

# **Peer Test Report**

PMoC  
3/24/2003  
Version 1.1

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Test Results</b>	<b>3</b>
2.1 PTC_1: Drawing free symbols	3
2.2 PTC_2: Moving graphical objects	3
2.3 PTC_3: Make free symbol a terminal symbol	3
2.4 PTC_4: Add terminals to symbol	4
2.5 PTC_5: Attach symbol to type	4
2.6 PTC_6: Create new category	4
2.7 PTC_7: Open net	4
2.8 PTC_8: Add components to net	5
2.9 PTC_9: Move components and graphics in a net	5
2.10 PTC_10: Delete component	5
<b>3. Evaluation and feedback</b>	<b>5</b>

## 1. Introduction

Summarize the contents of this report and state any deviations from this template or the test plan.

## 2. Test Results

### 2.1 PTC\_1: Drