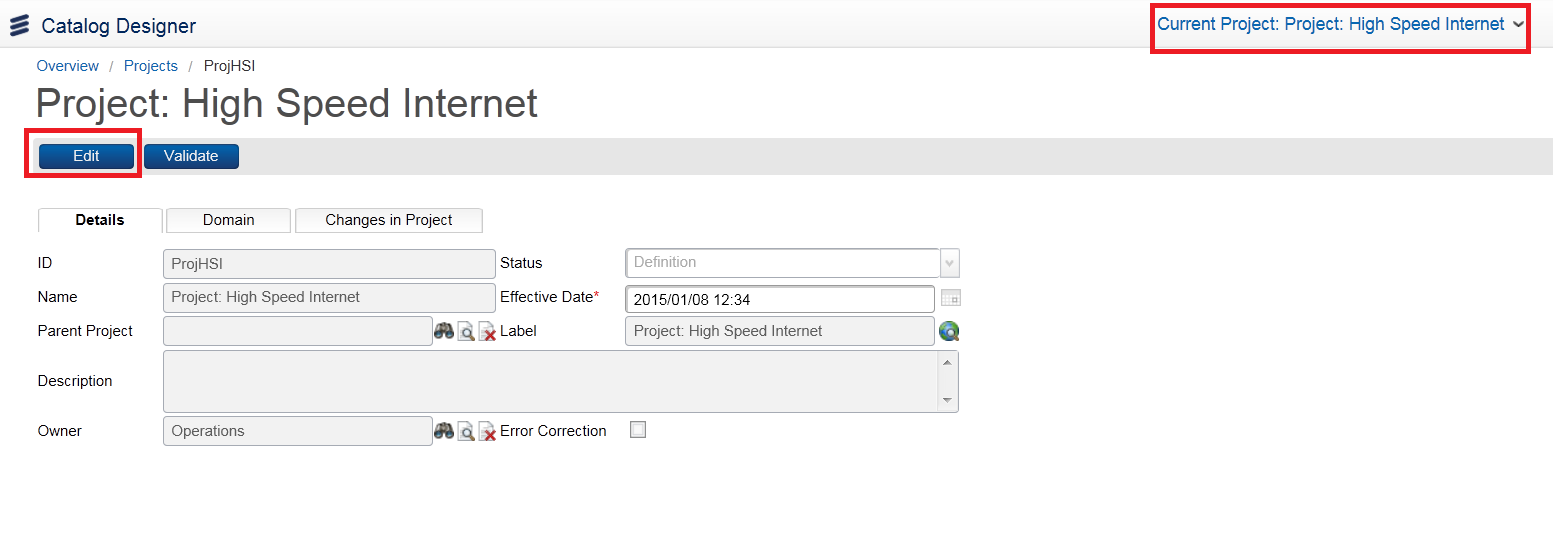


### Open project

1. Select the ‘**Project: High Speed Internet**’ project just created and open it by clicking on **>** next to the project.



1. Review and validate the project detail, and note that this project is displayed at the top as ‘**Current Project**’. You can only create and modify associated items if the project is ‘open’ in this way.



*Note: In the* ***Domain*** *tab users are able to add* ***organizational chart*** *positions to restrict access to selected users for a particular record, such as a* ***project*** *in this instance. If left blank, all users have access to this record.*

Module 4: Code Tables and Attribute Types

In this module we take a look at **code tables** and **attribute types**, and how they are created and managed in the **Catalog Designer** GUI. As a result of creating these, users will be able, later, to work with **items** and other related objects in the Catalog Manager processes defined in this document.

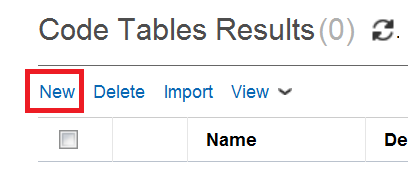
## Exercise 4: Code tables and attribute types

In this exercise we create a new **code table** called ‘**Bandwidths**’. We then create six different bandwidth codes inside the **Bandwidths** table. We learn how to **export** and **view** code tables, and finally we create two new **attribute types**, the basis for user menu choices in the live system.

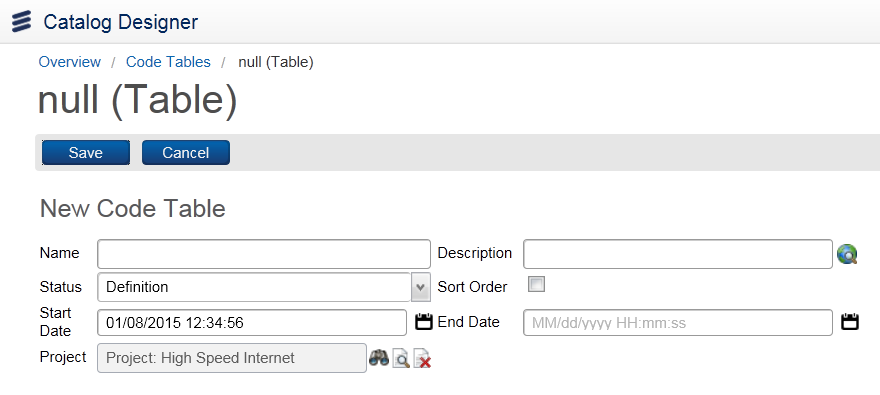
### Create code table

Creating a **code table** is a user’s way of enumerating a number of values that can be used later on to define **attribute types**.

1. If required, log back in to the **Catalog Designer** GUI as **upadmin**.
2. Go to *Reference Tables > Code Table* from the Quick Start Menu.
3. Click on New in the central panel. This takes you to the **General** screen, where we can add a new code table.

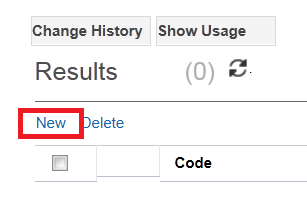


1. Enter a **Name** of ‘**Bandwidths**’, and leave all other options with their default values (as shown in the screenshot below). This new table will contain the different bandwidths we need for our Internet service connections.

`

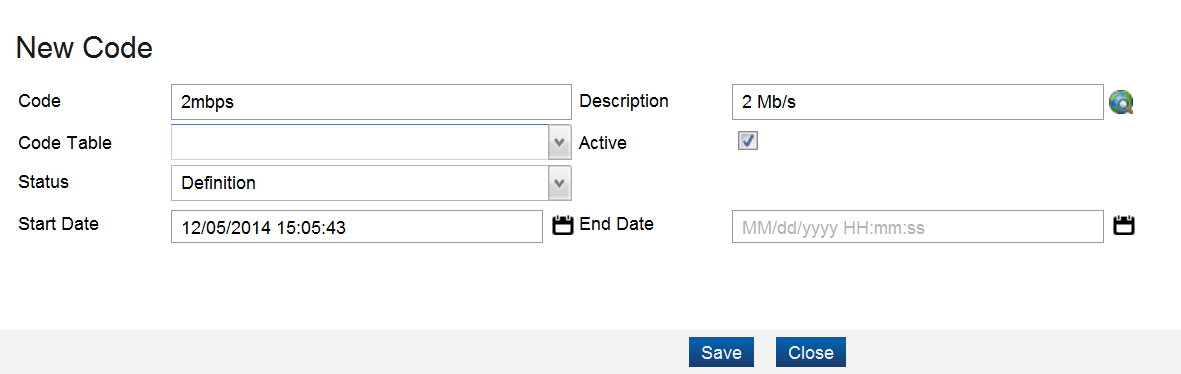
1. Click **Save** at the top of the screen (see screenshot above).

We now need to add the different **codes** to be included in the code table, using the **New** hyperlink which is now available in **Results** section.

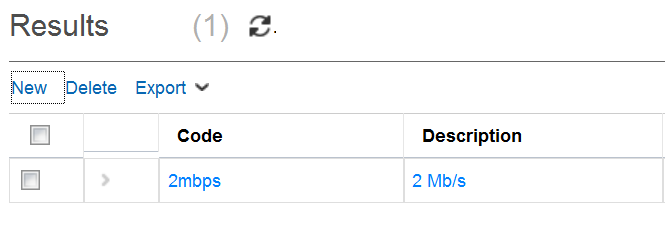


*Note: You can create a table by either adding individual codes one by one, or by adding a group of codes that come from a different table, - or both. It is also possible to add other tables to a code table, using the* ***Code Table*** *menu.*

1. In the **Results** panel click **New**, then enter the value ‘**2mbps**’ in the **Code** field.
2. Enter a **Description** of **’2 Mb/s**’ (something more readable, since this is what the user actually sees).
3. Ensure that the **Active** check box is ticked. If you leave this box unchecked, the code will be in the table, but it will not show in the dropdown list when the user tries to add a value in a field that is based on this code table.

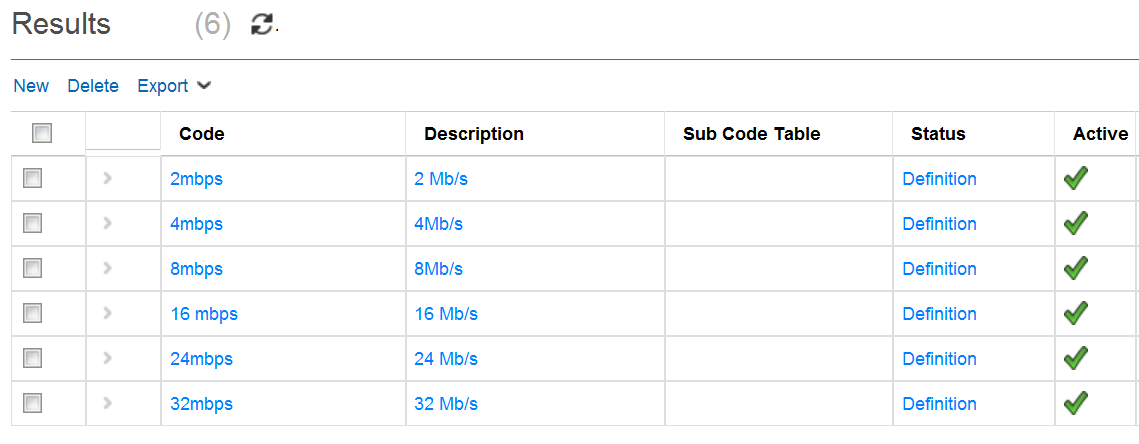


1. Click **Save** and **Close** and you will see the new value in the list of codes in the **Codes** box.



1. Now add the other bandwidth codes we need for our Internet services, as presented in Exercise 3. When done, you should have the following six codes configured:

* 2mbps, 4mbps, 8mbps, 16mbps, 24mbps, 32mbps,



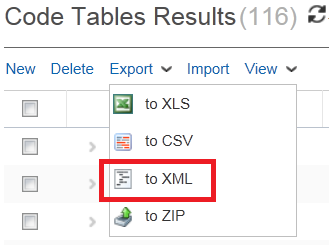
1. Click **Save** at the top of the screen.

*Note: If you go back to the Code Tables level and Search, you can see many existing code tables in the central panel. These come with the catalog. They are used for things like currency codes, Canadian provinces, US states, etc.*

### Export and view code tables

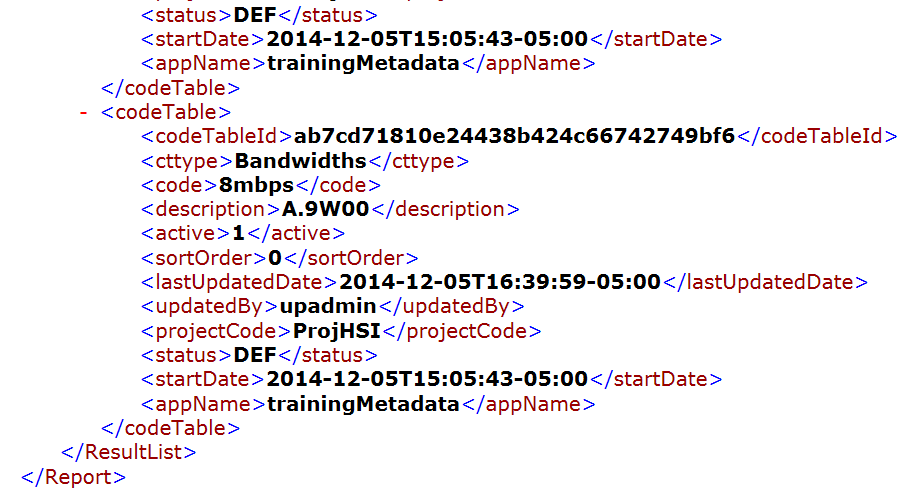
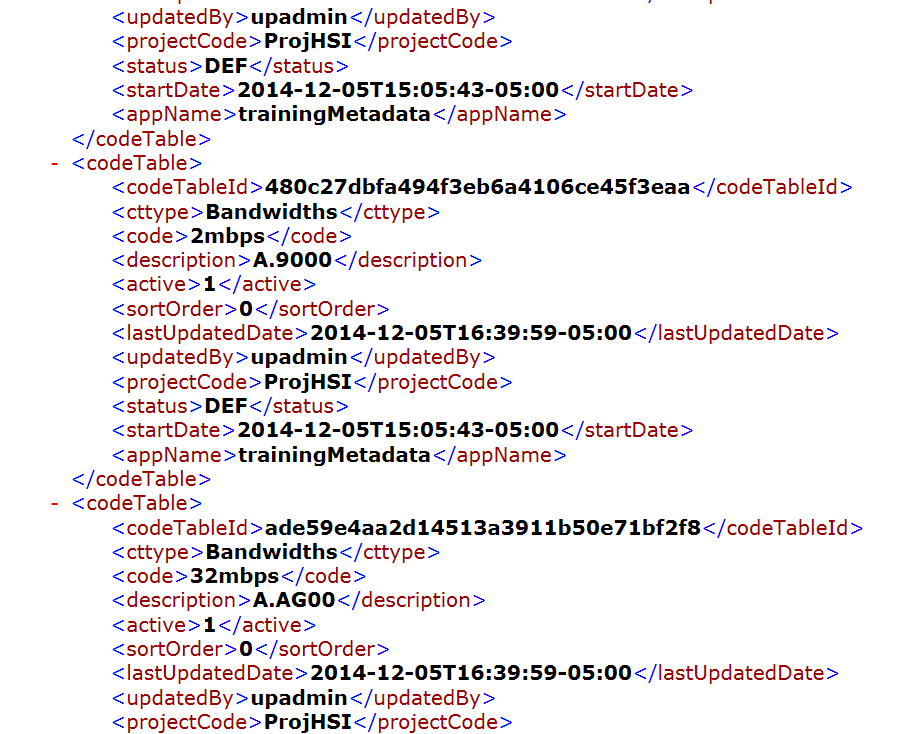
Code tables can be exported to an xml *file* when a single code table has been selected or to a *folder* of different xml files when multiple Code Tables have been selected.

1. If required, click back onto the **Bandwidths** code table entry in the left panel, then click on the **Export** hyperlink under **Results** in the center part of the screen and select **Export to XML**:



1. A pop-up window appears containing the XML. Click on File save as to *‘C:\CWOrderCare\Bandwidths.xml’*, then click **Export**.
2. Assuming the export was a success, locate and view the resulting output file, where you can see the **Bandwidths** code table and contained bandwidth values:





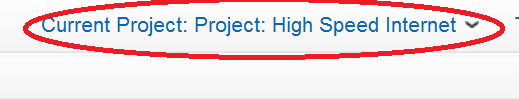
1. Select the ‘**x**’ in the top right-hand corner to close the **export XML** dialog box.
2. Go to Code Tables screen, select **Export to XML** from the **Export** icon menu options.
3. A pop-up window appears as before and you should click File/Save As to save xml to the following target directory: *‘C:\CWOrderCare’.*
4. Locate and view the target folder, where you can see that each exported table has resulted in its own *xml* file.
5. Spend a few minutes reviewing the contents of selected xml files from your export.

*Note: that all code tables are contained in the xml file.*

### Create bandwidth and string attribute types

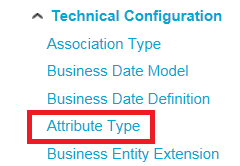
Attribute types are basically just ‘*data*’ types.

1. First make sure you have an open project (**Current project: Project: High Speed Internet**).



*Note: If necessary, go to Project & Request/Project from the Quick Start Menu to select the Project specified above.*

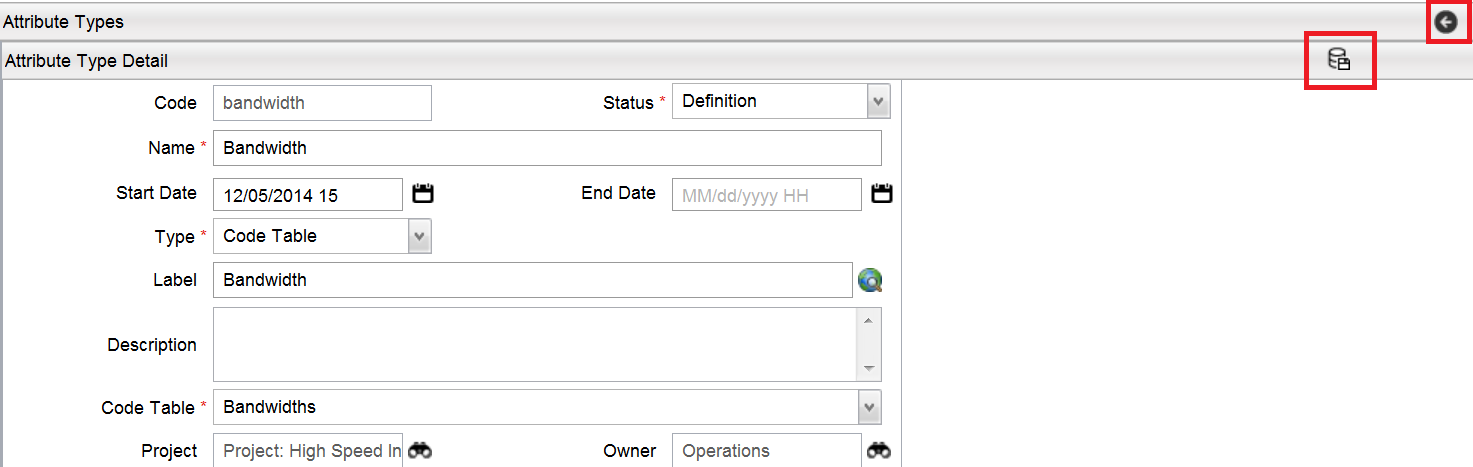
1. Go to *Technical Configuration > Attribute Type from the Quick Start Menu*.



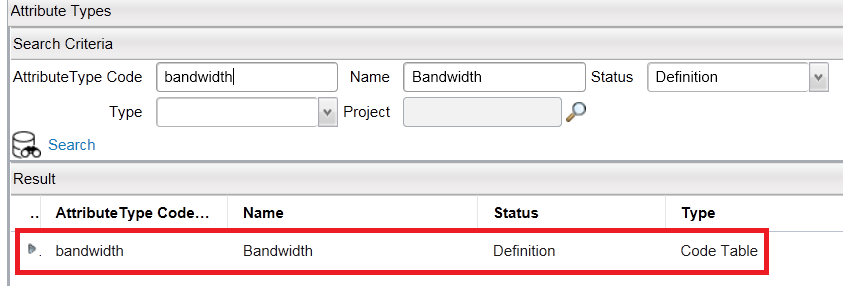
1. Click **Add** in the bottom left-hand corner of the screen, and enter the following values, as shown in the table below:

|  |  |
| --- | --- |
| **Attribute Type Code** | ‘bandwidth’ *(must be a Java name)* |
| **Status** | ‘Definition’ |
| **Name** | ‘Bandwidth’ *(visible to end users, with spaces and special characters allowed)* |
| **Start Date** | [Today] |
| **End Date** | [One year from today]  *The period defined by these two dates represents the effective ‘life’ of the attribute type* |
| **Type** | ‘Code Table’  *Provides a list of code tables in a new* ***Code Table*** *field. Each attribute type created must be based on a value from the* ***Type*** *list* |
| **Length** | N/A (leave as ‘0’) |
| **Label** | ‘Bandwidth’ *(same as* ***Name****)* |
| **Description** | N/A (leave blank) |
| **Code Table** | ‘Bandwidths’ *(created earlier)* |
| **Project** | ‘Project: High Speed Internet’ *(should already be populated)* |
| **Owner** | ‘Operations’  *As before, the* ***Finder*** *can be used to locate and select the required Owner, thus restricting access to this particular catalog object to owners at this level - and their managers. Accept the default entry here* |

*Note: If the Coe Table Bandwidths does not appear in drop down list, you may have to log out and back in and possibly stop and restart framework.*

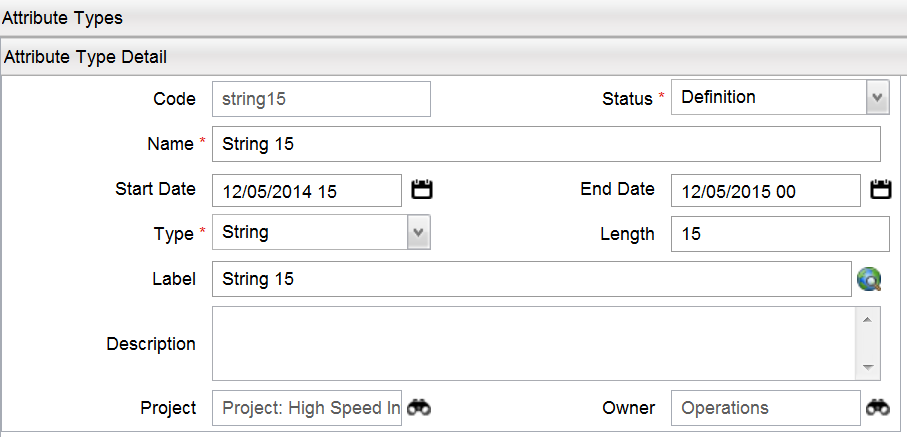


1. Save the new attribute type, using the **Save** icon in the top right-hand corner of the panel (see screenshot above).
2. Click the **Back to results** button (just above the **Save** button, see above) to go back to the attribute finder, where you will now see the new attribute type in the **Result** table, with a status of **Definition**:

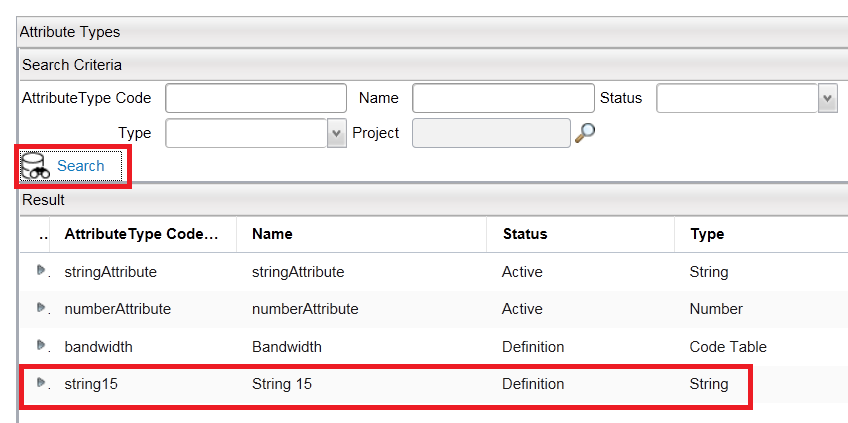


1. Repeat the process defined above to create another new attribute type called ‘**string15**’ (i.e. a string attributes type with a permitted length of 15 characters). Populate fields as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘string15’ |
| **Status** | ‘Definition’ |
| **Name** | ‘String 15’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Type** | ‘String’ |
| **Length** | ‘15’ |
| **Label** | ‘String 15’ |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |



1. Go back to the **Attribute Finder** panel and check that both new attribute types are visible in the **Result** table (You may have to Click on Search first to refresh list):



Later on you will see that item attributes based on attribute types of type **Code Table** will result in a drop-down list at runtime, from which users can select a value.

Module 5: Component Items and Associations

In this Module, which contains Exercises 5 and 6, we focus on **component items**, **bandwidth attributes**, **item relations** and **groups**. At the start of each exercise, summary information is provided giving the main objectives for that exercise. By the end of Exercise 6 you will have created a **base** item, several **component** items and a **product** item, together with required attribute and relational configurations.

## Exercise 5: Component items and bandwidths

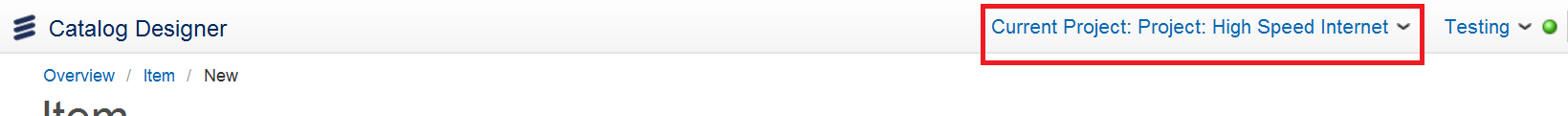
In this exercise we create a **component (‘base’) item** and two **bandwidth attributes**. We then create three additional component items which *inherit* the bandwidth attributes from the base item. We also add **restrictions** to the attribute values, in order to control what the users see in their dropdown menus.

### Create component item and add detail

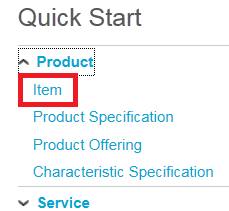
An **item** is a *product*, an *offer*, a *bundle*, a *component* etc. It can either be sold (‘*orderable*’), or is part of something else which can be sold.

We will now create a **component item** called ‘**Internet Access**’, and add appropriate detail.

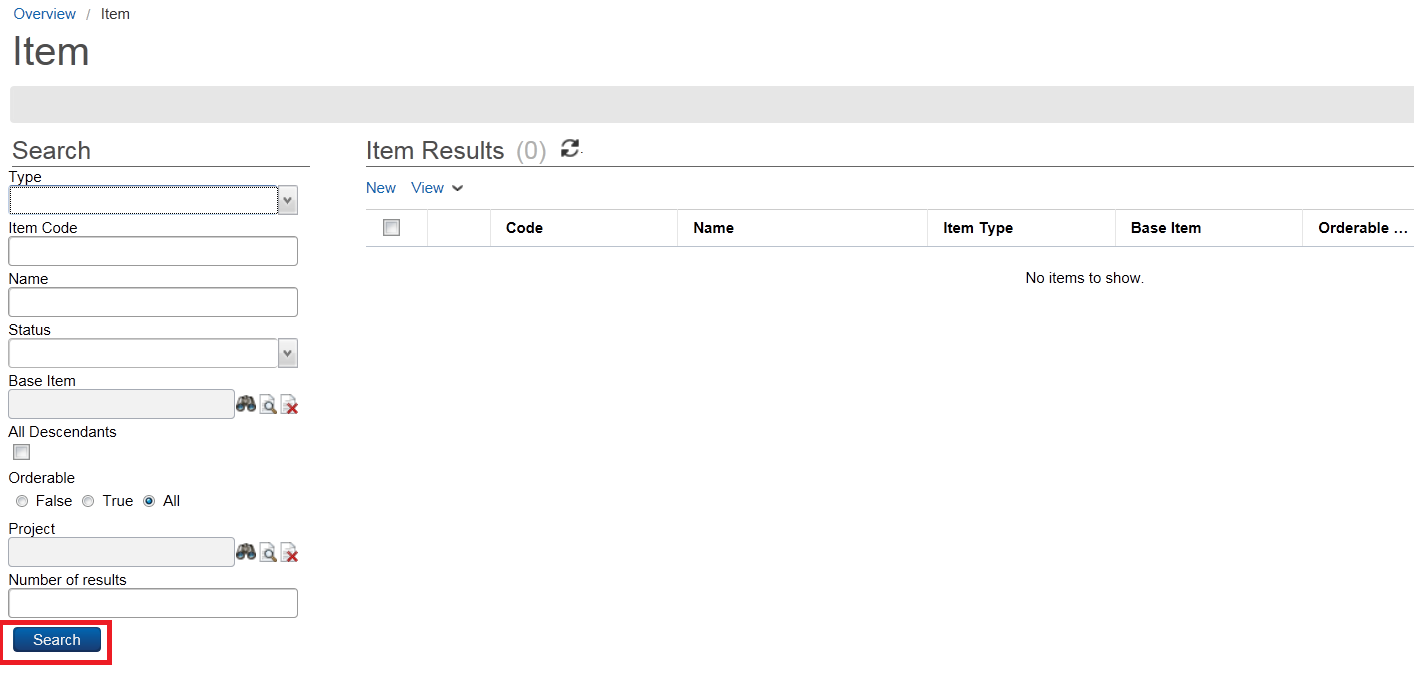
1. Ensure that your project is open, with the name ‘**Project: High Speed Internet**’ visible in the Catalog Designer toolbar area at the top of the screen:



1. Go to *Product > Item from the Quick Start Menu*.

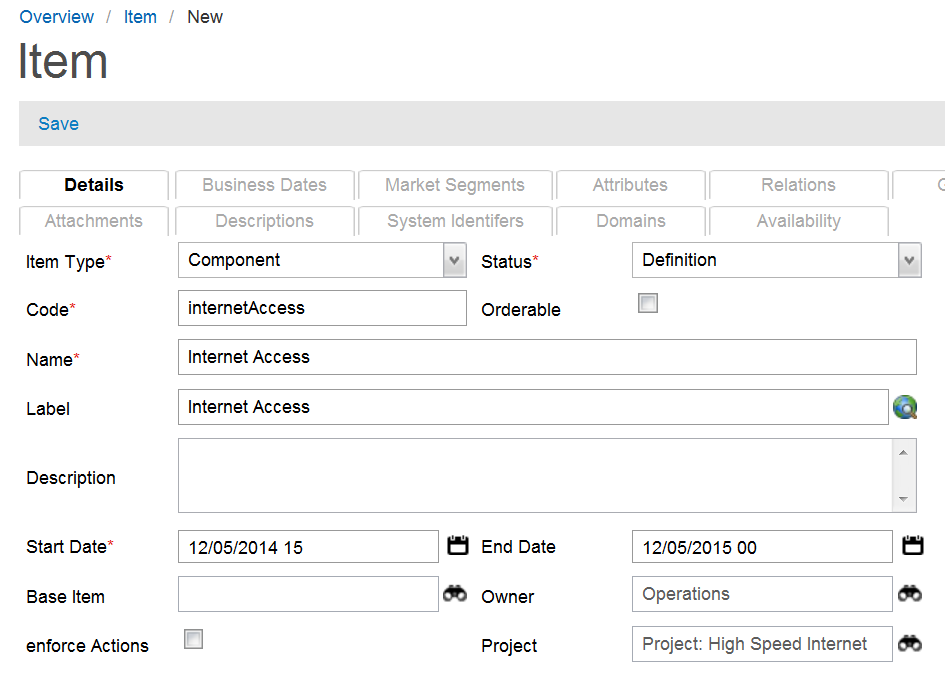


1. Click Search.

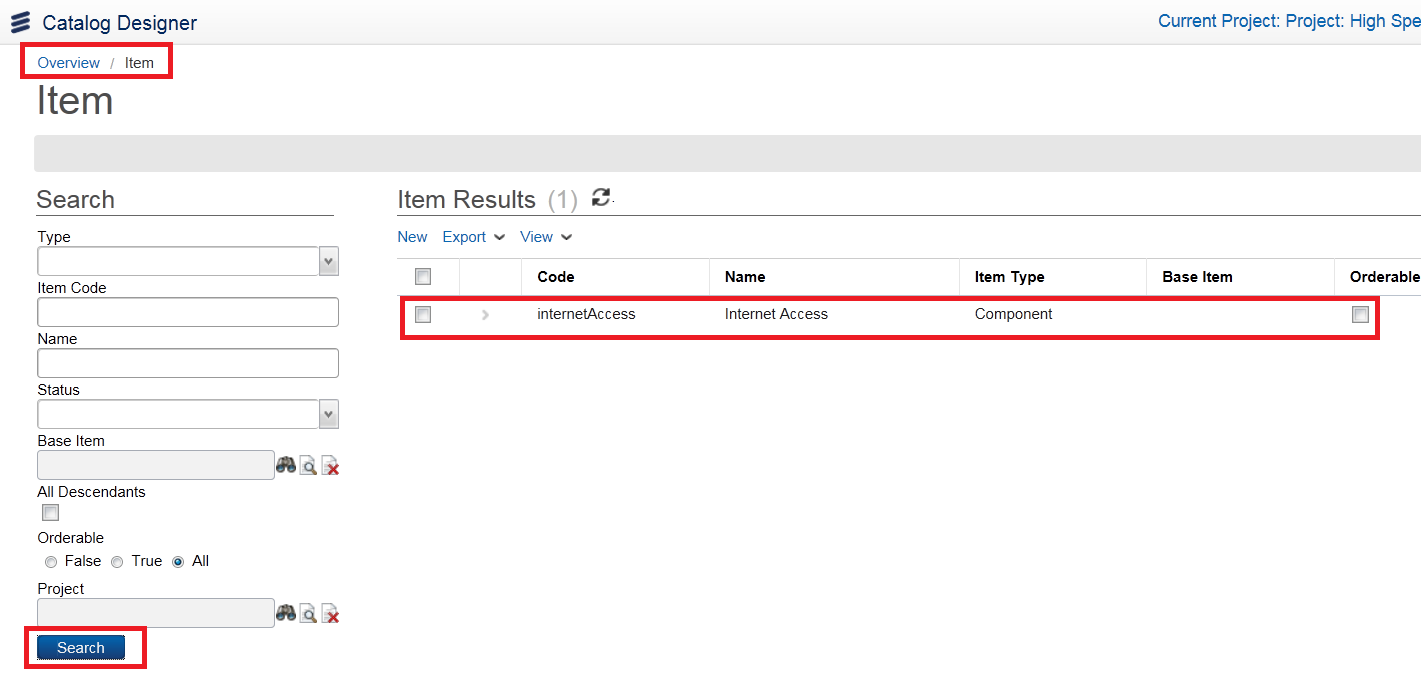


1. Click on New in central panel in order to add a new item in the **Item Detail** panel (see screenshot below).
2. Provide the detail information required for this item as follows:

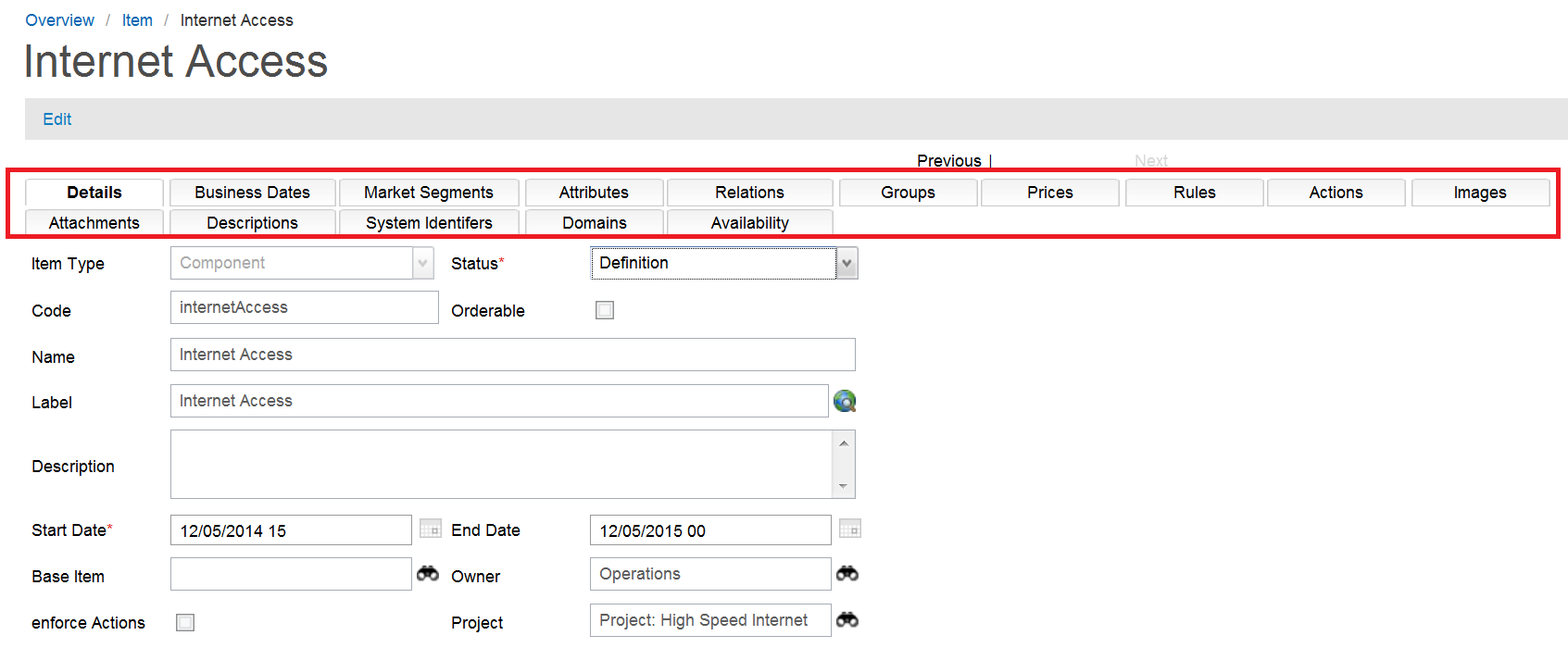
|  |  |
| --- | --- |
| **Item Type** | ‘Component’  *The options in this field can be changed by editing the item type* ***code table****. The default values are* ***Component****,* ***Offer*** *and* ***Product****. Anything else in this list has been added by a user via the* ***Add Code*** *option in the* ***item type*** *code table* |
| **Status** | ‘Definition’ |
| **Code** | ‘internetAccess’ |
| **Orderable** | N/A (leave blank)  *This item will not be ‘orderable’ (i.e. sellable) on its own, but will be a ‘component’ of another item which is orderable. The item will not be returned during catalog browsing* |
| **Name** | ‘Internet Access’ |
| **Label** | ‘Internet Access’ |
| **Description** | N/A (leave blank) |
| **Start Date** | [Today] |
| **End Date** | [One year from today]  *These two dates define the period for which this will item will be available* |
| **Base Item** | N/A (leave blank)  *Shows a previously created item whose attributes will be ‘inherited’ by this new component item* |
| **Owner** | ‘Operations’ |
| **Project** | ‘Project: High Speed Internet’  *(should be automatically populated)* |



1. Save the new item clicking Save at top (see screenshot above).
2. Return to Item screen and click Search to see new item.



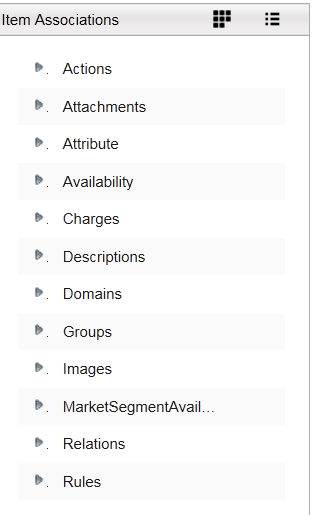
1. Now expand the **Internet Access by clicking on >** next to the item. The tabs represent catalog objects that this item can be associated with:



Note: 2 other sections are created with an Item as shown in the following screenshots.



This is a pictorial representation of the current object. This will become more meaningful later in the course when we establish relationships between objects.

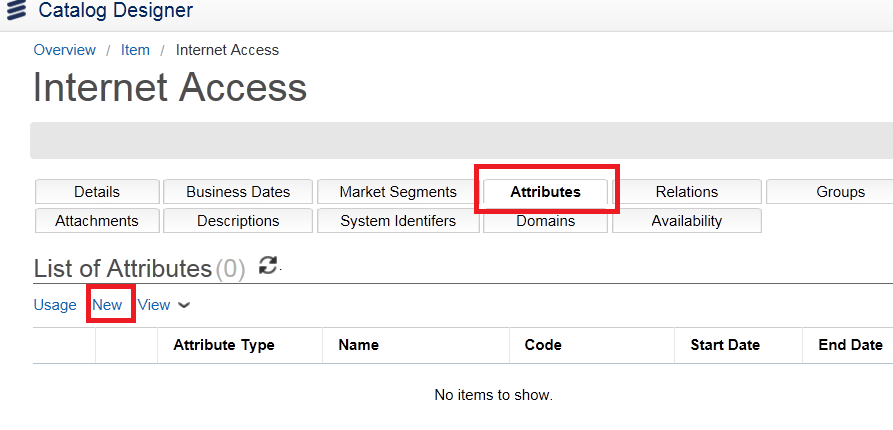


This appears on the right side of the Item screen. We will use it later in the course to use the drag-and-drop capability of the Catalog Designer.

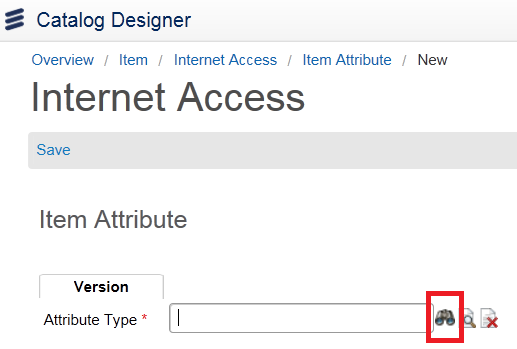
### Create bandwidth attributes, and associate with component item

We will now create two bandwidth attributes called ‘**Upload Bandwidth**’ and ‘**Download Bandwidth**’, associating them with the **Internet Access** component item created in the previous section. These will be based on the **bandwidth** attribute type created earlier in Exercise 4.

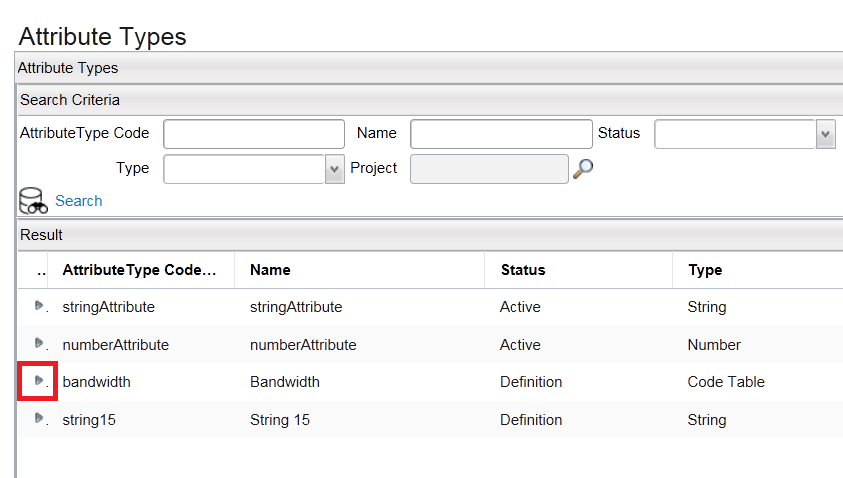
1. Click on the **Attributes** tab, then on New to add a new attribute.



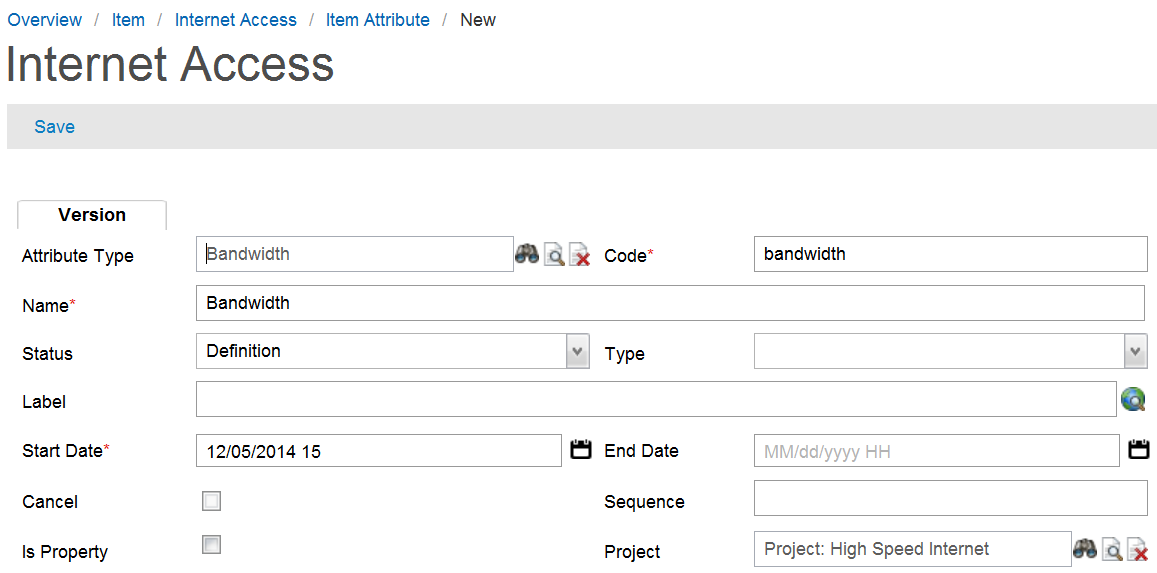
1. In the **Attribute Detail** panel, click on the finder icon next to the **Attribute Type** field:



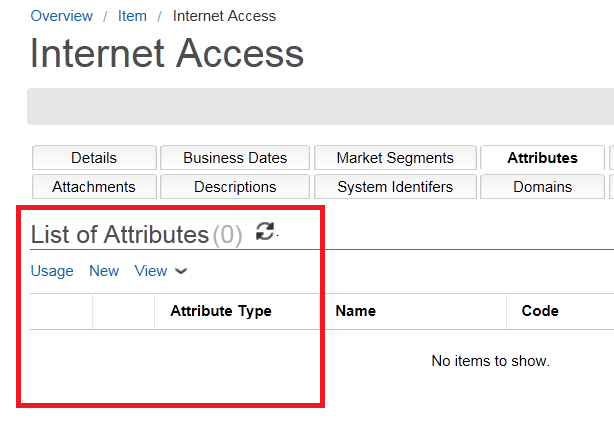
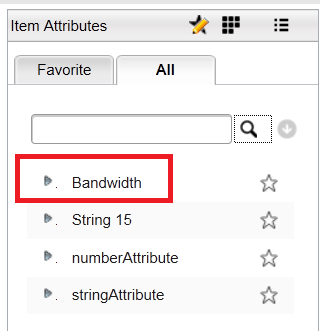
1. A separate window appears. Click Search to get results. Select the **bandwidth** attribute type code by clicking on the arrow at the start of the row. (This will also return the selected object to the **Attribute Type** field on the previous screen).



By clicking on bandwidth item you will get the following:

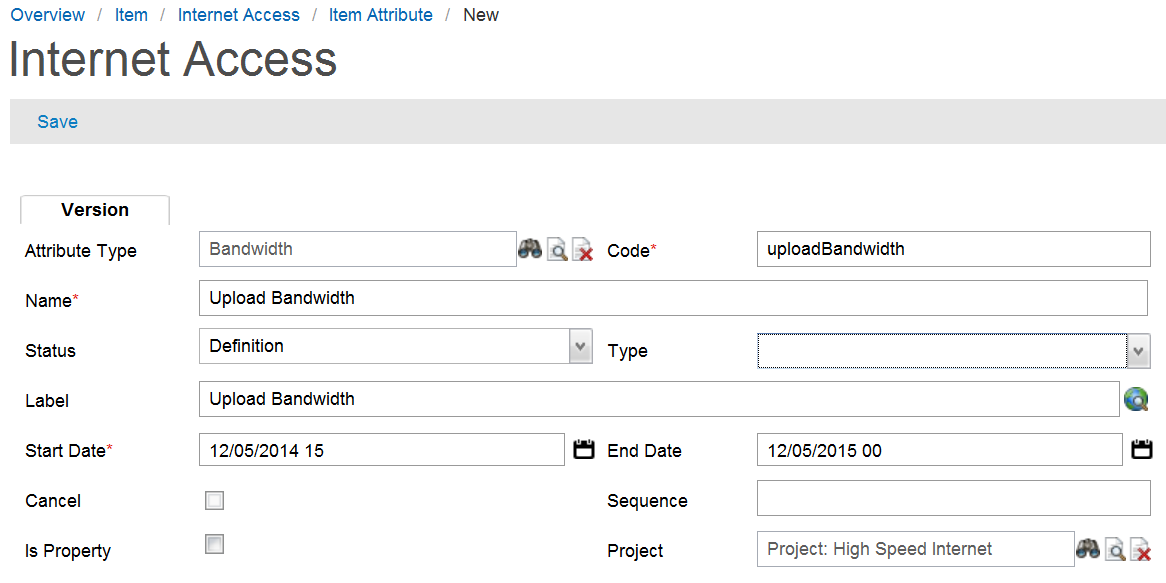


1. An alternative method to obtain the Bandwidth attribute is to drag and drop it from pane at right into List of Attributes:

1. Change the **Code** entry to ‘**uploadBandwidth**’, and the ‘**Name**’ entry to ‘**Upload Bandwidth**’, then enter the following information into the remaining fields:

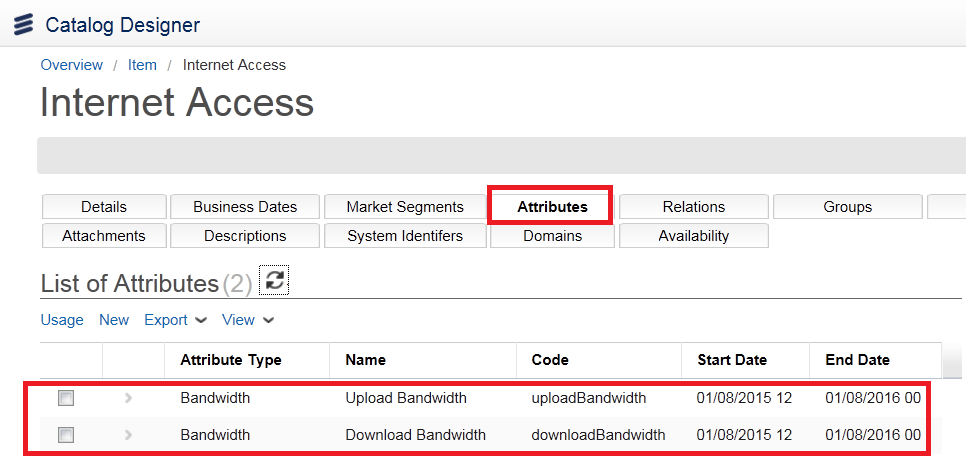
|  |  |
| --- | --- |
| **Status** | ‘Definition’ |
| **Type** | N/A (leave blank)  *This list is currently empty, and will only be populated if we were to create one or more* ***association types*** *of type ‘****item attribute****’* |
| **Label** | ‘Upload Bandwidth’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Cancel** | N/A (leave blank)  *If the Cancel box is checked, and this association is inherited, then the inheritance will be cancelled. If the association is not inherited, then the Cancel field does nothing* |
| **Sequence** | N/A (leave blank)  *Refers to a number value that is used for attribute restriction sorting* |
| **Is Property** | N/A (leave blank)  *If checked, this means that the attribute will be a constant, and the* ***Default Value Rule*** *field is not presented (see below). The attribute will not have different values at runtime. A rule can only be applied when this box is checked* |
| **Project** | ‘Project: High Speed Internet’ |
| **Default Value** | N/A (leave blank)  *Selecting a default value for this attribute is only required if the attribute is to be a ‘constant’* |
| **Default Value Rule** | N/A (leave blank)  *A rule can only be applied when the* ***Is Property*** *box is not checked (i.e. the attribute will be a constant). The* ***rule*** *has the logic which determines the value of the attribute* |



1. **Save** the new attribute by clicking on Save near the top (see screenshot above). Notice the new **Version** entry under List of Attribute Versions on the lower part of the screen .



1. Now create a second attribute called ‘**Download Bandwidth**’, following the same process described above for the ‘**Upload Bandwidth**’ attribute.
2. **Save** your second, new attribute.
3. Check the Internet Access **Attributes** folder once more to confirm that both bandwidth attributes now exist. (You may have to refresh your tree).



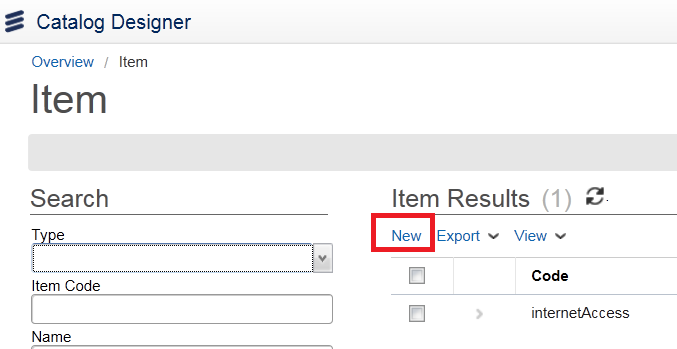
### Create more components items, and add attribute restrictions

We will now create three more component items called ‘**Light Internet Access**’, ‘**Regular Internet Access**’ and ‘**Heavy Internet Access**’, based on the ‘**Internet Access**’ item created earlier, and adding **restrictions** to the *values* associated with the items’ attributes. This means that users will then be restricted in which values they can assign to the attributes from their dropdown lists.

Here is a pictorial overview of the *items* and *restricted attribute values* that we will be creating:



1. As before, click on New in central pane on the Item screen.

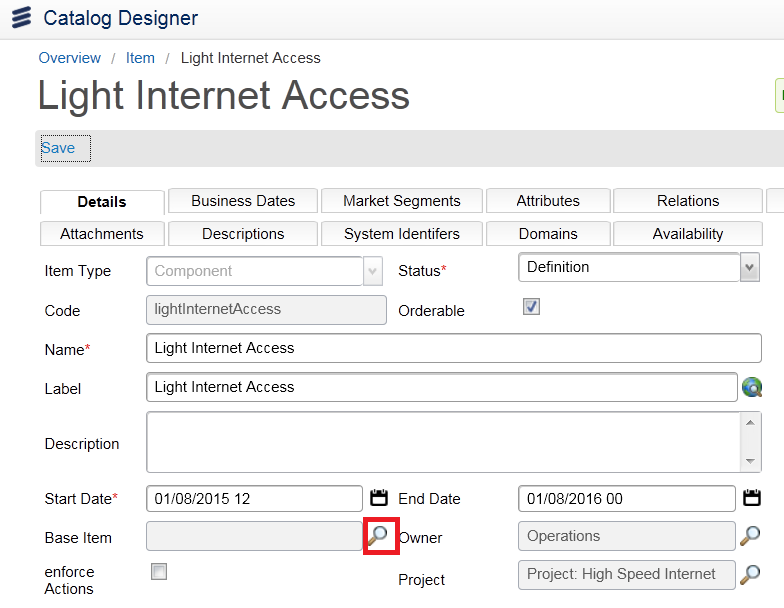


1. In the **Item Detail** panel, fill out fields as follows:

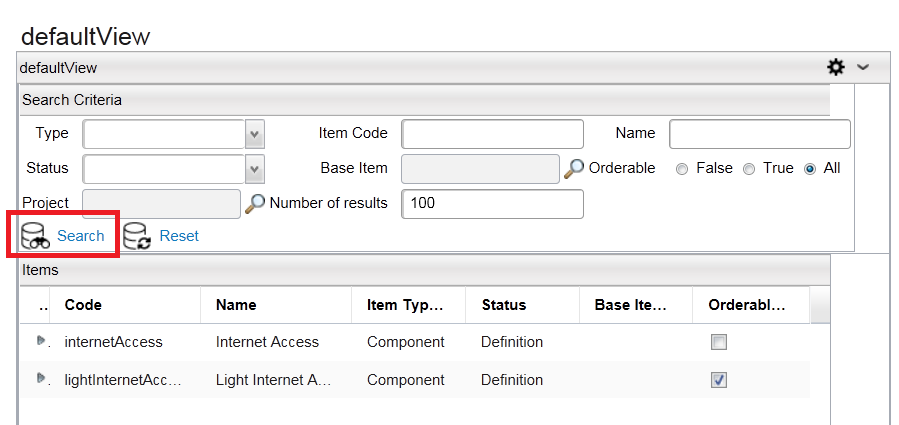
|  |  |
| --- | --- |
| **Item Type** | ‘Component’ |
| **Status** | ‘Definition’ |
| **Code** | ‘lightInternetAccess’ |
| **Orderable** | Yes (check box) |
| **Name** | ‘Light Internet Access’ |
| **Label** | ‘Light Internet Access’ |
| **Description** | N/A (leave blank) |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Owner** | ‘Operations’ |
| **Project** | ‘Project: High Speed Internet’ |

We still have the **Base Item** field to complete. We need to find and select the **Internet Access** item that we created earlier so that our new component item will inherit all its attributes.

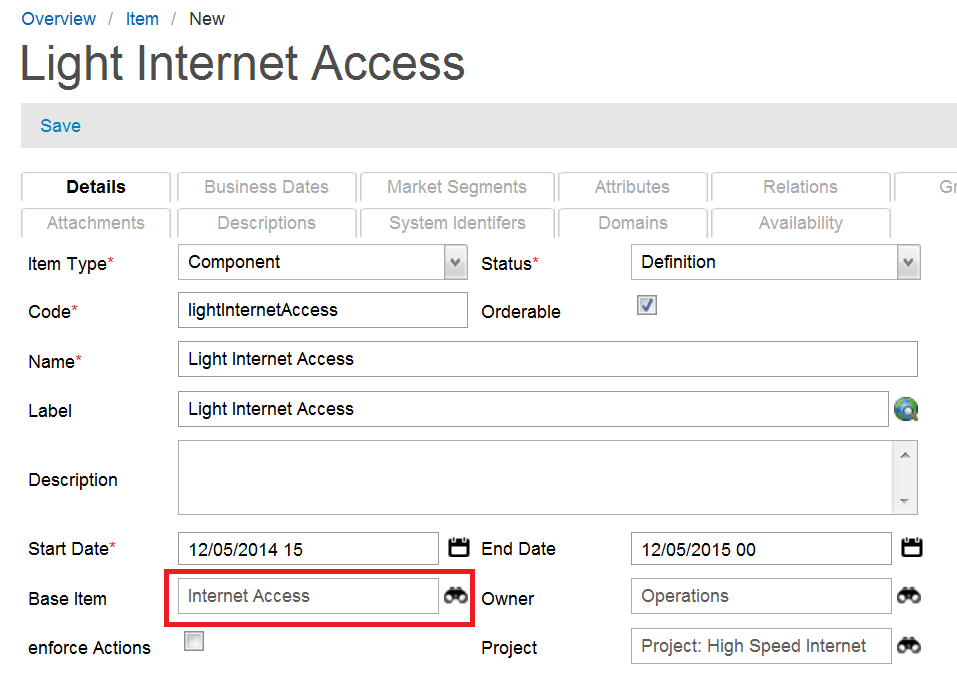
1. Click on the finder button next to the **Base Item** field.



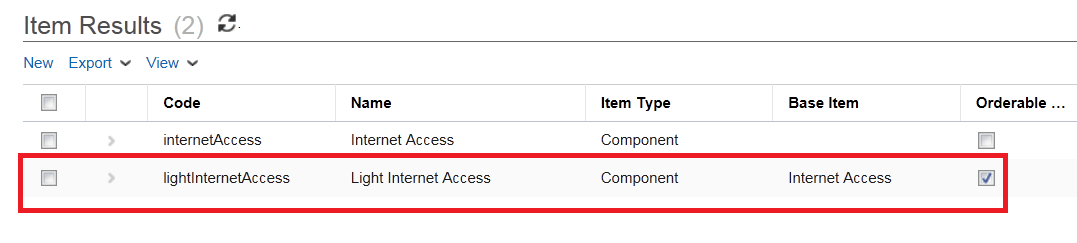
1. Click **Search** in the resulting **Search Criteria** panel, and you will see a list of the different items currently in the system.



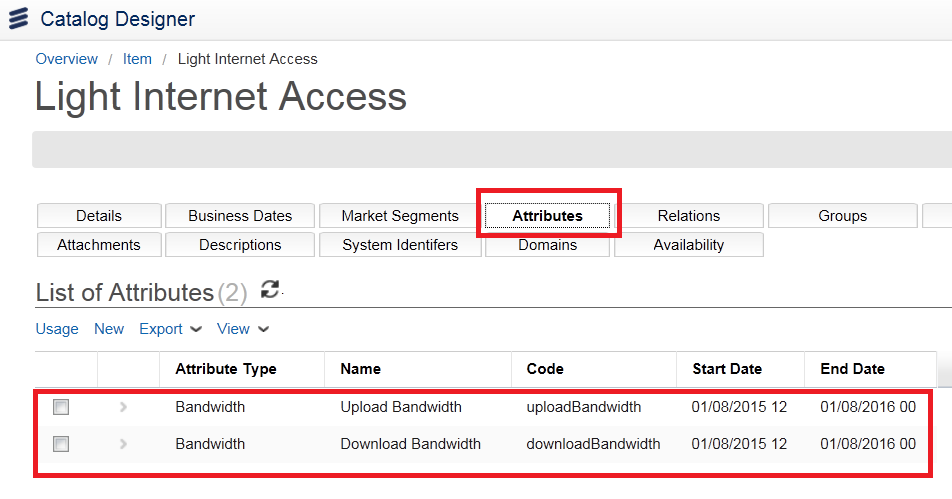
1. Click the arrow to the left of the **Internet Access** row. This not only selects the item, but also enters it into the **Base Item** field:



1. Click the **Save** icon in the **Item Detail** panel title bar, then return to Item screen and Click Search to see the new **Light Internet Access** component item.

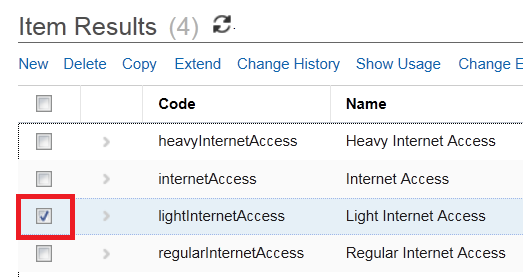


1. Click on arrow next to **Light Internet Access** to see item details.
2. Expand the items under **Light Internet Access by opening the Attributes tab** and you should see the **Upload Bandwidth** and **Download Bandwidth** attributes which are *inherited* from the base item **Internet Access**:

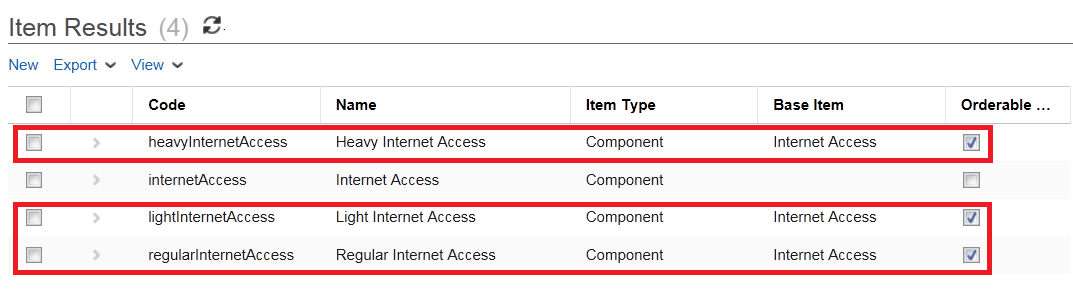


1. Now repeat the process described above two more times to create the **Regular Internet Access** and **Heavy Internet Access** component items, both based on the **Internet Access** item.

Note: A shortcut to do this is to click the checkbox next to lightInternetAccess and select **Copy**



1. Expand the tree in the **Item** panel, and review the new hierarchy you have created. Check that both new items have successfully inherited the bandwidth attribute details from **Internet Access** (as shown in the screenshot above).



We will now visit each of these new component items in turn, and *restrict* the values which users will be able to assign to the **Upload Bandwidth** and **Download Bandwidth** attributes. In other words, we will be defining which bandwidth options users will be presented with in the dropdown lists when applying bandwidth values to the **Upload Bandwidth** and **Download Bandwidth** attributes associated with a particular level of services.

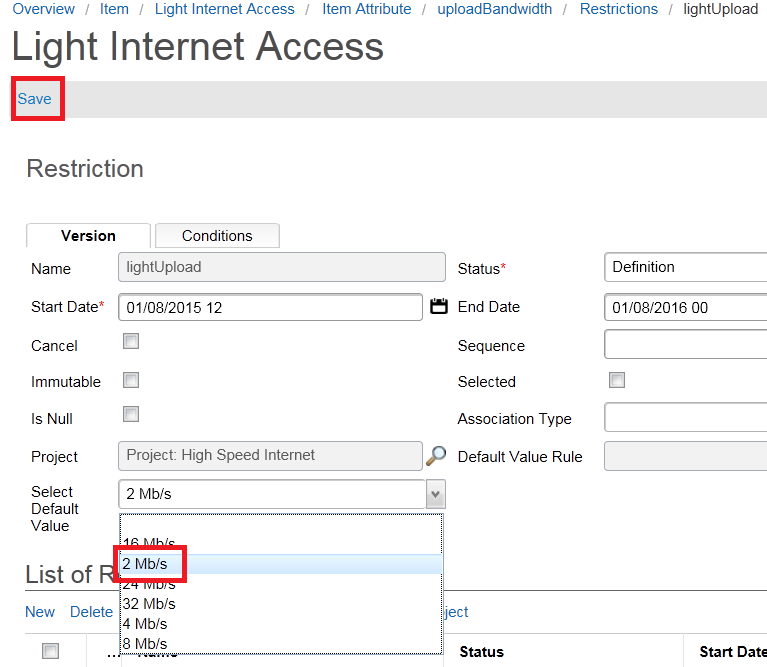
1. Select the **Upload Bandwidth** attribute associated with the **Light Internet Access** item. Select the **Restrictions** tab in the **Attribute Detail** panel, and then click New.



1. Fill in the fields in the resulting **Detail** window as follows:

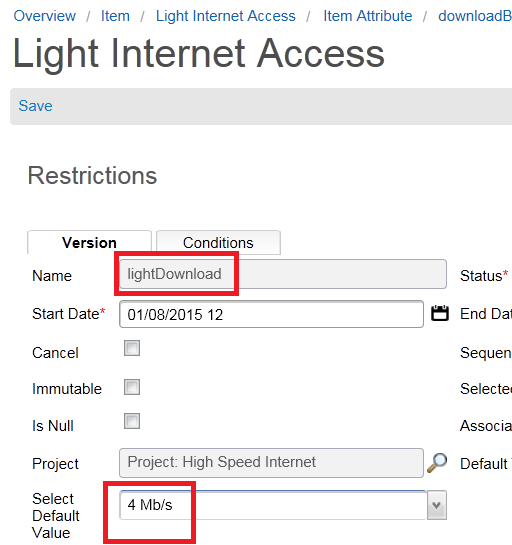
|  |  |
| --- | --- |
| **Name** | ‘lightUpload’ |
| **Status** | ‘Definition’ |
| **Start Date** | [Today’s date] |
| **End Date** | [One year from today] |
| **Cancel** | N/A (leave blank) |
| **Sequence** | N/A (leave blank) |
| **Immutable** | N/A (leave blank)  *Unchecked means that the item attribute can be modified at runtime* |
| **isNull** | N/A (leave blank)  *If left unchecked, a ‘null’ value is not allowed. If checked, it is allowed* |
| **Association Type** | N/A (leave blank)  *Similar to the* ***association type*** *field used for* ***items*** *and their attributes. Restrictions are related to attributes too, and that relation can have an association type specified for it. This can be created by going to the Association Types area in the Catalog Designer menu* |
| **Project** | ‘Project: High Speed Internet’ |
| **Default Value Rule** | N/A (leave blank) |
| **Default Value** | N/A (leave blank) |

1. In addition you must specify what values are applicable to this item attribute. You do this by selecting from the Select Default Value drop-down list. Select 2Mb/s and the Click Save:



Now we will create a similar restriction for the **Download Bandwidth**, using the given bandwidths of **4mbps**.

1. Click on the **Download Bandwidth** attributeunder the **Light Internet Access** component item, and once more click **Add** to add a restriction.
2. In the resulting **Detail** panel fill out the attribute fields as before, and select **4mbps**.



1. Repeat the process described above to restrict the **Upload Bandwidth** and **Download Bandwidth** attribute values for the **Regular Internet Access** and **Heavy Internet Access** component items, giving the restrictions **Names** as follows:

* regularUpload
* regularDownload
* heavyUpload
* heavyDownload

## Exercise 6: Item relations and groups

Items can be related to other items in any way the user wants. There are a couple of predefined relationships in Catalog Manager (i.e. ‘*association types*’), but it is possible to add or create any kind of association type that is required.

As part of this exercise you will be creating a ‘*product*’ item. Remember that this type of item is ‘orderable’, and users who are browsing or searching the catalog will see such items returned.

### Create association type and relate items

In this section we create a new ‘product’ item called ‘**Very High Speed Internet**’ and relate it to three component items in the system. This is done by creating an ‘**association type**’.

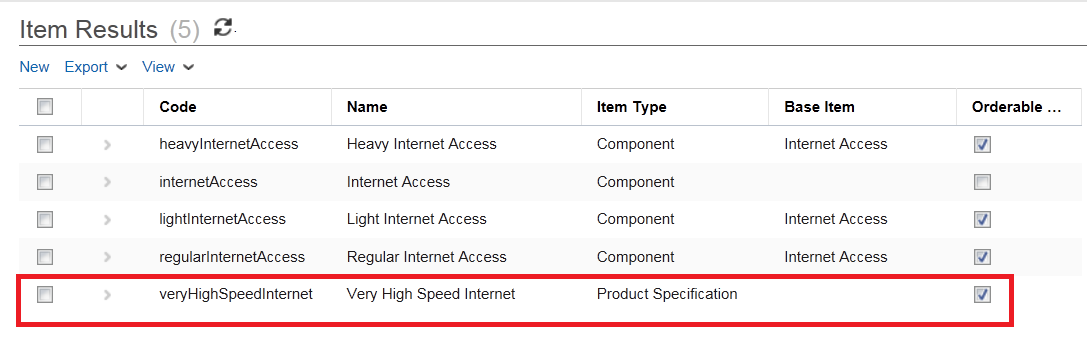
Our first steps, then, will be to create the new **product** item that will be related to the three component items created in Exercise 5: **Light Internet Access**, **Regular Internet Access** and **Heavy Internet Access**.

1. Check that your project is open and visible in the Catalog Designer toolbar area at the top of the screen.
2. Go to *Product > Item in Quick Start Menu*.
3. As before, select New for adding a new item, then enter field values in the **Item Detail** panel as shown in the table below:

|  |  |
| --- | --- |
| **Item Type** | ‘Product Specification’ |
| **Status** | ‘Definition’ |
| **Code** | ‘veryHighSpeedInternet’ |
| **Orderable** | Yes (check box) |
| **Name** | ‘Very High Speed Internet’ |
| **Label** | ‘Very High Speed Internet’ |
| **Description** | N/A (leave blank) |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Base Item** | N/A (leave blank) |
| **Owner** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |



1. Save the item (see above), and check the product in the **Item** panel:



We now need to relate the three **component** items created earlier with this new **product** item. But first it is necessary to create the ‘**association type**’ (i.e. relationship type).

1. Go to *Technical Configuration > Association Type*, then click **Search** in the **Search Criteria** panel.
2. In the **Result** screen underneath, click the **Add** button at the bottom. (Once again, you may have to scroll down to see this button).

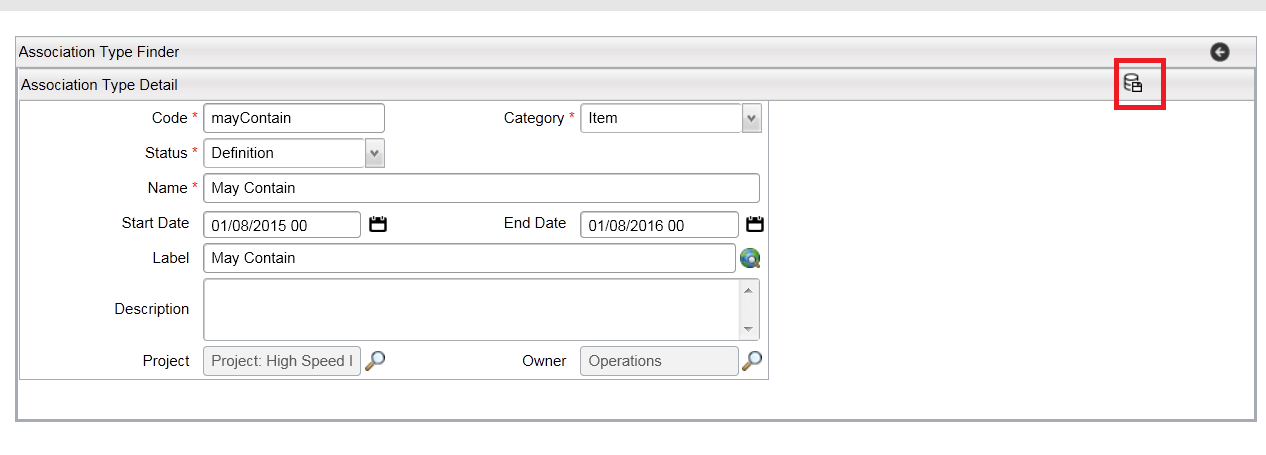


The new **association type** code will be called ‘**mayContain**’, because our **Very** **High Speed Internet** product may (optionally) contain one of the three **Internet Access** components.

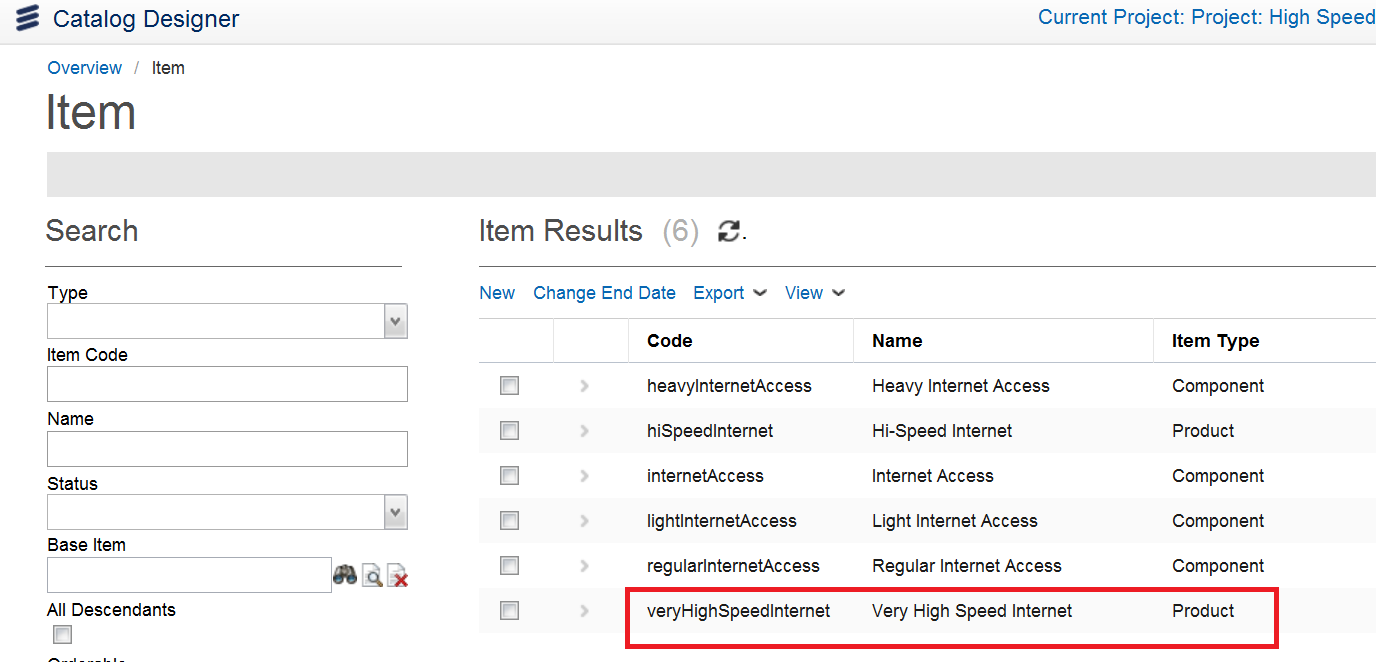
1. For the new **association type** (relationship), enter field values in the **Association Type** **Detail** panel as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘mayContain’ |
| **Category** | ‘Item’  *Means that the new association type sets up a relationship between ‘items’* |
| **Status** | ‘Definition’ |
| **Name** | ‘May Contain’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Label** | ‘May Contain’ |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |

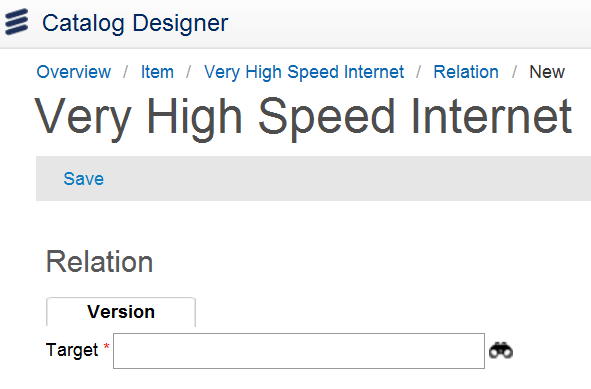
1. **Save** the new association type (icon on the right-hand side of the **Association Type Detail** panel title bar):



1. Return to the **Item** panel (*Catalog Designer > Items*), and expand the tree so you can see the **Very High Speed Internet** product item created earlier.

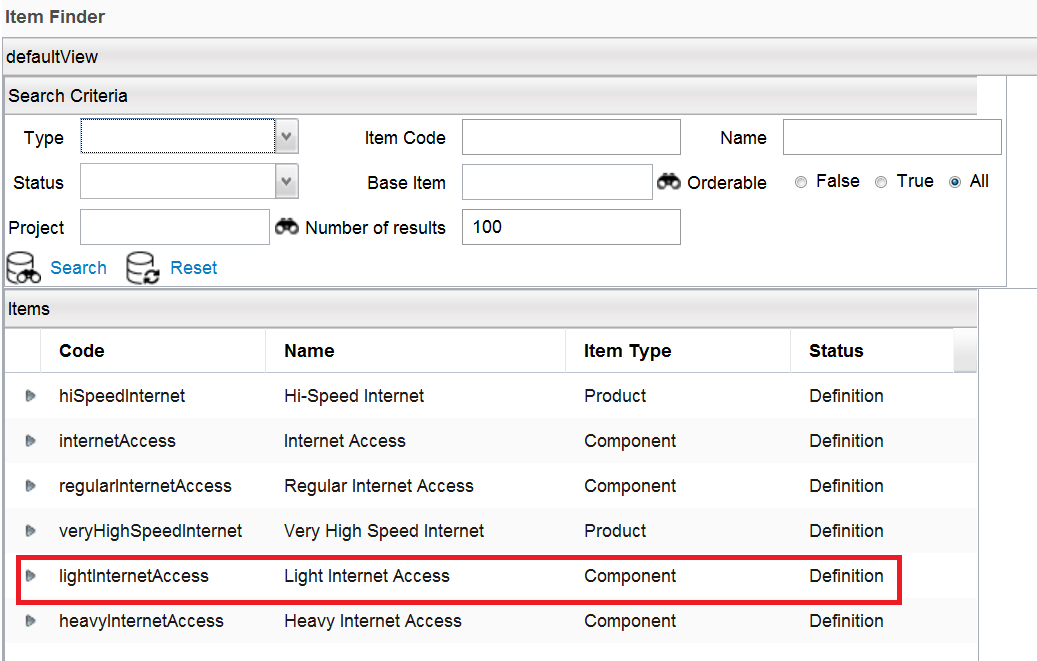


1. Expand the **Very High Speed Internet** item and click on **Relations** tab and then **New**, so we can add a new ‘association’ to this product. The **Relation Detail** panel will appear:



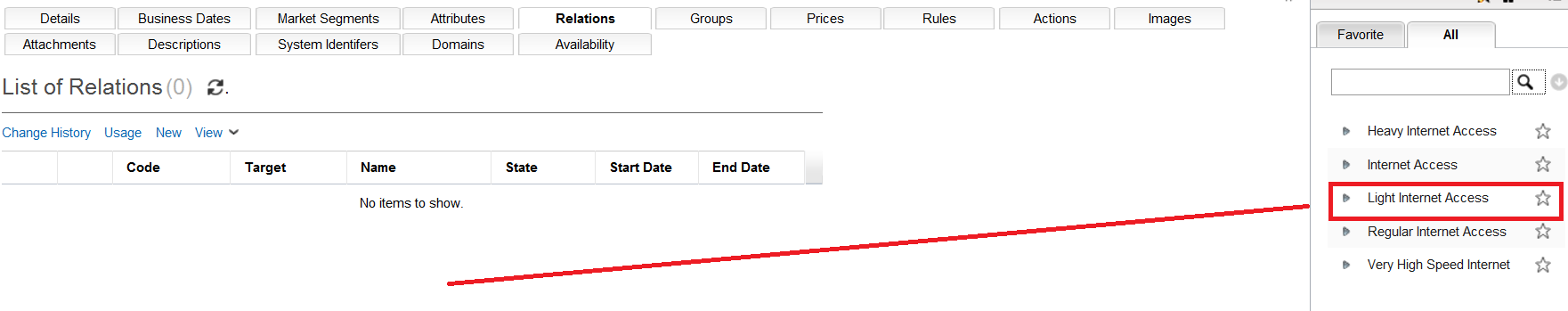
First we need to identify the target item we want to associate the **Very High Speed Internet** item with.

1. In the **Relation Detail** panel, click on the **finder** icon next to the **Target** field.
2. In the list of items returned, select and enter **lightInternetAccess** by clicking on the blue arrow icon to the left of the row, or simply double-click on the row.



You will see that by selecting this item the relation inherits the ‘**Code**’ and ‘**Name**’ values of the target item.

1. An alternative method is to drag and drop the Light Internet Access from the right panel under the List of Relations.

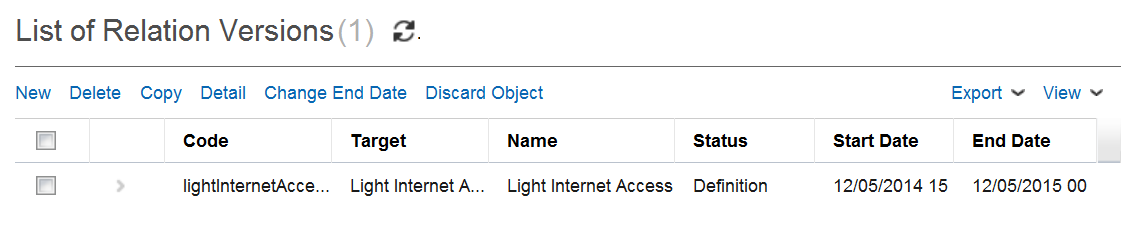


1. Complete the remaining fields as shown in the table below:

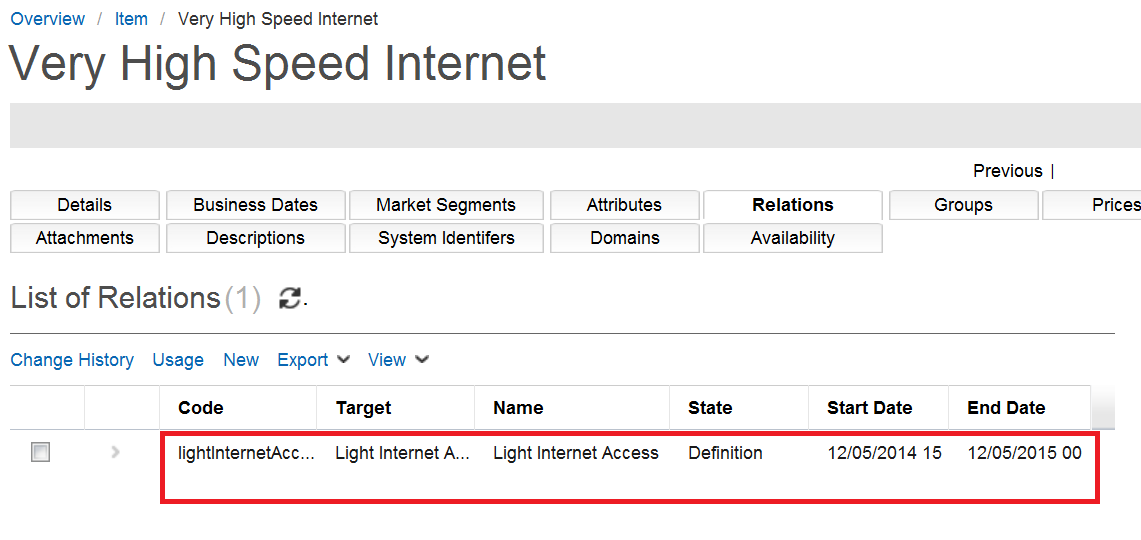
|  |  |
| --- | --- |
| **Type** | ‘May Contain’  *There are also three predefined relationships in this list:* ***Contains****,* ***Related*** *and* ***Dependent*** |
| **Status** | ‘Definition’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Label** | ‘Light Internet Access’ |
| **Cancel** | N/A (leave blank) |
| **Sequence** | N/A (leave blank) |
| **Min Quantity** | ‘0’  *Means that the* ***Light Internet Access*** *item doesn’t have to be associated with* ***Very High Speed Internet*** |
| **Max Quantity** | ‘1’  *Means that only one such relationship will be allowed* |
| **Default Cardinality** | N/A (leave blank)  *Allows you to specify a number for the item relation. If the parent item is added to the basket, and the item relation association it contains, then the same number of child items are automatically added to the basket as well* |
| **Project** | ‘Project: High Speed Internet’ |



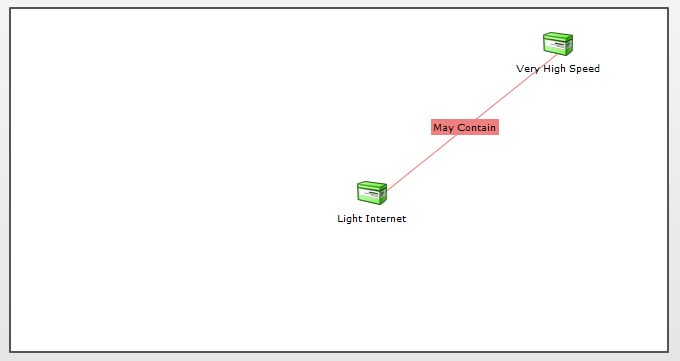
1. Click **Save** (icon on the **Relation Detail** title bar, see above screenshot). New Version will appear in List of Related Versions at bottom of screen.



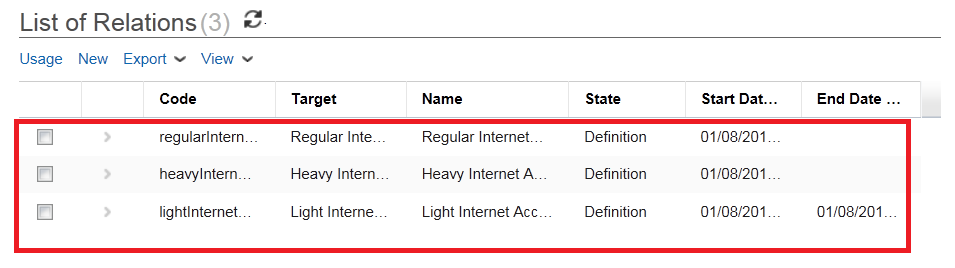
1. Now expand the **Relations** folder to see the new relation:

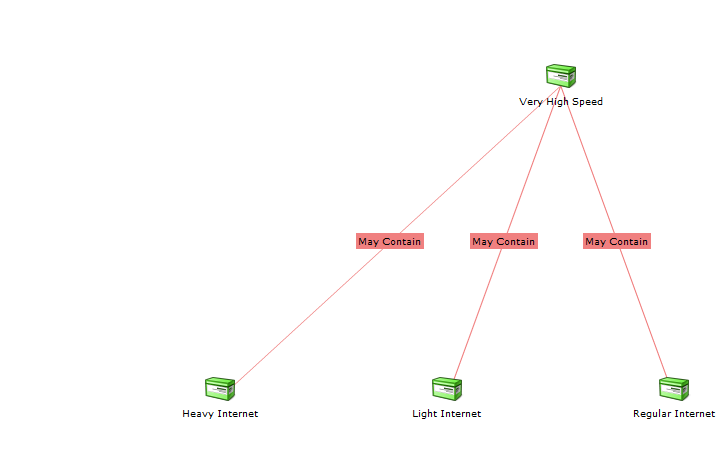


You can also see the relationship graphically at the bottom of the Item page:



1. Now repeat the process described above to create an association between **Very High Speed Internet** and the **Regular Internet Access** component item.
2. And finally, follow the same process one more time to create an association between **Very High Speed Internet** and the **Heavy Internet Access** component item.
3. Check that all three new relations are now visible in the **Item** panel under **Very High Speed Internet**:





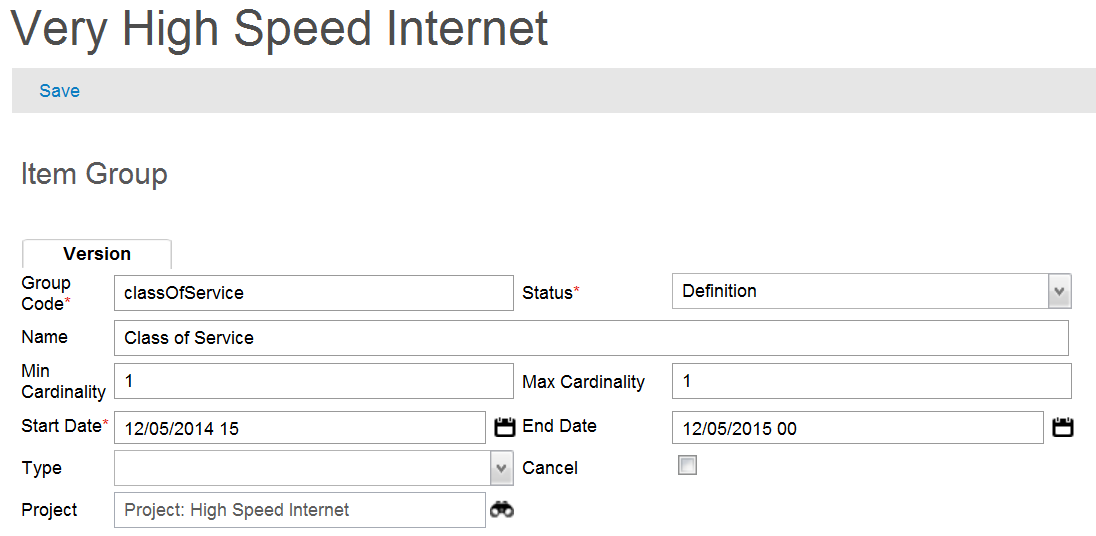
*Note: At runtime the user will be asked to select only one of the three optional related items:* ***Light Internet Access****,* ***Regular Internet Access*** *and* ***Heavy Internet Access****. When a customer is buying a* ***Very******High Speed Internet*** *service, he/she requires only one ‘flavor’ of this type of service.*

### Create item group and add members

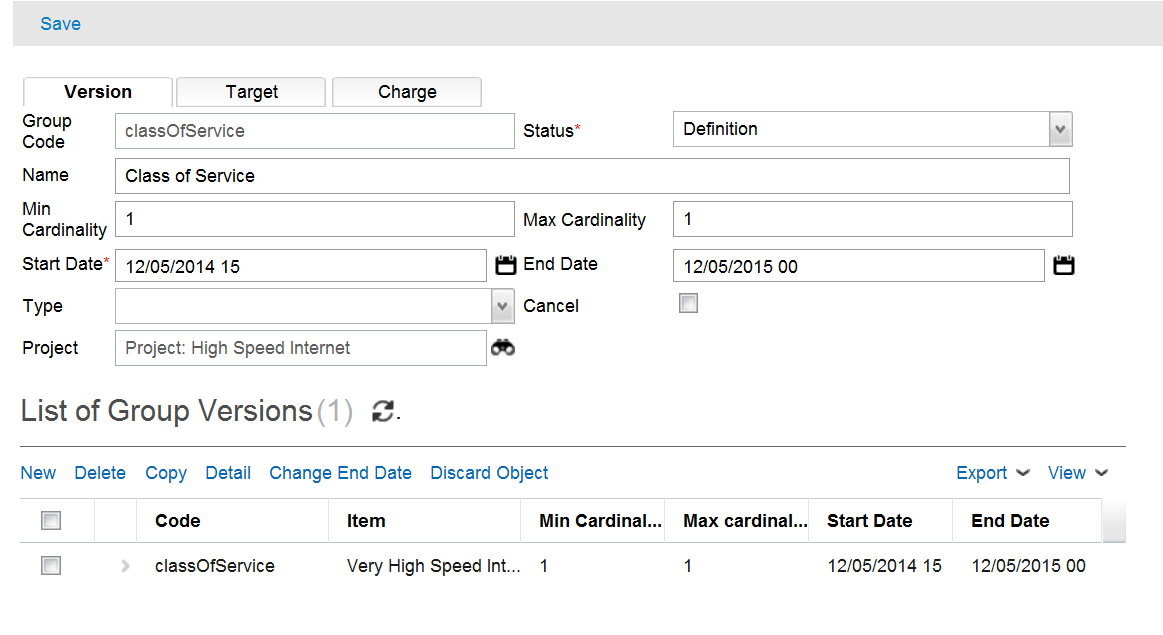
As a service provider we need to be able to force the user to select one of the three **Very High Speed Internet ‘**flavors’. To achieve this we must now create an item ‘**Group**’, and specify the cardinality of this new group.

1. Click on New on the **Groups** tab under the **Very High Speed Internet** item, then complete the fields as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘classOfService’ |
| **Type** | N/A (leave blank) |
| **Status** | ‘Definition’ |
| **Cancel** | N/A (leave blank) |
| **Name** | ‘Class of Service’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Min Cardinality** | ‘1’  *Means that the user must select at least one item* |
| **Max Cardinality** | ‘1’  *Means that the user cannot select more than one item* |
| **Project** | ‘Project: High Speed Internet’ |

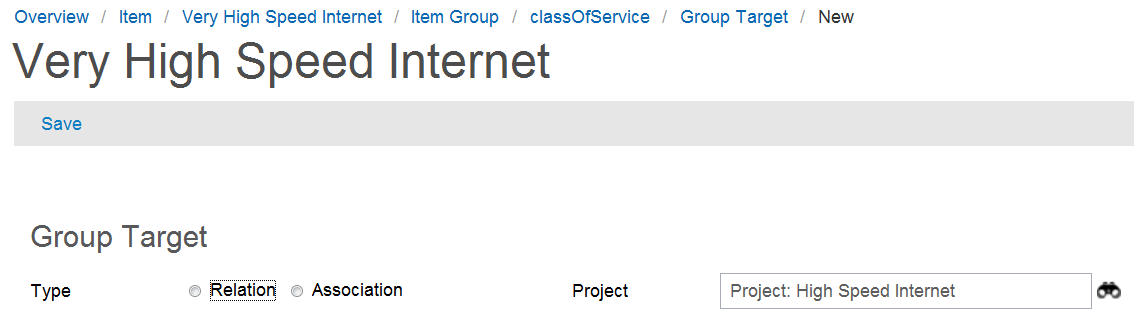


1. Click **Save** (icon on the **Group Detail** title bar, see above screenshot). The new item will appear under List of Group Versions at bottom of page.



Having created the new group, we must now specify which items should be ‘members’ of this group.

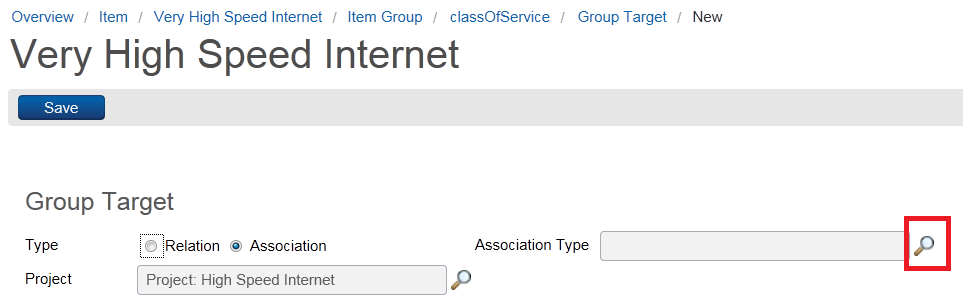
1. Click on the **Targets** tab under the attribute details in the **Group Detail** panel, and then click **New**. This will result in a **Target Detail** box appearing underneath.

**

In the **Target Detail** box you will see two options for ‘**Type**’: ‘**Relation**’ and ‘**Association**’ (see above screenshot). We now have to choose one of these types.

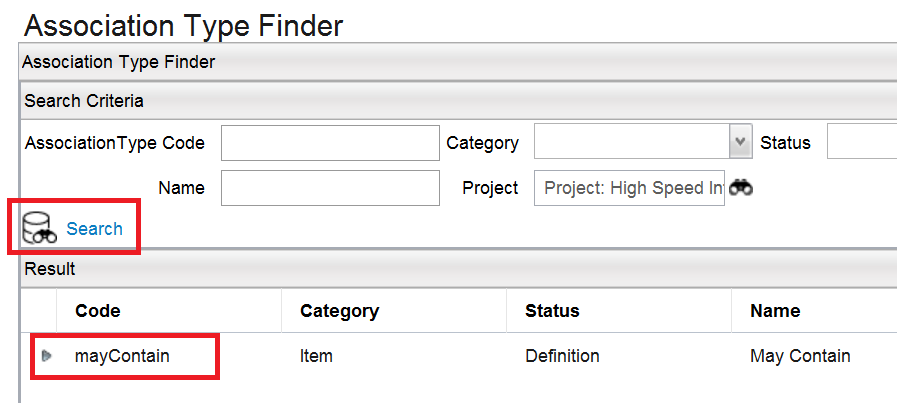
It is possible to add the targets one by one. This is done by selecting the **Relation** option. This would be appropriate where there may be only a couple of items to add to the group. But if there are a large number of items of the *same* association type to add to the group, it is better to choose the **Association** option. This will allow us to bring in all the relations of the **association type**, and add them to the group in one operation.

1. Select the **Association** radio button, and then click on the **Finder** icon next to the **Association Type** field.

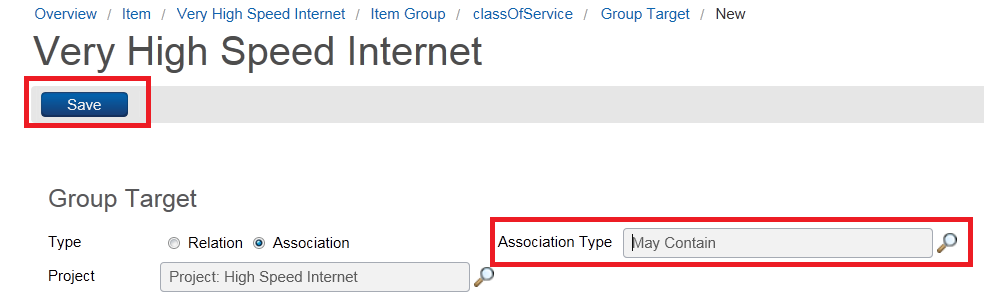


This action results in a user being forced to select one - and only one - item from the list of related items (i.e. service offerings), when creating an order.

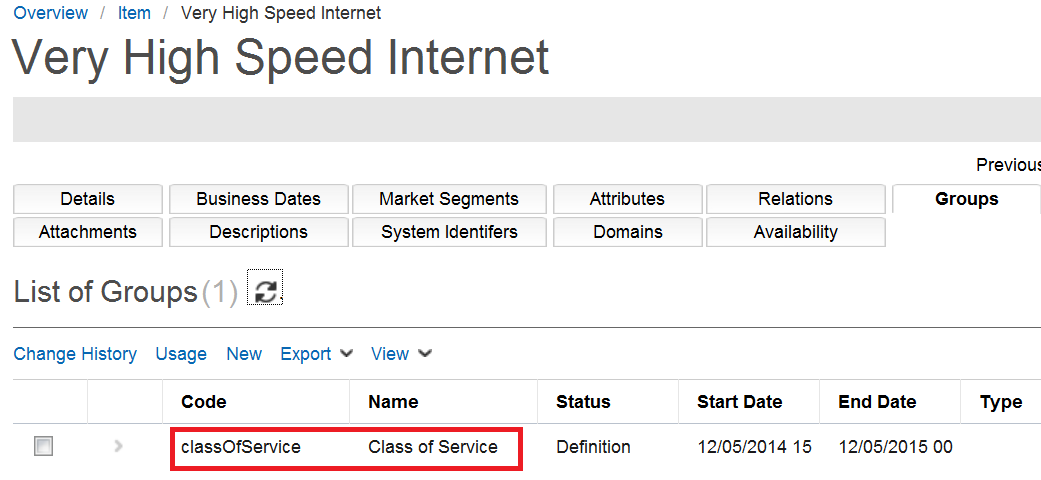
1. Click **Search** when the project name has been entered into the **Project** field, and you will see the only relation code currently associated with this project: ‘**mayContain**’:

**

1. Select and enter the relation code by using the blue arrow, or by double-clicking on the row (see above screenshot). You will then see the new association type **‘May** Contain’ entered in the **Target Detail** panel **Association Type** field:

**

1. Click **Save** (icon in the **Target Detail** panel title bar - see above screenshot).
2. Expand the **Groups** folder under **Very High Speed Internet** to see the new group ‘**Class of Service**’:



Module 6: Catalog Hierarchy

Module 6 takes a look at how to create **Catalog Hierarchies**, add items to the hierarchy, and then *test* the catalog before it goes live. As a result of these actions, you can then be sure that your resulting catalog if structured in a way that helps users manage service requests effectively and with accuracy.

## Exercise 7: Catalog hierarchy and testing

In this exercise we will create a new **Catalog Hierarchy**, add a ‘product’ item to one of the hierarchy levels, then go into **Test** mode to test the Catalog.

### Create catalog hierarchy

First let us create a new **Catalog Hierarchy**. This mechanism allows a company to organize their items into meaningful categories.

The following diagram provides a view of the hierarchy levels and names which we shall create:



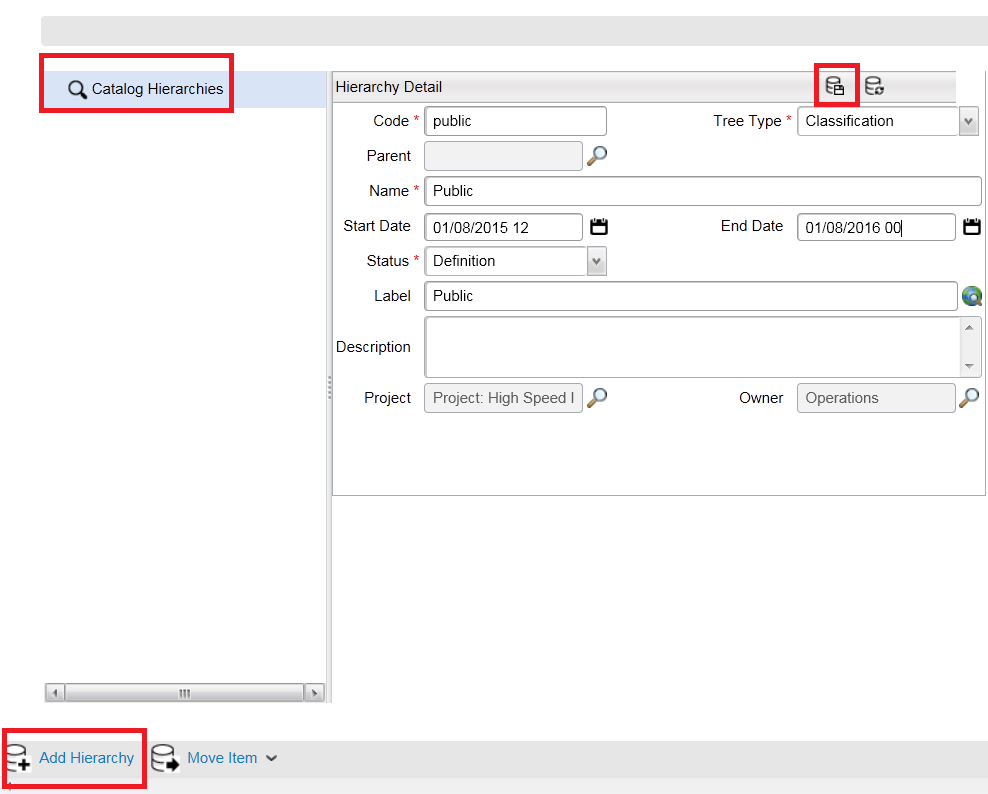
First we need to build out the two top-level nodes called ‘Public’ and ‘Private’.

1. Check that your project is open and visible in the left panel.
2. Go to *Rules > Hierarchy in the Quick Start Menu*:

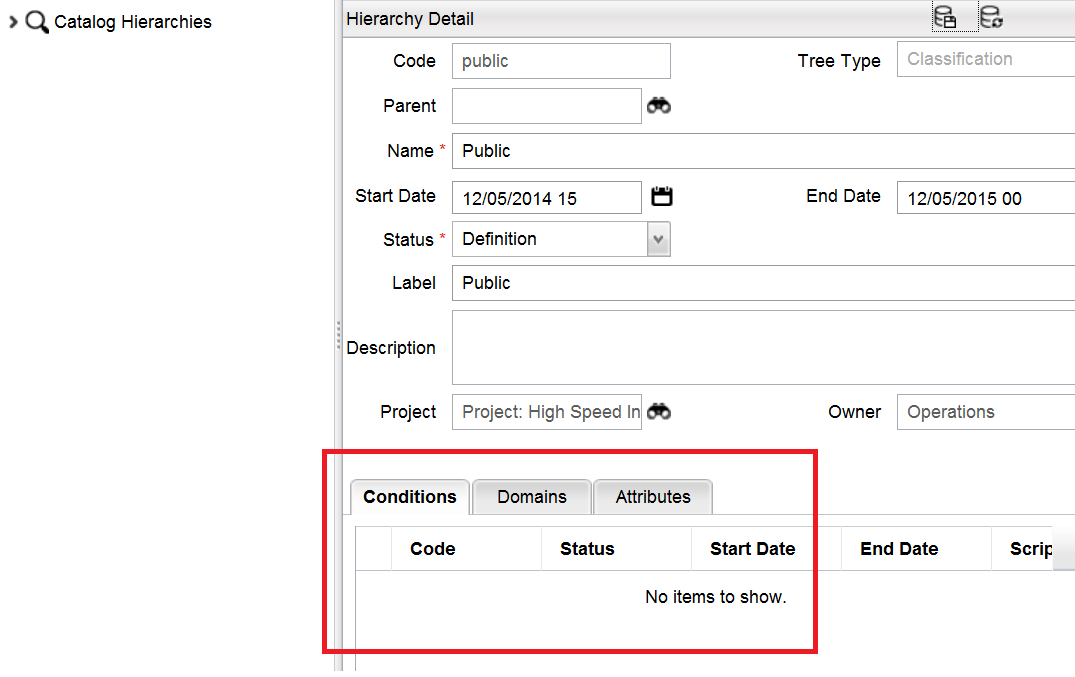


1. Select the **Catalog Hierarchies** row in the left panel, and then click **Add Hierarchy** at the bottom.
2. To create the first of the top-level hierarchy levels (from the diagram above), enter field values in the **Hierarchy Detail** panel, as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘public’ |
| **Tree Type** | ‘Classification’  *Choose* ***Classification*** *for the ‘top-down’ approach. This is easier to understand, but can be tedious to build. Choose* ***Availability*** *for the ‘bottom-up’ approach. Here objects can be multiple-selected into different parent objects, and is better if there are a lot of redundancies* |
| **Parent** | N/A (leave blank)  *The name of the top-level hierarchy node goes here, but since this new node is the top level, no parent reference is required* |
| **Name** | ‘Public’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Status** | ‘Definition’ |
| **Label** | ‘Public’ |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |

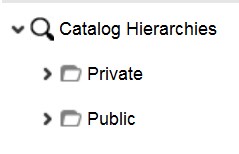


1. Click **Save** (using the button in the **Hierarchy Detail** panel title bar), then expand the tree in the left-hand panel and check that the newly-created node is now visible in the hierarchy. (You will see that additional tabbed windows appear on saving this object: **Conditions**, **Domains** and **Attributes**. We will not be using these in this exercise).



*Note: Any additional hierarchies you may see in the list were of course created at another time by a different user.*

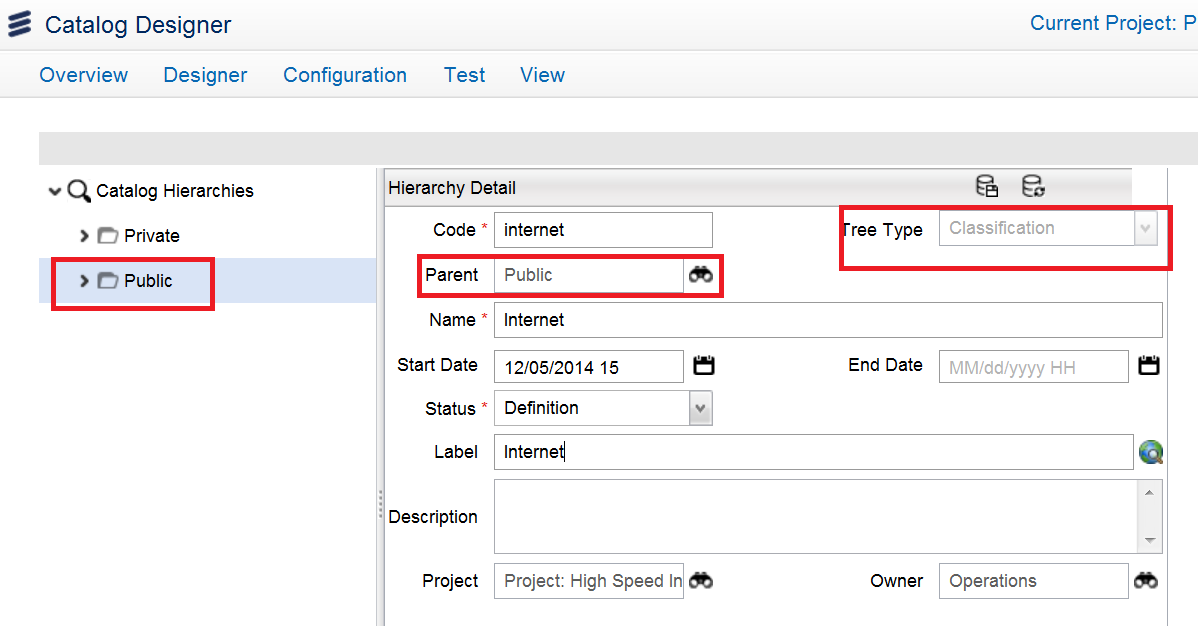
1. Using the same process, add a second node called ‘**Private**’ at the same ‘root’ level, then check that both nodes are visible in the hierarchy on the left:



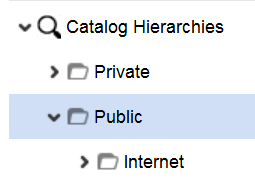
We now have the two hierarchy nodes at ‘root’ level which can be built out: **Public** and **Private**. IN the next few steps we will create a second-level node called ‘**Internet**’, which will be a ‘child’ of **Public**.

1. Select **Public** in the **Catalog Hierarchies** panel, and then click **Add Hierarchy** at the bottom of the screen. When you do this, notice that the **Parent** field is *automatically* populated with ‘**Public**’, since this will be the nominated *parent* of the new second-level node we are creating here.
2. Complete all other fields in the same way as you did when creating the **Public** and **Private** hierarchy levels, giving this node the name ‘**Internet**’.

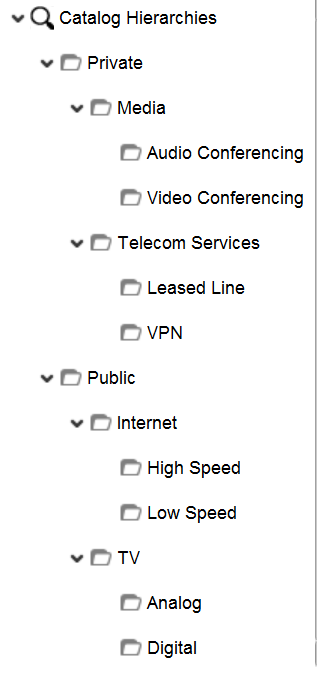
*Note: The* ***Tree Type*** *value of ‘****Classification****’ has been automatically inherited from the parent node, and cannot be changed.*



1. **Save** the **Internet** hierarchy, then expand the tree in the left panel to check that it visible and correctly positioned in the hierarchy structure:



1. Continue building the different catalog hierarchies to complete the structure provided at the beginning of this exercise.



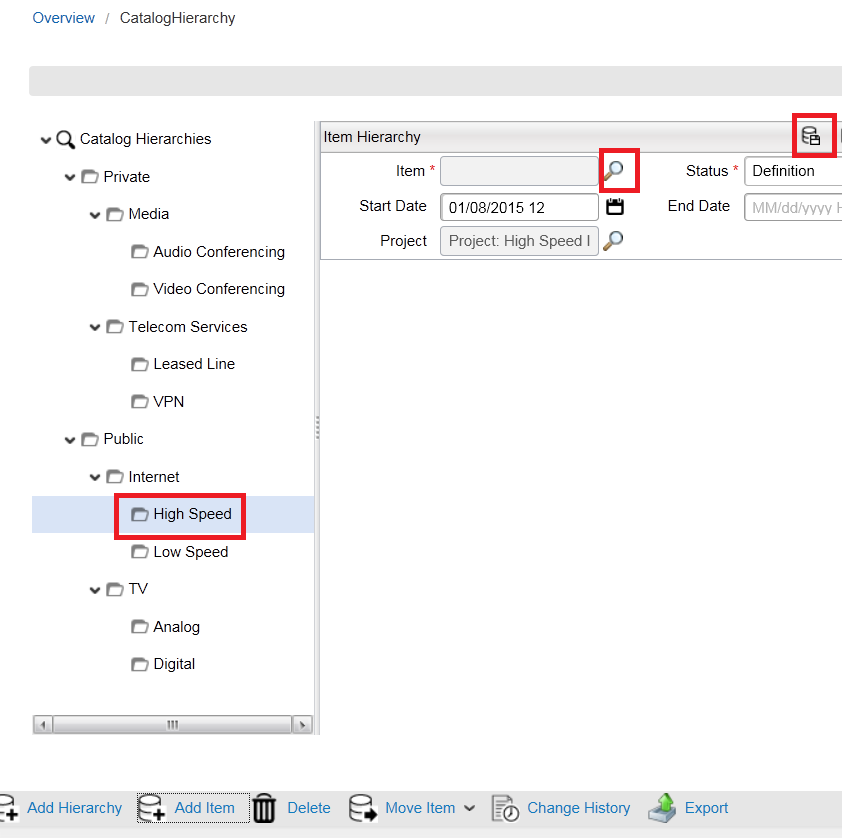
*Note: If you make a mistake, and put a node in the wrong place in the hierarchy, you can correct your error by selecting it and manually changing the* ***Parent*** *field value.*

### Add item to hierarchy node

In this section we will demonstrate the process of adding an item to a node in the **catalog hierarchy** that we just created. In this example the item will be ‘sellable’ or ‘orderable’, - i.e. a **product** item. This gives a company the means to apply all their products to the different categories or departments in their organization.

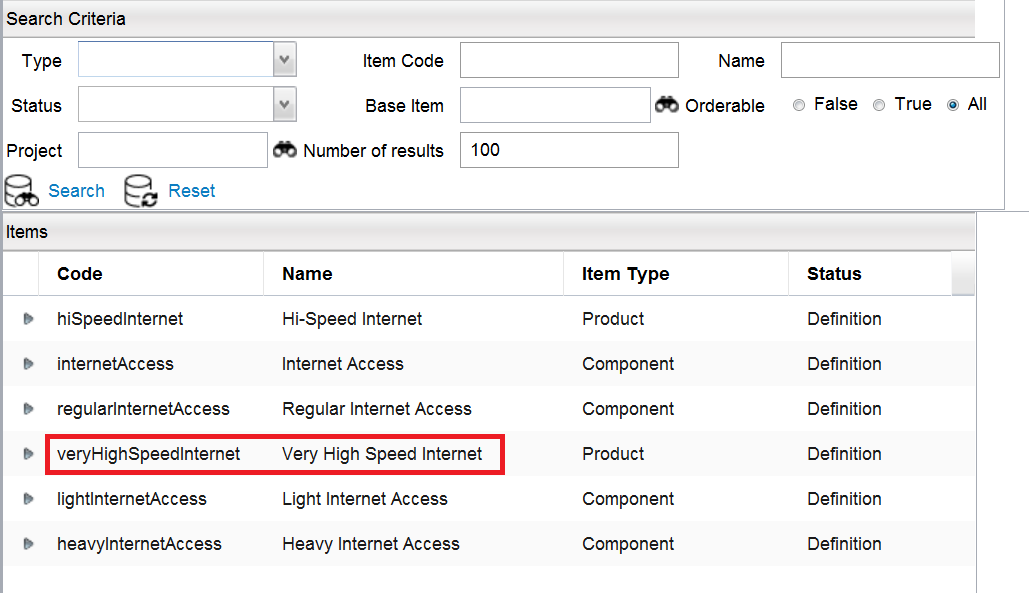
Then, when someone queries the catalog, he/she will be able to request a list of items that reside at a particular node in the hierarchy. The **name** of a particular node will be used for searching the catalog.

1. In the catalog hierarchy click on the ‘**High Speed**’node under **Internet** (from the **Public** folder), then click **Add Item** at the bottom of the screen.
2. Now click on the **finder** icon next to the **Item** field:



A list of created items will appear straightaway in the **Items** panel. Remember that only ‘orderable’ items are added to a catalog hierarchy.

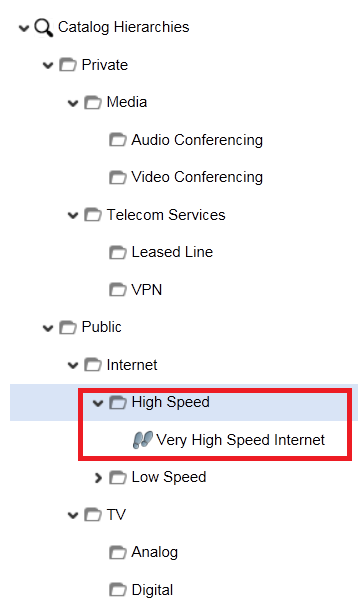
1. Select and enter the only **product** item we have created in our project so far: ‘**Very High Speed Internet**’:



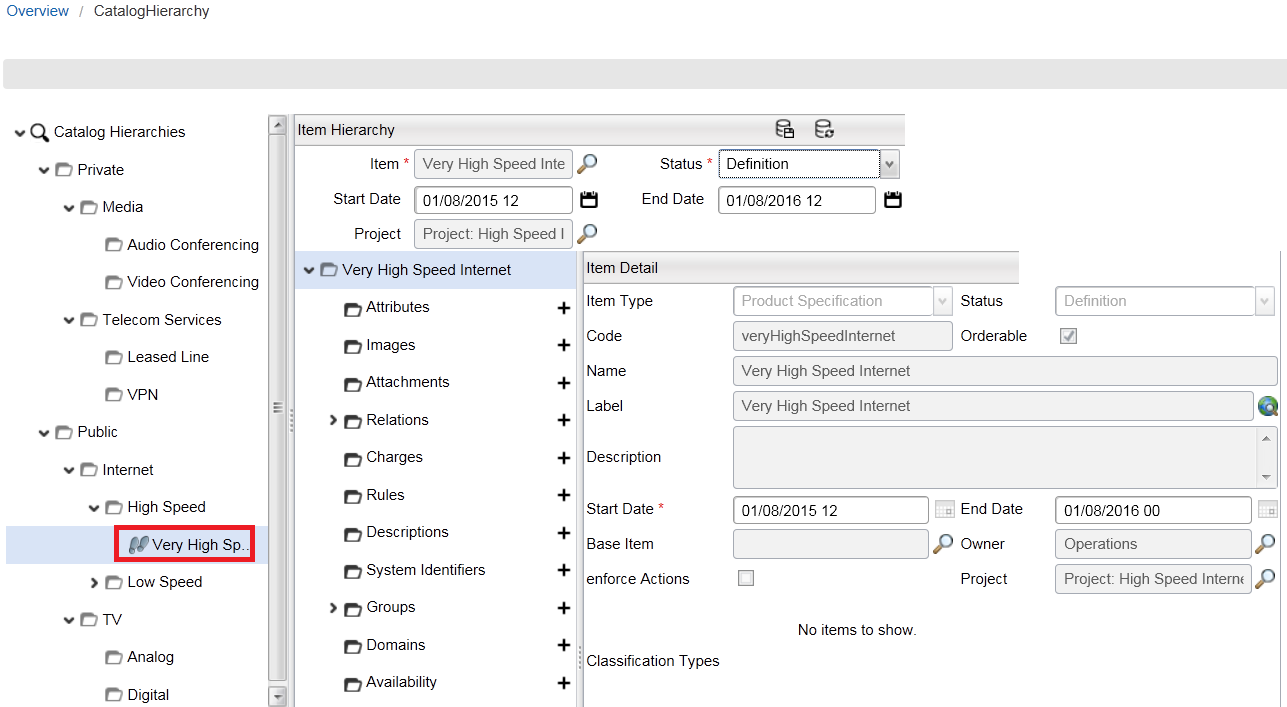
1. Now complete the remaining fields, as shown in the table below:

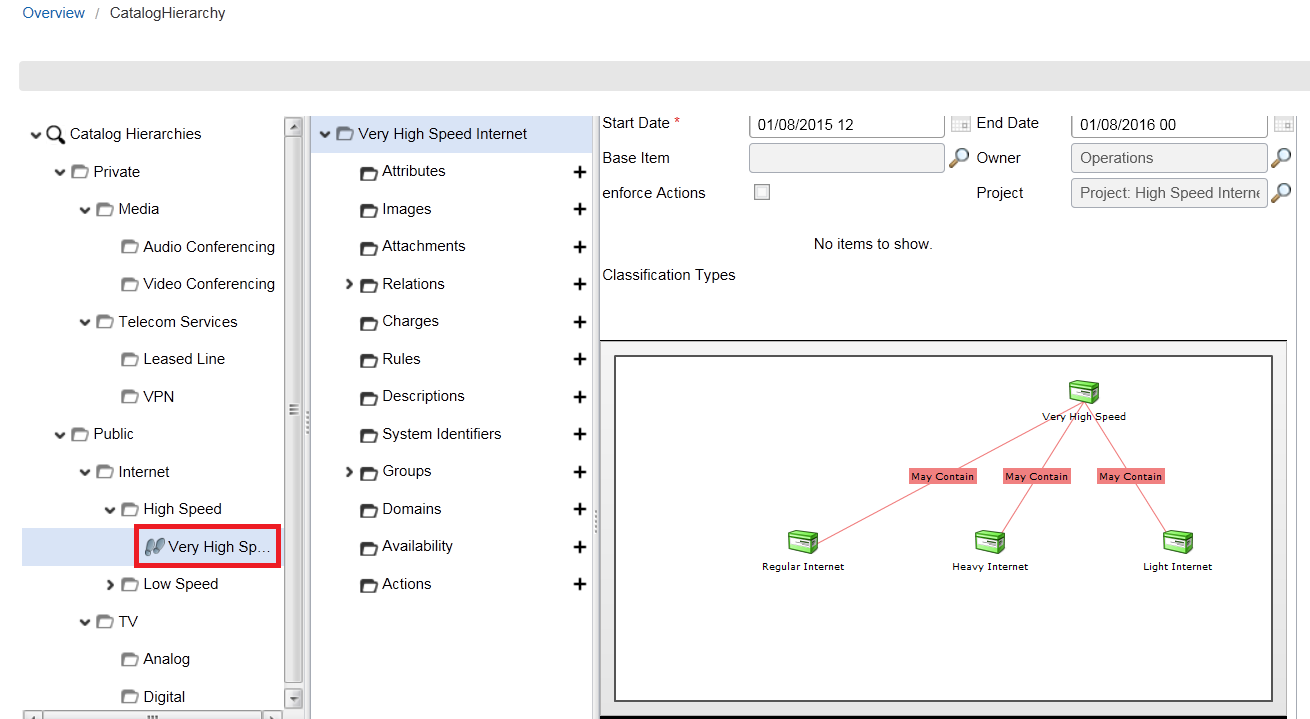
|  |  |
| --- | --- |
| **Status** | ‘Definition’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Project** | ‘Project: High Speed Internet’ |

1. Review the updated hierarchy, and check that the newly-added item is visible under the correct node (**High Speed**).



1. Select the item and you will see in the right-hand panel not only the item’s main attribute fields, but also any related attributes, with the list of potential items and relationships that could be added. (We saw this list before in the main **Item** panel).





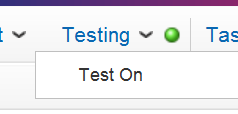
### Test catalog hierarchy

We should now test our catalog, make sure that everything is working as expected, and ‘debug’ as required.

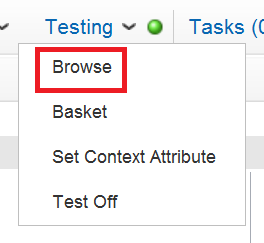
The user-friendly testing mechanism in Catalog Manager, checking the configuration we have done in the Catalog Designer GUI, replaces the need to dig around in the metadata configuration to locate and fix any perceived errors. It allows us to easily test the structure of the hierarchy, as well as the ordering process using the catalog ‘basket’. It simulates the process of a user creating an order, adding items to the basket, and, if necessary, configuring those items whilst in the basket.

First we will test the catalog hierarchy structure itself, using the hierarchy as part of a simulated order.

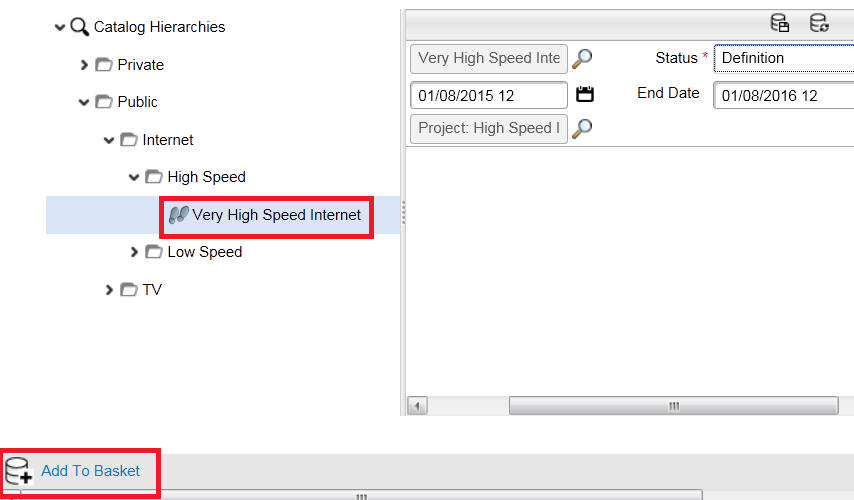
1. Go to *Test Mode > Test On*:



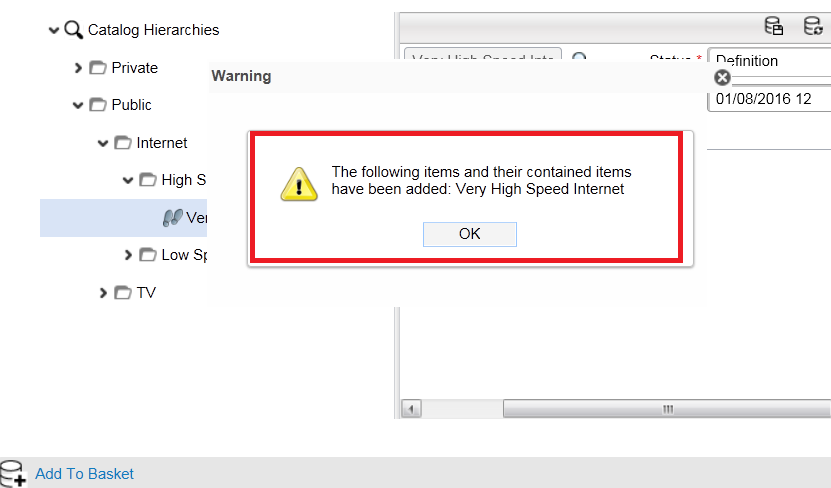
1. Then, immediately after doing this, go to the same menu again, this time choosing *Test Mode > Browse*:



1. When your catalog hierarchy appears, drill down and select the **Very High Speed Internet** product item we added to the **High Speed** node earlier (*Public > Internet > High Speed > Very High Speed Internet*).

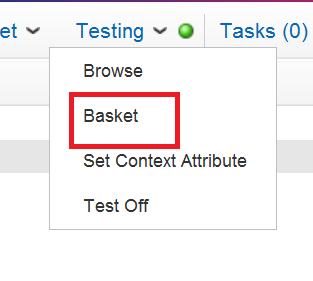


1. Click on **Add to Basket** at the bottom of the screen. This simulates an item being added to an order.
2. Review the confirmation message and click **OK**:



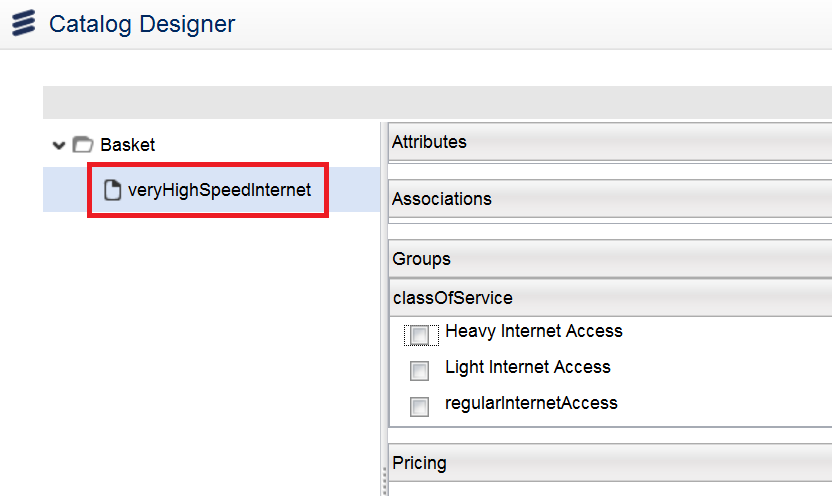
Now we will test the *basket*.

1. Go to *Test Mode > Basket* :



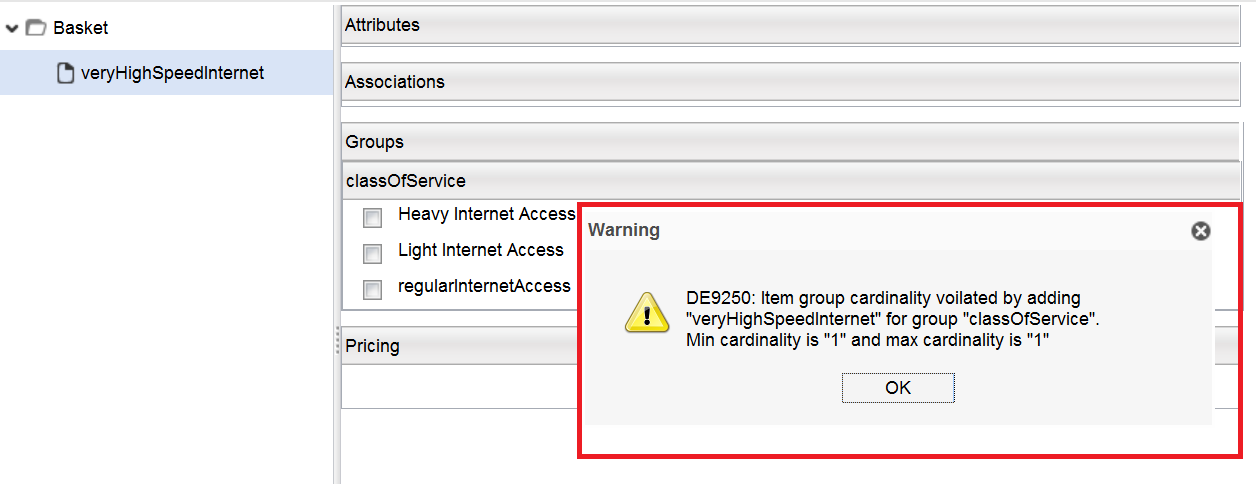
This will show you all the items that have been added to the basket (just one item here, of course).

1. Select the **Very High Speed Internet** item, and you will see all the associations (relations) that you created earlier as part of the **Class of Service** group:



As part of the testing activity we will now validate the minimum and maximum constraints we specified when we created the group.

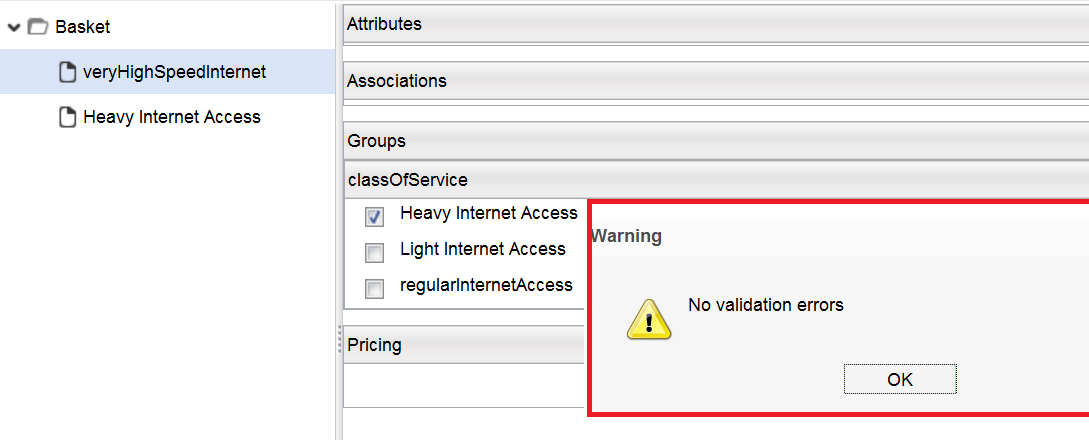
1. Without selecting any of the items in the **Class of Service** group, click the **Validate** button at the bottom of the screen (see screenshot above). You should get a warning message with the reason for the validation failure (‘*Item group cardinality violated’*):



1. Click **OK** to close the warning dialog box.
2. Now select two services from the list: **Heavy Internet Access** and **Light Internet Access**, then click the **Validate** button again. Once again you should get the validation failure message.
3. Close the warning dialog box by clicking **OK**.

Now we will select just one service from the list.

1. Remove the tick from the **Light Internet Access** option and acknowledge the confirmation message.
2. Ensure that the **Heavy Internet Access** option is still selected, and then click the **Validate** button once more. This time you should get a confirmation message that indicates there are no validation errors:

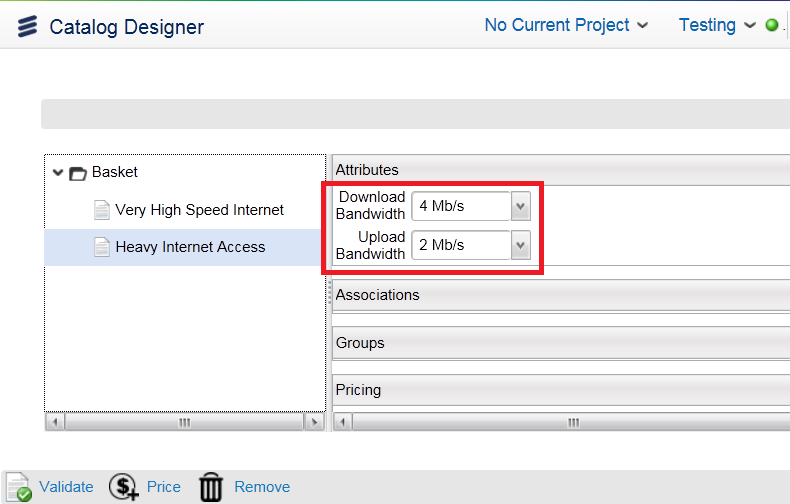


1. Click **OK** to close the confirmation dialog box (see above).

You can also provide values for the different attributes that belong to the added item in the catalog hierarchy (i.e. make configuration changes to an item while it’s still in the basket).

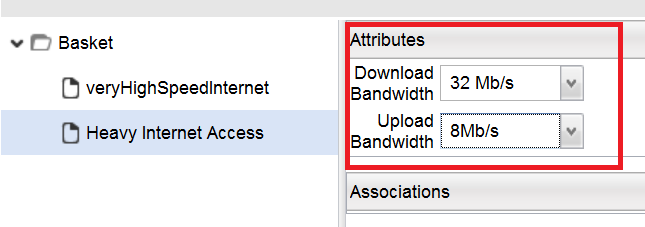
1. Select the **Heavy Internet Access** in the left panel.

On the right-hand side you will now see the **Download Bandwidth** and **Upload** **Bandwidth** attributes we created earlier, with pull-down menu bandwidth options:



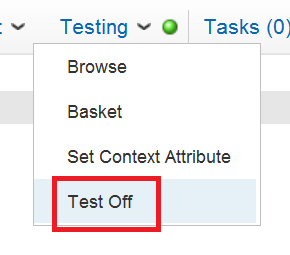
The default bandwidth values in these lists are a direct result of the ‘**restrictions**’ we created in a previous exercise.

1. Open the **Download Bandwidth** pull-down list and select ‘**32mbps**’.
2. Open the **Upload Bandwidth** pull-down list and select ‘**8mbps**’.



That completes the testing activity.

1. Turn off the test mode by going to *Test Mode > Test Off*:



The Catalog Designer screen will go to the Overview screen, awaiting the next user action.

Module 7: Pricing

In this Module we take a look at the main considerations for **pricing**, - how to create and manage **charge types**, how to associate them with **items**, and how to **test** the basket pricing mechanism. Note that the catalog only *lists* charges and prices, and does not make any calculations. It is not a ‘billing’ service, just a repository where all the information can be stored.

Charge types can be created *independently* from the items, thereby making it possible for the *same* charge type to be associated with *different* items. It works the other way round as well - *multiple* charge types can be associated with a *single* item.

## Exercise 8: Charge types

In this exercise we create several different charge types, and associate them with ‘**product**’ and ‘**component**’ items in the catalog, - items we created in an earlier exercise. We also go back into ‘**test**’ mode, and explore how the catalog returns pricing information for the different service offerings.

Charge types can indicate things like the amount, currency, frequency, and start/end dates of the actual charges, giving users the power to associate an item with a particular charge and its attributes.

The act of pricing items happens in two different steps:

* Create the charge types
* Associate the item(s) with the charge types

Testing will have to be done as well, of course.

### Create charge types

In this section we create seven different charge types in all. (Remember the scenario overview diagram in Exercise 3?)

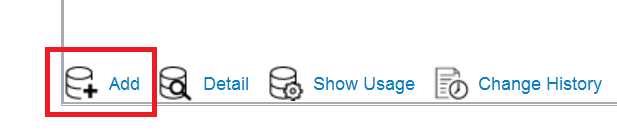
* ‘**Setup Fee**’ (a one-time charge of $100)
* 2 x ‘**Heavy Internet Access**’ monthly subscription charges ($30 and $25)
* 2 x ‘**Regular Internet Access**’ monthly subscription charges ($20 and $15)
* 2 x ‘**Light Internet Access**’ monthly subscription charges ($12 and $10)

The **Setup Fee** charge type will be associated with the *product* item called ‘**Very High Speed Internet**’, created earlier. Clearly the six subscription charge types need to be associated with the three **Internet Access** *component* items, also created earlier. (The system allows us to create charge types for ‘non-orderable’ items).

1. Check that your project is selected and opened.
2. Go to *Pricing & Tax > Charge Type in the Quick Start Menu*:

We will now create the ‘**Setup Fee**’ charge type for the ‘**Very High Speed Internet**’ *product* item.

1. Click **Search** in the **Search Criteria** panel, and then click **Add** at the bottom of the screen. (As before, you may have to scroll down to see this button).

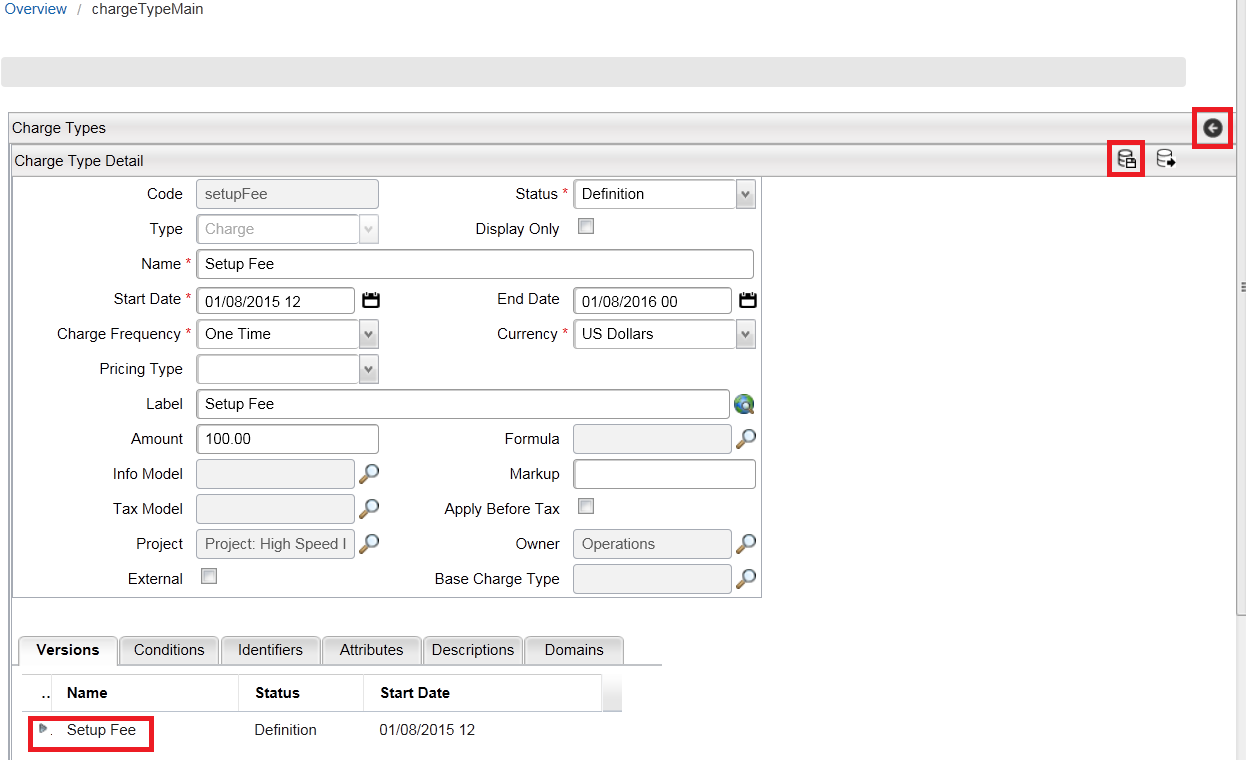


**Note**: If you do not see the Add option, make sure your project is open

1. Complete field entries as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘setupFee’ |
| **Status** | ‘Definition’ |
| **Type** | ‘Charge’  *The other options are ‘****Discount****’ and ‘****Cost****’, providing for the different types of charges which can be applied* |
| **Display Only** | N/A (leave unchecked)  *Used when the charge, discount or cost will appear in the invoice only, and not used to calculate the price of the item (e.g. a free handset)* |
| **Name** | ‘Setup Fee’ |
| **Start Date** | [Today’s date] |
| **End Date** | [One year from today] |
| **Charge Frequency** | ‘One Time’  *The values in this pull-down list come from* ***code tables****, and can be modified, removed or added to.* |
| **Currency** | ‘US Dollars’  *Appears by default. We set this as the default currency when we configured the catalog in Exercise 1. However you can change this option manually at any time* |
| **Pricing Type** | N/A (leave blank)  *The default options are* ***Flat Rate****, ‘****Flat Rate Band****’, ‘****Band****’ and ‘****Stepped****’.* |
| **Label** | ‘Setup Fee’ |
| **Amount** | ‘100’  *For charges that can be applied as a ’fixed amount’ under any circumstances* |
| **Formula** | N/A (leave blank)  *For charges that are specified by a formula. The charge might change because of the customer’s geographical location, type of customer (residential or commercial), time of year, etc. This is done via code (i.e. a ‘****rule****’) that specifies the required logic* |
| **Info Model** | N/A (leave blank) |
| **Markup** | N/A (leave blank)  *The percentage of desired price increase* |
| **Tax Model** | N/A (leave blank) |
| **Apply Before Tax** | N/A (leave unchecked)  *Tax can be added to the charge if required. The tax is added as part of charge creation in the basket* |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |
| **External** | N/A (leave unchecked) |
| **Base Charge Type** | N/A (leave blank – charge can be based on an existing charge type) |

1. Click **Save** in the **Charge Type Detail** panel title bar, and you will see the new charge type in the **Versions** tab underneath. The ‘**Setup Fee**’ charge type has now been created:



1. Click Save and then the back arrow in the **Charge Types** title bar (see screenshot above) to go back to the Charge Types **Result** screen.
2. Click **Add** once more to create another new charge type, - this time for the ‘**Heavy Internet Access**’ *component* item.

Remember from the information provided at the start of this section (and from the earlier scenario overview) that we have two levels of charges against each of the three **Internet Access** services. We will call these ‘**Heavy Subscription** **1**’ and ‘**Heavy Subscription 2**’, etc.

1. Complete field values as shown in the table below for the first of the two charge types associated with **Heavy Internet Access** (‘**Heavy Subscription 1**’):

|  |  |
| --- | --- |
| **Code** | ‘heavySubscription1’ |
| **Status** | ‘Definition’ |
| **Type** | ‘Charge’ |
| **Display Only** | N/A (leave unchecked) |
| **Name** | ‘Heavy Subscription 1’ |
| **Start Date** | [Today’s date] |
| **End Date** | [One year from today] |
| **Charge Frequency** | ‘Monthly’ |
| **Currency** | ‘US Dollars’ |
| **Pricing Type** | N/A (leave blank) |
| **Label** | ‘Heavy Subscription 1’ |
| **Amount** | ‘30’ |
| **Formula** | N/A (leave blank) |
| **Info Model** | N/A (leave blank) |
| **Markup** | N/A (leave blank) |
| **Tax Model** | N/A (leave blank) |
| **Apply Before Tax** | N/A (leave unchecked) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |
| **External** | N/A (leave unchecked) |
| **Base Charge Type** | N/A (leave blank – charge can be based on an existing charge type) |

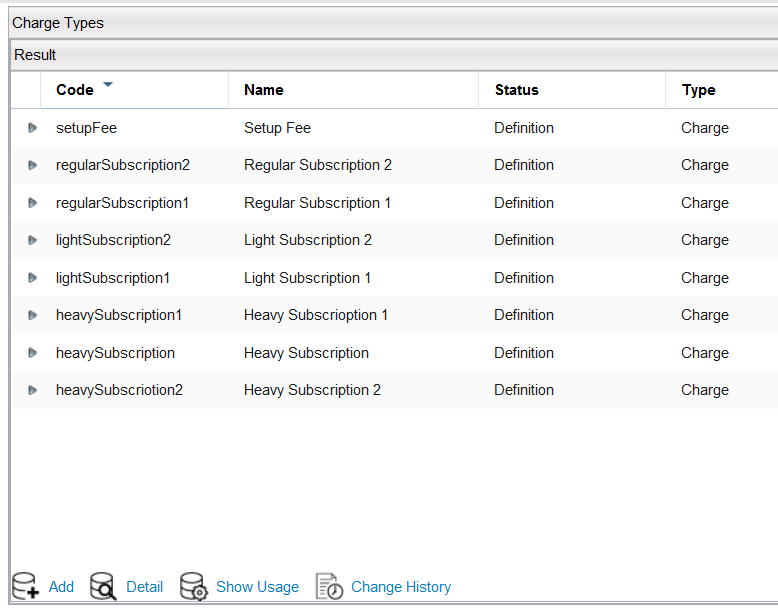
1. **Save** the new subscription charge type, using the button in the **Charge Type Detail** panel title bar:



1. Now repeat the process described above to create the remaining five charge types: **Heavy Subscription 2**, **Regular Subscription 1, Regular Subscription 2**, **Light Subscription 1** and **Light Subscription 2**. The fields which require values for these remaining four charge types are provided in the table below. (Leave all other fields blank or unchecked).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code** | ‘heavySubscription2’ | ‘regularSubscription1’ | ‘regularSubscription2’ | ‘lightSubscription1’ | ‘lightSubscription2’ |
| **Status** | ‘Definition’ | ‘Definition’ | ‘Definition’ | ‘Definition’ | ‘Definition’ |
| **Type** | ‘Charge’ | ‘Charge’ | ‘Charge’ | ‘Charge’ | ‘Charge’ |
| **Name** | ‘Heavy Subscription 2’ | ‘Regular Subscription 1’ | ‘Regular Subscription 2’ | ‘Light Subscription 1’ | ‘Light Subscription 2’ |
| **Start Date** | [Today’s date] | [Today’s date] | [Today’s date] | [Today’s date] | [Today’s date] |
| **End Date** | [One year from today] | [One year from today] | [One year from today] | [One year from today] | [One year from today] |
| **Charge Frequency** | ‘Monthly’ | ‘Monthly’ | ‘Monthly’ | ‘Monthly’ | ‘Monthly’ |
| **Currency** | ‘US Dollars’ | ‘US Dollars’ | ‘US Dollars’ | ‘US Dollars’ | ‘US Dollars’ |
| **Label** | ‘Heavy Subscription 2’ | ‘Regular Subscription 1’ | ‘Regular Subscription 2’ | ‘Light Subscription 1’ | ‘Light Subscription 2’ |
| **Amount** | ‘25’ | ‘20’ | ‘15’ | ‘12’ | ‘10’ |
| **Project** | ‘Project: High Speed Internet’ | ‘Project: High Speed Internet’ | ‘Project: High Speed Internet’ | ‘Project: High Speed Internet’ | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ | ‘Operations’ | ‘Operations’ | ‘Operations’ | ‘Operations’ |

After this work is done, you should see all seven charge types listed in the **Result** window:

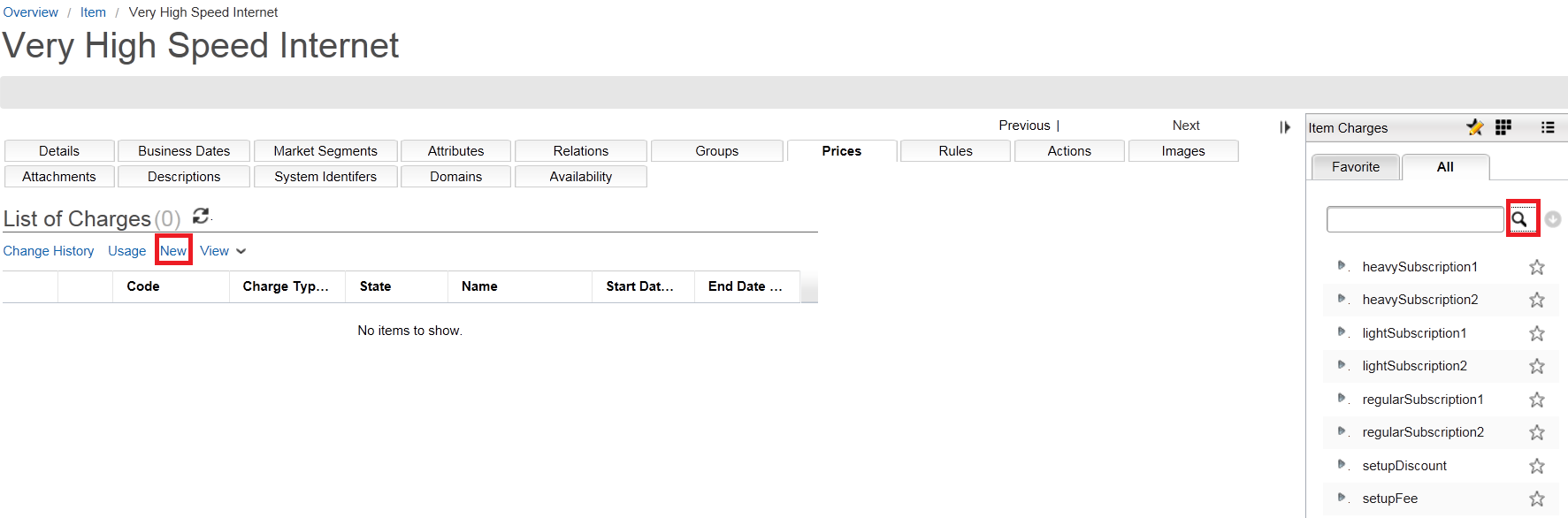


### Associate charge types with items

Having created the charge types in the previous section, we now associate them with the appropriate items, in order to be able to provide a pricing structure for our service model.

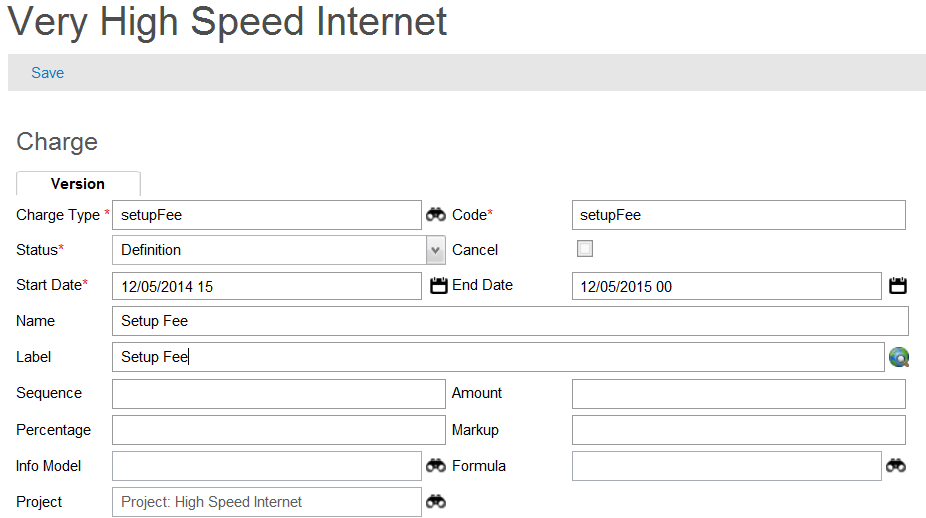
1. Go to *Product > Item in Quick Start Menu*.
2. Click on Search
3. Locate the **Very High Speed Internet** product item, and click the ‘**>**’ sign to the left of the item to show this item’s details.
4. Go to the **Prices** tab to add a new charge.
5. Then click on the **finder** icon next to the **Item Charges** window on the right.

(You should see all seven charge types we created)



1. Select and drag the **Setup Fee** charge type over under List of Charges. As we saw with other objects, as well as populating the **Charge Type** field, the system also updates the **Code** and **Name** fields, - inherited from the selected charge type.
2. Amend the **Name** value to ‘**Setup Fee**’, then ensure that all other fields are completed, as shown in the table below:

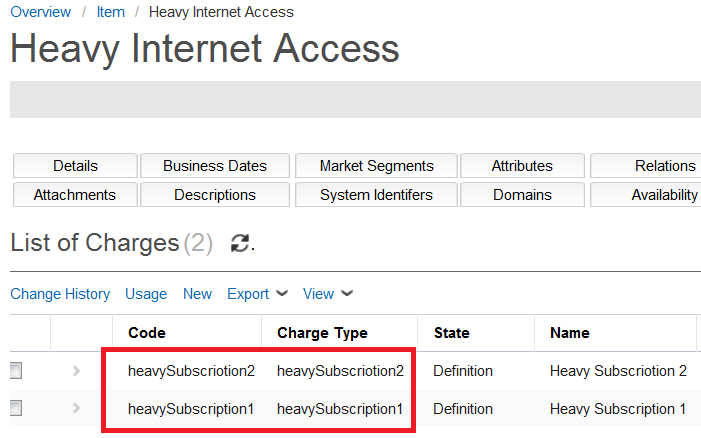
|  |  |
| --- | --- |
| **Status** | ‘Definition’ |
| **Cancel** | N/A (leave unchecked) |
| **Start Date** | [Today’s date] |
| **End Date** | [One year from today] |
| **Label** | ‘Setup Fee’ |
| **Sequence** | N/A (leave blank) |
| **Project** | ‘Project: High Speed internet’ |



1. Click **Save** in the **Charge Detail** panel title bar, and check that the new charge is visible in the **List of Charge**s (under **Very High Speed internet Prices Tab**):



1. Repeat the process described above to also associate the three component items (**Heavy Internet Access**, **Regular Internet Access** and **Light Internet Access**) with their corresponding six charge types, making sure you select both ‘1’ and ‘2’ subscription items for each of the three component items.
2. Check that these six additional charges are visible in the **Item** tree under their appropriate items:



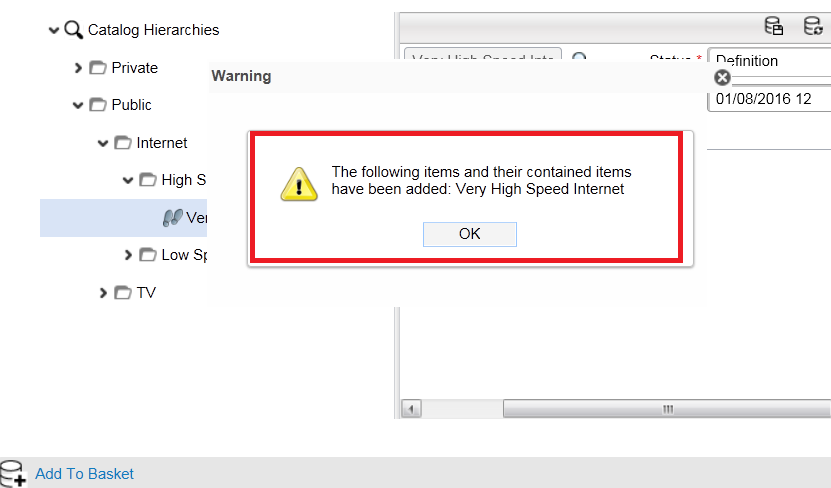
*Note: Remember that it is possible to have just one charge or multiple charges associated with a particular item. In this exercise we have explored both these scenarios: the product* *item* ***Very High Speed Internet*** *has just one charge associated with it, but the three component items (****Heavy internet Access****,* ***Regular internet Access*** *and* ***Light internet Access****) have two charges associated with each of them.*

Later in this document will explore how these charges can be applied ‘conditionally’, so that the charge does not apply all the time, - only when a certain condition is met.

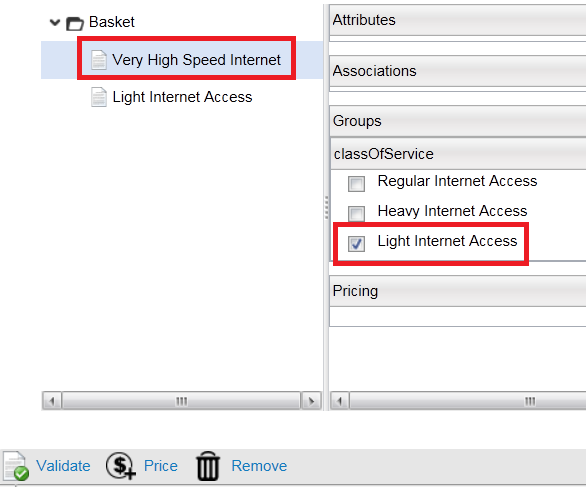
### Test basket pricing

In the final section of this exercise we go back into ‘**test**’ mode to browse the catalog, put a service item in the basket, and check that appropriate pricing is generated.

1. Go to *Testing > Test On*.
2. Go to *Testing > Browse.*
3. Drill down in the **Catalog Hierarchy** on the left, select **Very High Speed Internet** (*Public > Internet > High Speed > Very High Speed Internet*), then click **Add To Basket** at the bottom of the screen.
4. Click **OK** to acknowledge the confirmation message.

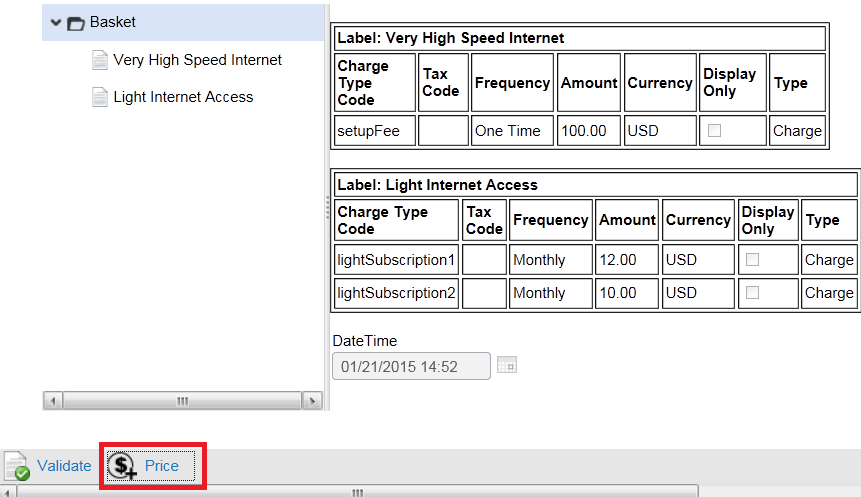


1. Go to  *Testing > Basket*.
2. Click on **Very High Speed Internet** under the **Basket** folder on the left, and then select **Light Internet Access** class of service using the same testing process as we did in an earlier exercise. This service item is then added to the basket list in the left panel:



1. After the **Light Internet Access** item has been added, click back into the **Basket** folder in the left panel (top level). We will now ask for the entire basket to be priced.
2. Click **Price** at the bottom of the screen (see screenshot below). You should see a list of all the charges that are associated with the current basket:

* One-time **Setup Fee**: ($100)
* Two monthly **Light Subscription** charges ($12 and $10)

**

1. Test the other service offerings in the basket, making sure that all expected and relevant charges are displayed. (Remember that you can only select one service at a time!).
2. Turn off the testing mode: *Test Mode > Turn Off*, and the Catalog Designer screen will go back to the Overview screen as it did before, awaiting the next user action.

Module 8: Context Attributes and Rules

In this Module we focus on **contexts**, **context attributes** and **catalog rules**.

Context attributes are basically ‘variables’ which hold information about the current context, things like whether the current customer is a *residential* or *commercial* type of customer, where the customer is located, how long he/she has been with the company, etc.

The attributes represent information obtained from the **Order Negotiation** application, and not from the **Catalog Manager**. The catalog needs this kind of information to be able to execute rules and make decisions which relate to the attribute details, and must therefore be able to communicate with the **Order Negotiation** application. When the catalog gets this information from Order Negotiation, it needs to store it somewhere, and this is the purpose of **context attributes**, which are *defined* in the **Catalog Designer** GUI, but *populated* by the **Order Negotiation** application.

Context attributes do not exist on their own, but in *groups* called ‘**contexts**’. Every **group** of context attributes therefore belongs to a specific **context**. Multiple **contexts** can be defined, each one containing multiple **context attributes**. Only one context can be ‘active’ or ‘current’ at any given time, however, and only the context attributes belonging to the *current context* can be used in **catalog rules**.

A **catalog rule** is basically a script with some built-in logic that is used to determine a ‘condition’. This script will be ‘**Catalog Rules Language**’ by default; otherwise JavaScript can be used for earlier versions of the software.

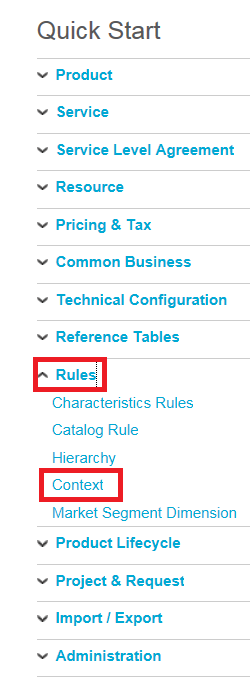
## Exercise 9: Context, context attributes and rules

In this exercise we create a context called ‘**Context 1**’, and add to it a context attribute called ‘**Market Segment**’. Then we create a new rule called ‘**Is Residential**’, using the Market Segment attribute in its logic.

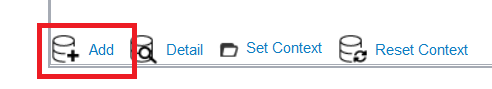
### Create new context and context attributes

First, then, we will create the new context and one context attributes (‘**Market Segment**’), which will indicate whether or not the customer is *residential* or *commercial*.

1. Check that your project is selected and open.
2. Go to *Rules > Context* under the Quick Start Menu:

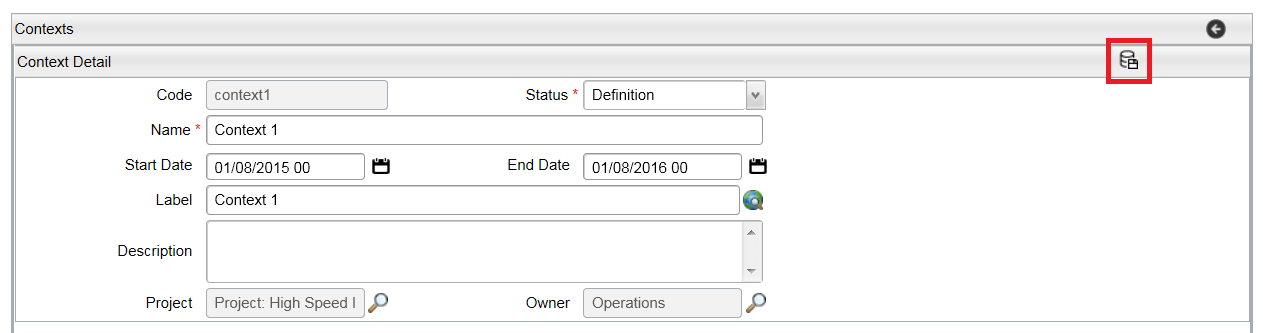


1. Click **Add** at the bottom of the screen. (You may need to scroll down to find this button).



1. Fill out the fields in the resulting **Context Detail** panel, as shown in the table below:

|  |  |
| --- | --- |
| **Code** | ‘context1’ |
| **Status** | ‘Definition’ |
| **Name** | Context 1 |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Label** | Context 1 |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |

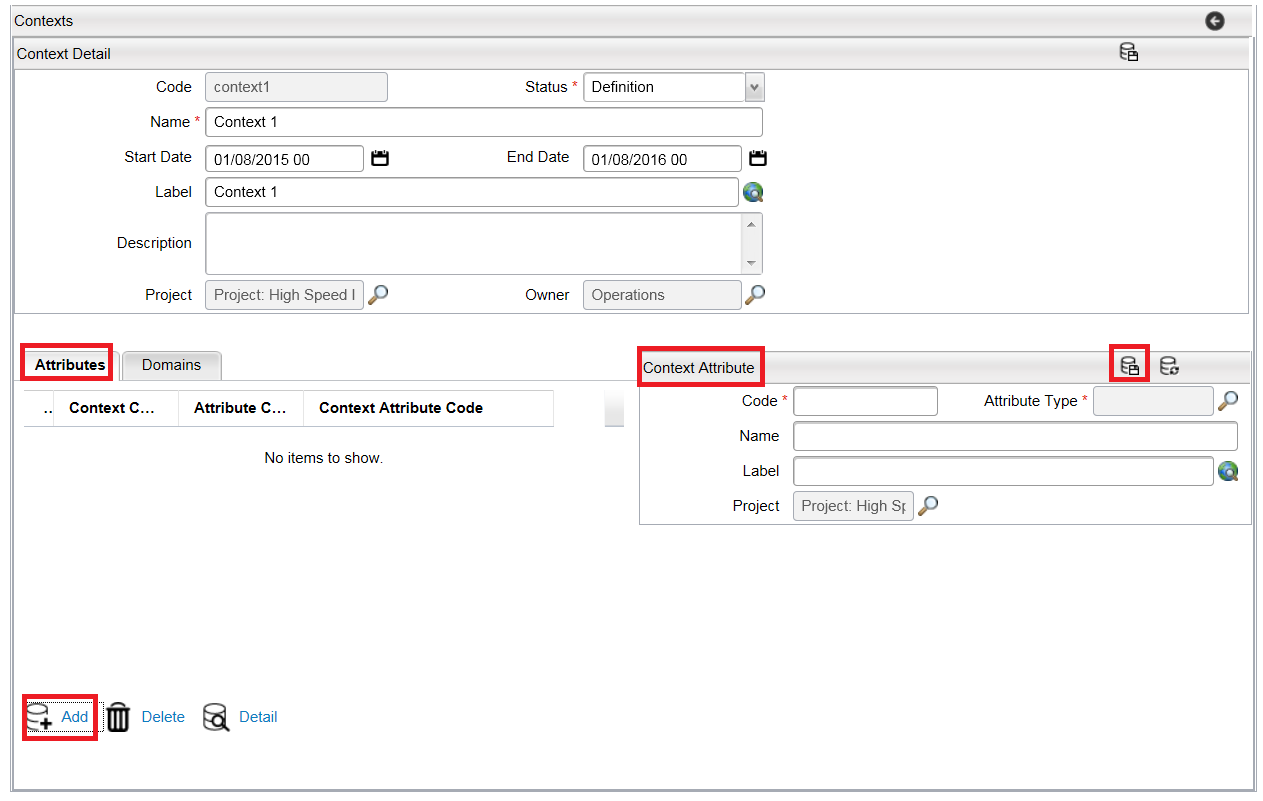


1. Save the new context, using the icon in the **Context Detail** panel title bar (see screenshot above).

Once the new context is saved, we can start adding attributes. (We will add just one attribute in this exercise).

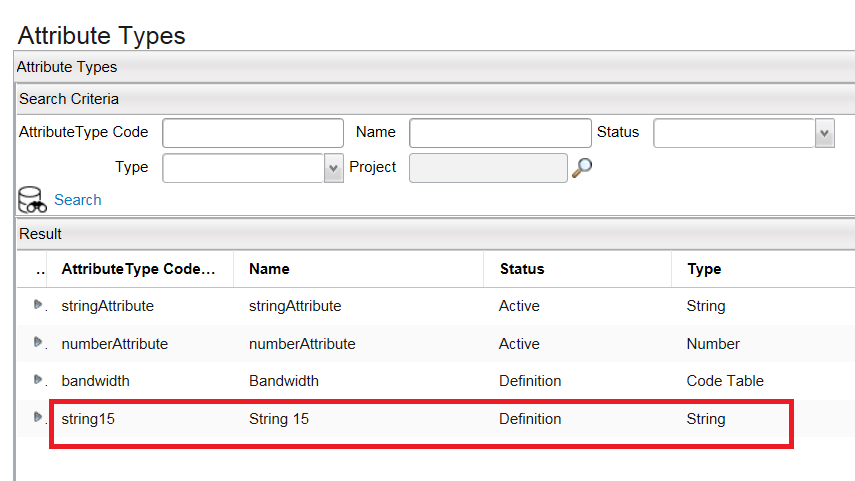
*Note: Once the* ***Save*** *button has been clicked, two tabbed panels appear underneath: ‘****Attributes****’ and ‘****Domains****’ (see screenshot below). We will now add a new context attribute to the context just created in the* ***Attributes*** *tab. (‘****Domains****’ were briefly discussed earlier in this document, and we will not be addressing this topic any further here).*

1. Make sure the **Attributes** tab is selected, and then click on the **Add** button at the bottom. This results in a ‘**Context Attribute**’ box appearing to the right of the **Attributes** panel:



As we said earlier, a **context attribute** is like a *variable*, and needs a ‘data type’ to be associated with it.

1. Click on the **finder** icon next to the **Attribute Type** field (see screenshot above).
2. In the resulting **Attribute Types** panel, click **Search**, and you will see a list of data types, including the two we created previously: ‘**String 15**’and‘**Bandwidth**’.
3. Select the **String 15** data type by clicking on the arrow before it:



As before, the **Code** and **Name** values will be inherited from the selected data type, and will automatically be populated in the appropriate fields.

1. The name ‘**String 15**’is too generic for our purposes here, so change the **Code** value to ‘**marketSegment**’, and the **Name** and **Label** values to ‘**Market Segment**’. The **Project** field should have been automatically populated as usual.
2. Click **Save**, using the icon in the **Context Attribute** panel title bar. We now have a **context** called ‘**Context 1**’ and a **context attribute** called ‘**Market Segment**’:



### Create rule and associate with context attribute

Now we will create a ‘**catalog rule**’ that will use the context attribute in its logic. This rule will be called ‘**Is Residential**’, and will return ‘true’ if the current customer is of *residential* type, otherwise it will return ‘false’. It will be used to create a condition for a specified discount to be applied in pricing, thereby making it available only to residential customers.

*Note: This discount will be created and applied to the rule in a later exercise. For now we will focus on creating the rule and associating it with the context attribute.*

1. Go to *Rules > Catalog Rule in the Quick Start Menu*:

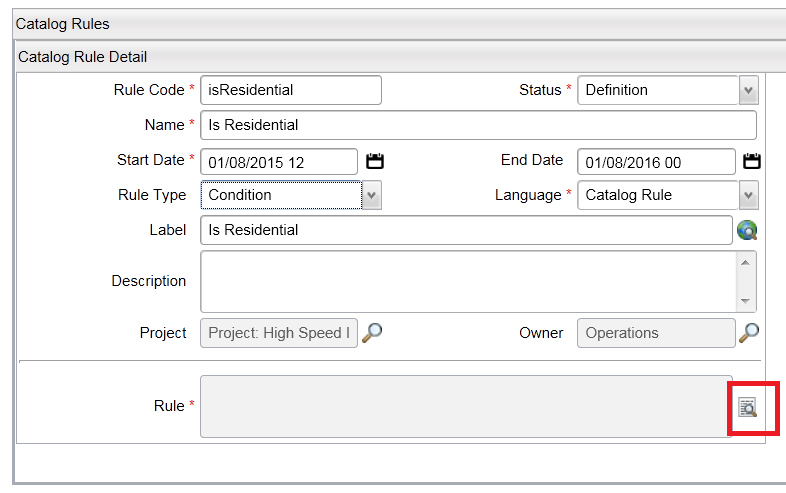


1. Click **Add** at the bottom of the screen, and fill out the fields as shown in the table below:

|  |  |
| --- | --- |
| **Rule Code** | ‘isResidential’ |
| **Status** | ‘Definition’ |
| **Name** | ‘Is Residential’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Rule Type** | ‘Condition’  *When creating a rule you have three* ***Rule Type*** *options:* ***Validation****,* ***Price*** *and* ***Condition****. When associating the rule with an element you have four* ***Rule Type*** *options:* ***Availability****,* ***Eligibility****,* ***Ranking*** *and* ***Pricing****. The type of the association determines when the rule will be used. For example,* ***Availability*** *rules take effect during catalog browsing, whilst* ***Eligibility*** *rules take effect when items are added to the basket. (See* ***Help*** *documentation for further details)* |
| **Language** | ‘Catalog Rule’  *Means that the new catalog rule language should be used. The other option ‘4.2 Script’ is for backward compatibility with 4.x catalogs.* |
| **Label** | ‘Is Residential’ |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |

There remains one important field to complete: ‘**Rule**’ (a mandatory field).

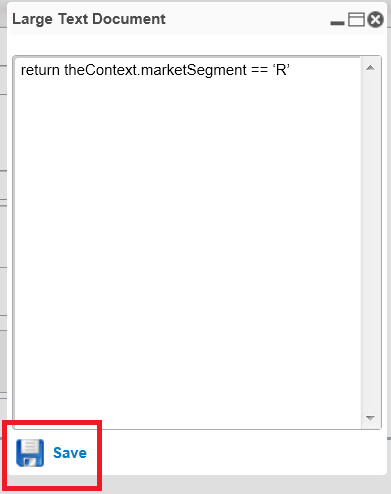
1. Click on the **finder** icon next to the **Rule** box:



You will now see a ‘**Large Text Document**’ window open.

1. Type in the following single statement to create the rule logic:

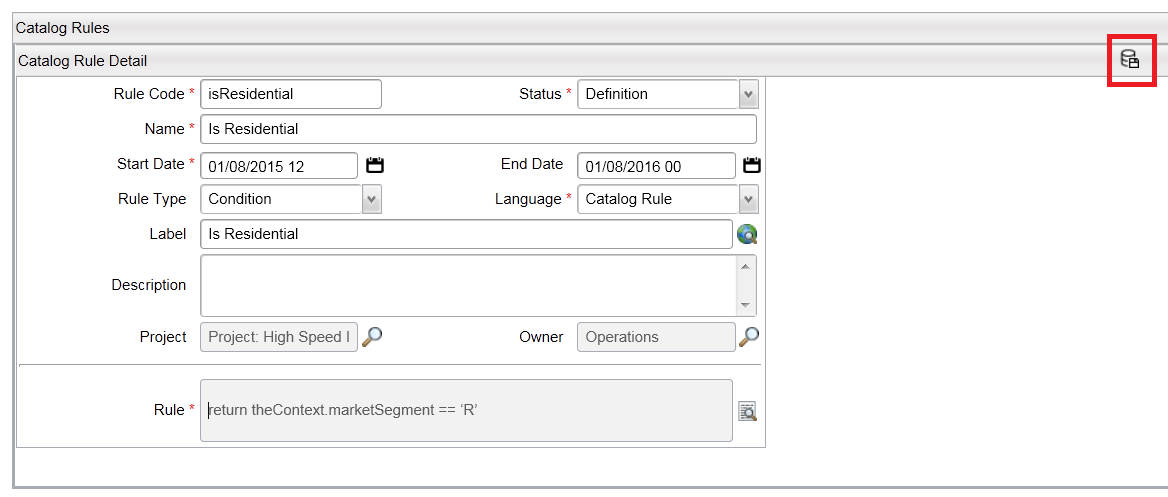
*return theContext.marketSegment == ‘R’*

**

This logic will return the following condition: “…is the **Market Segment** context attribute equal to ‘R’ (i.e. residential)?” If ‘yes’, then any configured condition can be applied *automatically*, such as a setup fee discount. (This scenario is explored in the *extension* exercise at the end of this document).

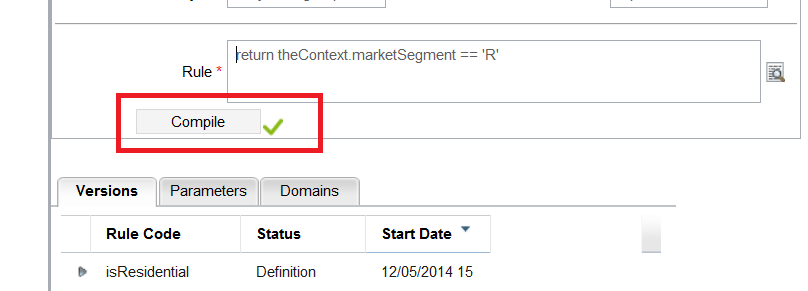
‘**theContext**’ in this rule is a system-defined variable that references the current context, whatever the name of the context is.

1. Click **Save** at the bottom of the **Large Text Document** window (see screenshot above) to save the rule code and return to the **Catalog Rule Detail** window.
2. Now click the **Save** button in the **Catalog Rule Detail** panel title bar, to update the rule object.

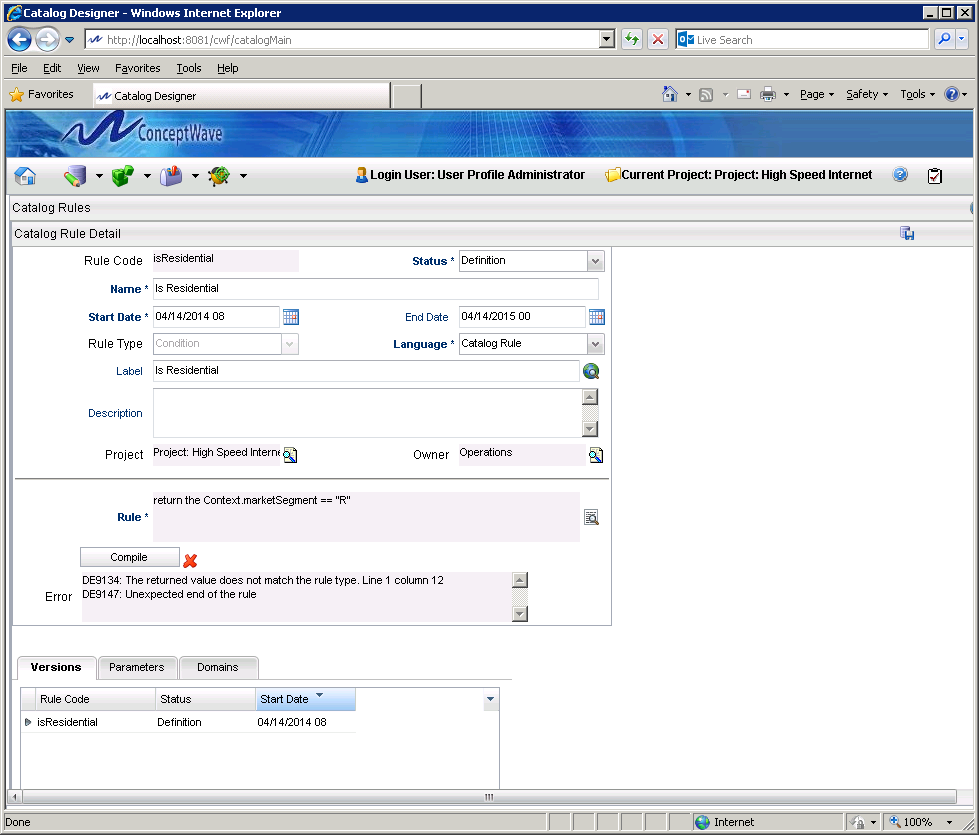


After the *second* save step, apart from the rule now being listed in the ‘**Versions**’ box which was generated, you will also see that a ‘**Compile**’ button has appeared under the rule definition (see screenshot below). This feature will be important in checking that the rule code is syntactically correct.

1. Click the **Compile** button to see if the rule’s syntax is correct. If good, a green tick will appear next to the Compile button:



If there is an error in the code, however, you will get a red cross and an error message:

**

You will then need to re-visit and correct the code logic. This is done by clicking on the **finder** icon while the required version is selected in the **Versions** tab underneath, resulting in the re-appearance of the **Large Text Document** window.

The new rule is now ready to be applied as required in the system.

*Note: If you want to use a context attribute in a catalog rule, you must first ensure that the context to which the context attribute belongs is set as the ‘current’ context. This is done by searching for and selecting the required context, then using the ‘****Set Context****’ option at the bottom of the screen. After this the selected context attribute needs to be set using the ‘****Set Context Attributes****’ option in the* ***Testing*** *pull-down menu.*

Module 9: Conditional Charges

Having learnt how to create **charge types** and **rules** in previous exercises, we now take a look at how we can apply charges ‘*conditionally*’.

Conditional charges are created by associating the charge type with a rule that returns ‘true’ or ‘false’. If the rule returns ‘true’, then the charge is applied, but if the rule returns ‘false’, then the charge is not applied.

## Exercise 10: Conditional charges and testing

Earlier we discussed how the amount of a charge can be specified, based on the logic included in a rule. In this exercise we will now create and apply a rule which will determine if the charge will be applied or not. We will then test the rule logic, and make sure that the charge is only applied as appropriate to the business case.

### Create new rule and associate with charge type

In this part of the exercise we create a new rule called ‘**Not Heavy Subscription**’. This rule will return ‘true’ if the user has selected any of the **Regular** or **Light** subscription options, and will return ‘false’ if the user has selected either of the **Heavy** subscription options. This logic is achieved by associating the rule with the ‘**Setup Fee**’ charge type created earlier (i.e. applying the condition).

*Note: For the customer, the impact of this ‘condition’ will be that if he/she opts for a* ***Heavy Internet Access*** *service, then the setup fee of $100 will be waived. This charge will therefore only be applied (conditionally) to* ***Regular Internet Access*** *or* ***Light Internet Access*** *subscribers.*

The process of creating a new rule in Catalog Manager (Catalog Designer GUI) was detailed in the previous exercise, where a rule called ‘**Is Residential**’ was created and associated with the ‘**Market Segment**’ context attribute, which itself was created and associated with the new ‘**Context 1**’ context in the same exercise:



1. Using your knowledge gained from Exercise 9, create now a new rule called ‘**Not Heavy Subscription**’, entering field values as shown in the table below:

|  |  |
| --- | --- |
| **Rule Code** | ‘notHeavySubscription’ |
| **Status** | ‘Definition’ |
| **Name** | ‘Not Heavy Subscription’ |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Rule Type** | ‘Condition’ |
| **Language** | ‘Catalog Rule’ |
| **Label** | ‘Not Heavy Subscription’ |
| **Description** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |
| **Owner** | ‘Operations’ |

1. At the appropriate point, and where required, use the following code for the *rule logic*:

*if exist associated item in theItem where code is ‘heavyInternetAccess’*

*return false*

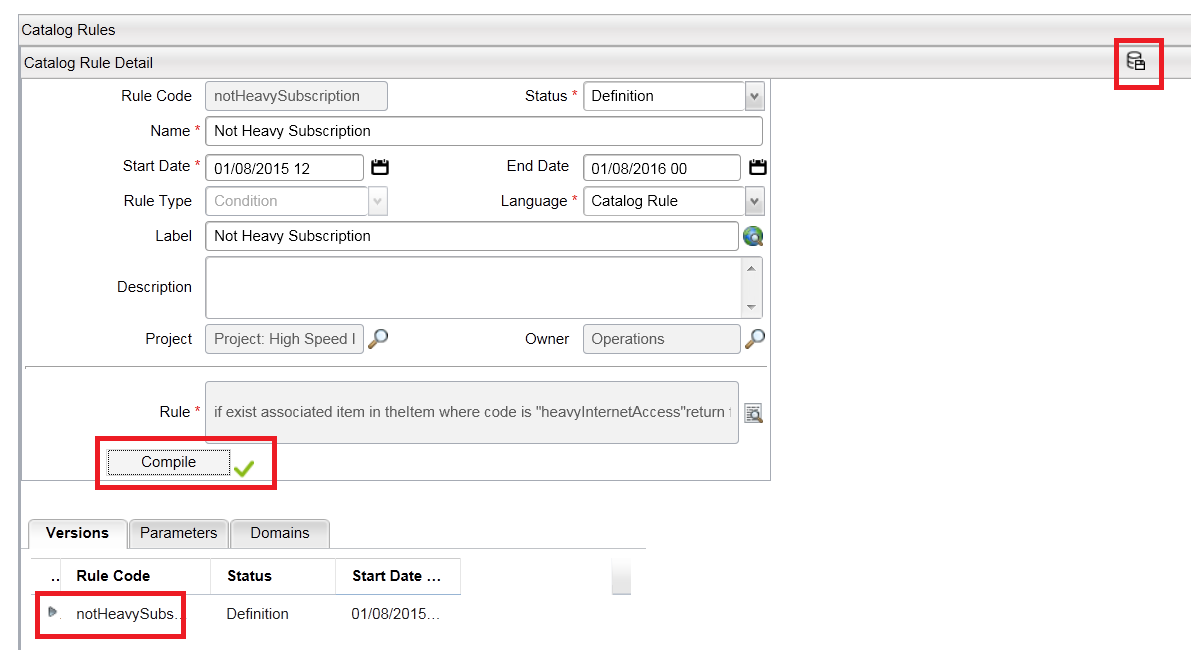
*else*

*return true*

*end if*

(Don’t forget to test your code with the ‘**compile**’ function, and **save** it of course).

If all goes well you will see your new rule in the **Versions** tab in the **Catalog Rule Detail** panel:

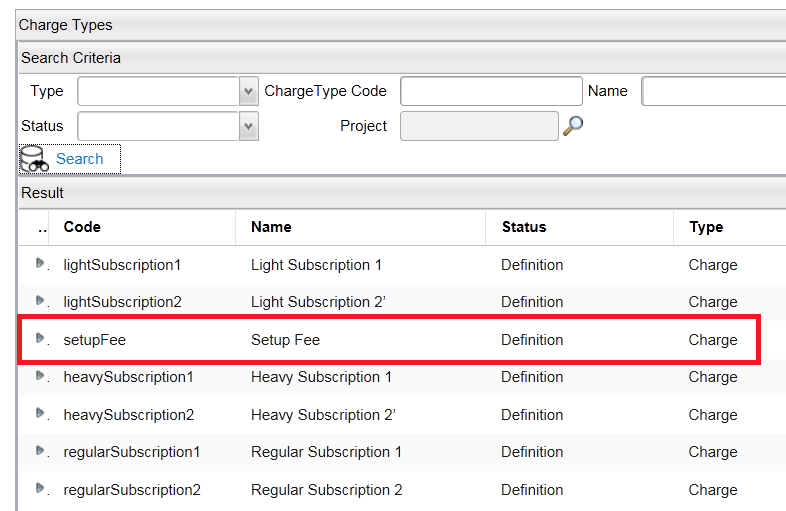


What the code means

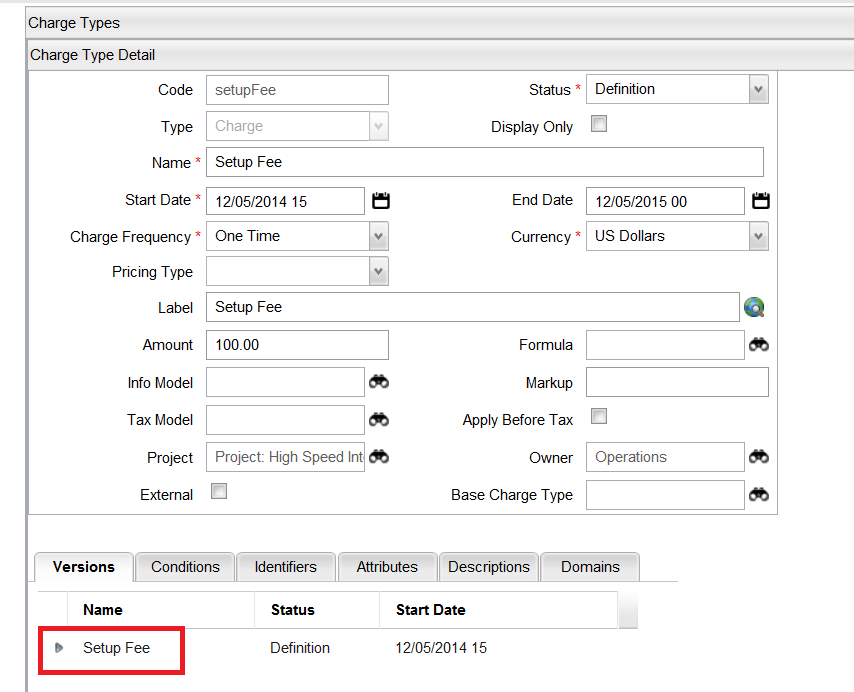
The code used in this rule checks on the name of the item selected by the user (‘*if exist associated item…’*). ‘*theItem*’ is a key word which refers to the item being priced at this point (whatever that item actually may be). This means that if the item being referenced is associated with an item that has the code ‘*heavyInternetAccess*’, then the rule returns ‘false’, which in turn means that the $100 charge is not applied. If the rule returns ‘true’, however, then the charge is applied.

Remember that on this occasion the rule is going to be associated with the ‘**Setup Fee**’ charge type, so that the charge can be conditionally applied based on the return value from the rule. We will do this now.

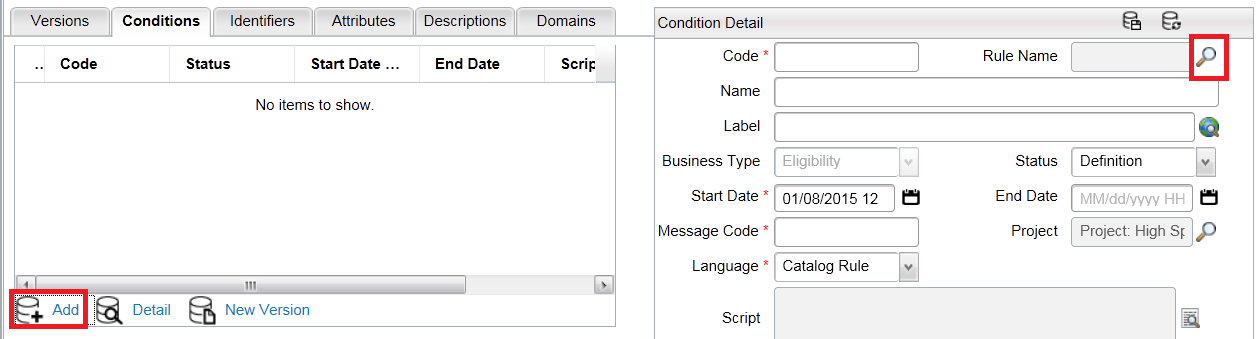
1. Go to *Pricing & Tax > Charge Types* in the Quick Start Menu.
2. Click **Search** to locate and open the **Setup Fee** charge type we created earlier in Exercise 8:



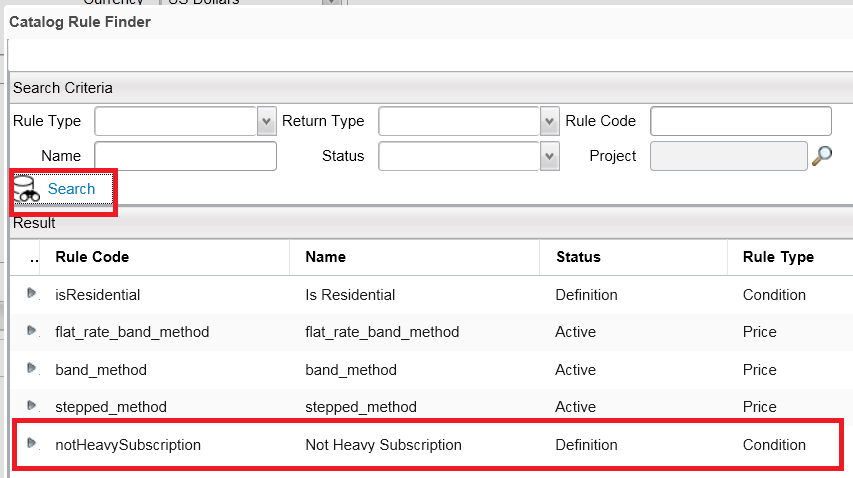
You should now see the details of the Setup Fee Charge type with the Setup Fee also appearing in the Versions tab in the lower pane:



1. Select the **Conditions** tab, and then click **Add** at the bottom. As a result of this action you will see a **Condition Detail** panel appear on the right:

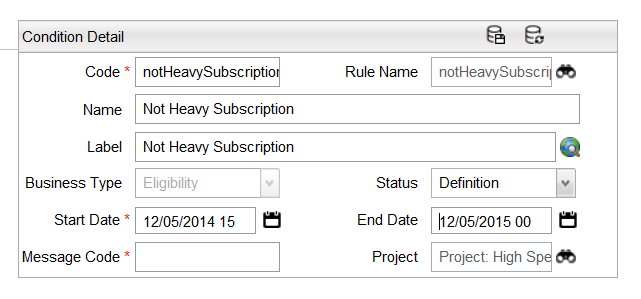


1. Click on the **finder** icon next to the **Rule Name** field (see screenshot above), then search for the ‘**notHeavySubscription**’ rule code:

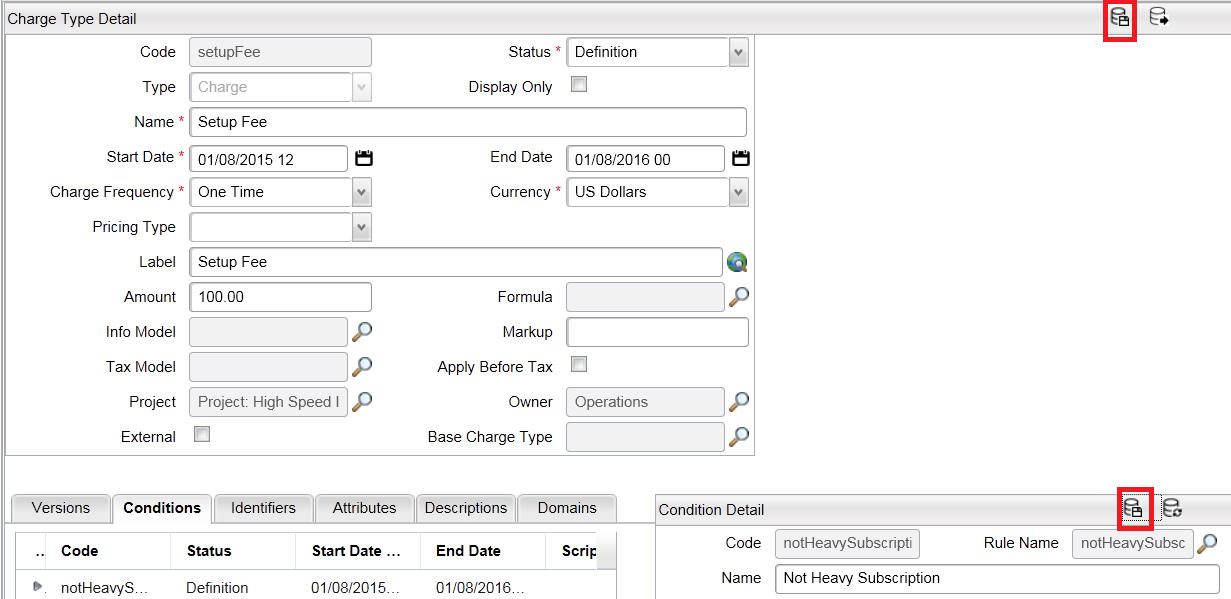


1. Select and enter this rule code by double-clicking the row (or by clicking the small blue arrow at the start of the row). As before, you will see that the **Code** and **Name** fields inherit their values from the rule name, though you will want to edit the **Name** field of course.
2. Complete the remaining fields values as shown in the table below:

|  |  |
| --- | --- |
| **Name** | ‘Not Heavy Subscription’ |
| **Label** | ‘Not Heavy Subscription’ |
| **Business Type** | ‘Eligibility’ (can’t be changed) |
| **Status** | Definition |
| **Start Date** | [Today] |
| **End Date** | [One year from today] |
| **Message Code** | N/A (leave blank) |
| **Project** | ‘Project: High Speed Internet’ |



1. Click **Save** using the icon in the **Condition Detail** panel title bar, which results in the newly-created condition being visible under the **Conditions** tab:



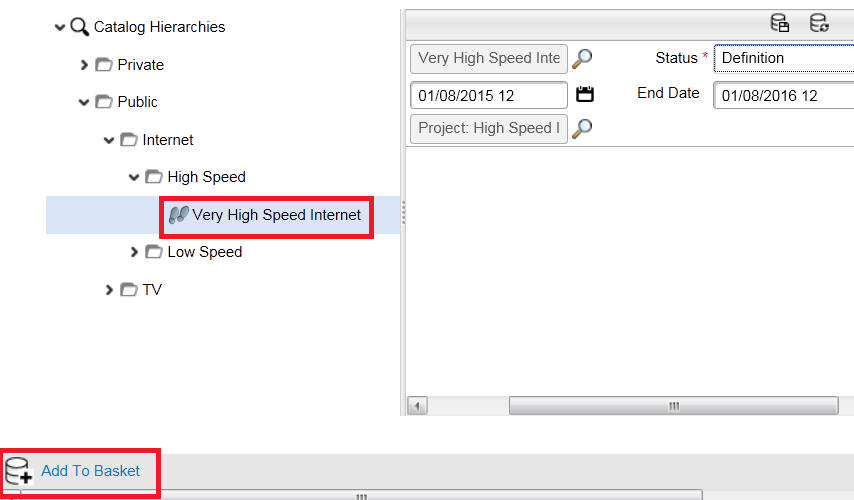
1. Finally click **Save** in the **Charge Type Detail** panel title bar to commit the updated charge type to the database (see screenshot above).

The **Setup Fee** charge will now be conditionally applied based on the component that the user selects at runtime when he/she places an order that includes the **Very High Speed Internet** product.

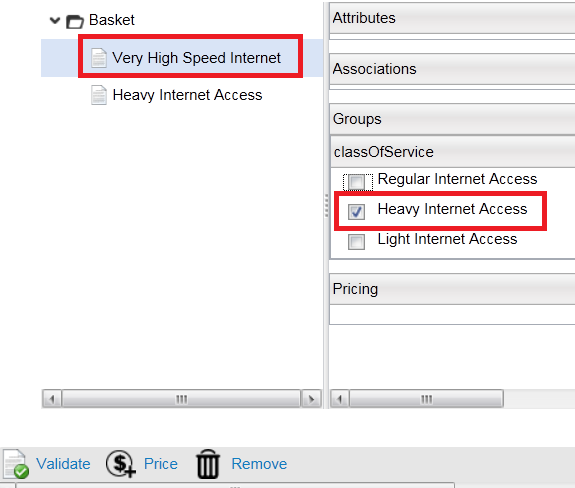
### Test rule logic

We will now use the testing functionality in Catalog Designer to test that the **Setup Fee** charge will be appropriately applied to newly-requested services, according to the rule logic implemented earlier.

1. Go to *Test Mode > Test On* then *Test Mode > Browse*.
2. Browse the hierarchy in the left panel to locate and select the **Very High Speed Internet** item (*Public > Internet > High Speed > Very High Speed Internet*):

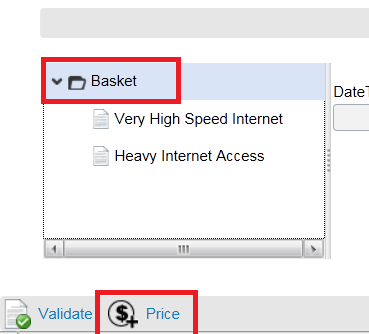


1. Click **Add To Basket** at the bottom of the screen, and click **OK** in the resulting confirm message dialog box.
2. Go to *Browse Basket*, then locate and select the **Very High Speed Internet** item in the left panel.
3. Select the **Heavy Internet Access** service optionin the **Class of Service** panel.

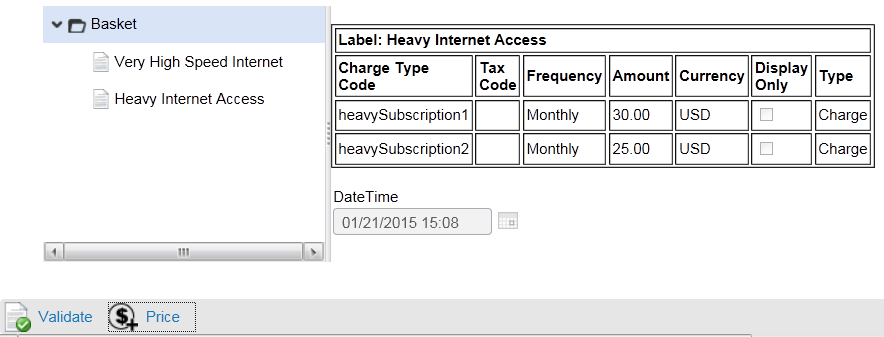


We will now test the pricing of the basket contents, which contains the selected service.

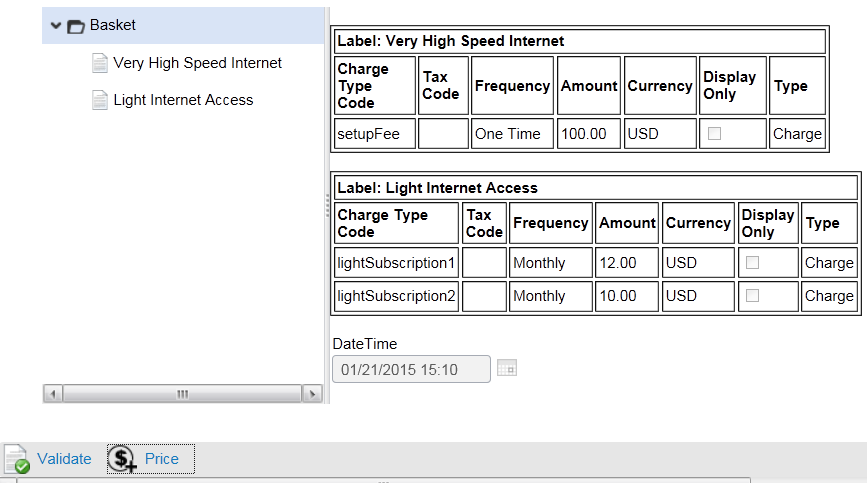
1. Click on **Basket** at the top of the left-hand panel and then click **Price** at the bottom of the screen.



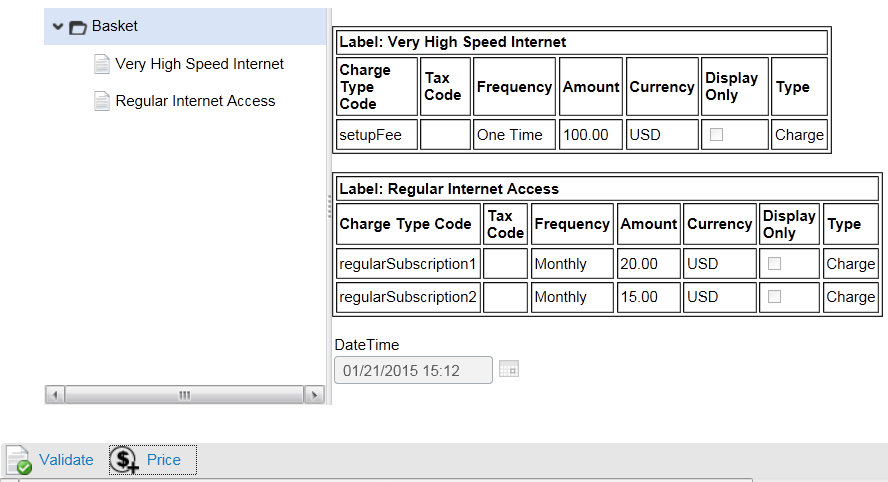
As a result of these actions you should see the appropriate price information displayed (see screenshot below), giving the two **Heavy Subscription** pricing options ($30 and $25), but not the **Setup Fee** charge ($100), which we configured earlier to be applied only in the cases of the **Regular** and **Light** subscription options.



1. Click back onto the **Very High Speed Internet** option on the left, then de-select **Heavy Internet Access** in the **Class of Service** panel, clicking **OK** in the resulting confirmation dialog box.
2. Now select the **Light Internet Access** instead and test the basket once more for pricing. This time you should see that, along with the **Light Subscription** prices ($12 and $10), the **Setup Fee** charge of $100 is also applied:



To complete your testing procedure, test the pricing of the basket when the **Regular Subscription** option is chosen. Once again the setup fee charge should be applied, along with the subscription charge options of $20 and $15.



1. Turn off the testing mode: *Test Mode > Turn Off*, and the Catalog Designer screen will go blank as it did before, awaiting the next user action

## Extension exercise 11 (OPTIONAL): Introduce setup discount for residential customers

In this optional extension exercise we ask you to perform some actions in the Catalog Designer without the detailed support you have been provided within all the previous exercises.

To ensure you are working along the right track, we have provided you here with the declared objective, and a high-level list of actions required to achieve that objective. The rest is up to you, using the knowledge you have gained during this course and your general understanding of how things work in the system. If you really get stuck you can ask the instructor, whose advice will be free, though your pride may be dented slightly…

In Exercise 9 you created a new context called ‘**Context 1**’ and a context attribute called ‘**Market Segment**, the purpose of which was to define whether or not the current customer was *residential* or *commercial* when **Context 1** was set as the ‘default context’. The logic for this scenario was embedded in a related rule called ‘**Is Residential**’.

Your objective here is to configure the system so that when services are added to the basket, a setup fee discount of 20% is given to customers who are *residential*, at the same time ensuring that *commercial* customers pay the full amount ($100).

To help you with this task, here is a high-level list of actions that you need to take:

1. Create a new **Charge Type** called ‘**Setup Discount**’, offering a 20% discount on the setup fee for *residential* customers. (*Hint: you will need a percentage option somewhere in the field values*).
2. Add the **Setup Discount** charge to the **Very High Speed Internet** item.
3. Create a **Condition** which associates the ‘**Is Residential**’ rule created in Exercise 9 with this new charge type.
4. Set the context as ‘**Context 1**’, which was created in Exercise 8. The way to do this is described at the very end of Exercise 9. Then set the **Market Segment** context attribute to ‘**R**’.
5. **Test** that the pricing of the basket shows 20% discount on the setup fee after selecting one of the **Very High Speed Internet** services.

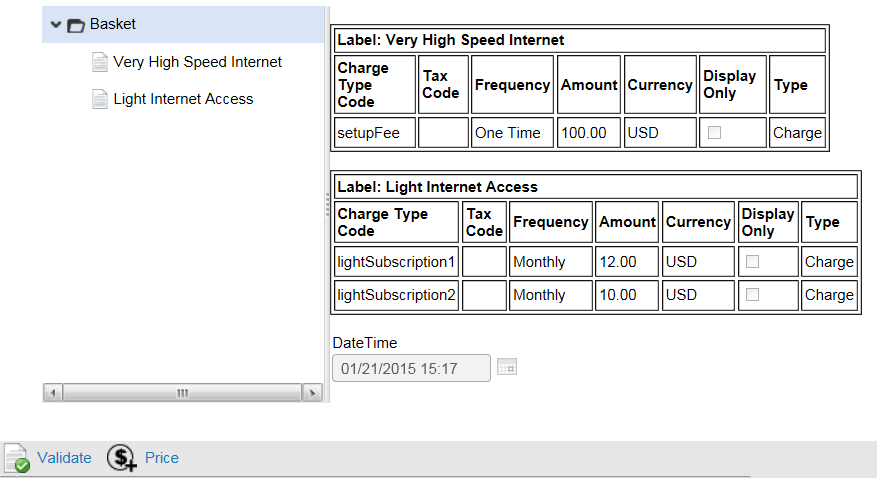
If you attempt this exercise……good luck! ☺

## Extension exercise 12 (OPTIONAL): Resolve subscription charge issue

This is another optional extension exercise, where we ask you to perform some actions in the Catalog Designer, but this time with little or no support. Just how good is your knowledge of Catalog Manager after this training…..?

As before, we provide a short introduction to the exercise scenario, and a review of the end objective.

As a result of doing the exercises laid out in this document, you will have seen that when testing that the **Setup Fee** is only applied to the **Regular Internet Access** and **Light Internet Access** services, *both* subscription options were generated in the basket for each of the three service selections:



Actually the end objective is to ensure that only *one* subscription option is generated when the user selects the required bandwidth from the pull-down lists for the required subscription (see Exercise 7).

To achieve this you must make the six different subscription charge types you created in Exercise 8 *conditional* on the specific bandwidth required. You must therefore modify the charge types by adding a rule for each one along the lines of ‘*if….else….*’.

Don’t forget to test the new conditions via the ‘*Browse’* and ‘*Basket*’ **Test Mode** options. This exercise provides no other help, and presents the biggest challenge for you on this course.

If you attempt this exercise……good luck! ☺

THAT COMPLETES THE EXERCISES FOR **ERICSSON CATALOG MANAGER**, AND IN PARTICULAR FOR THE **CATALOG DESIGNER** APPLICATION.

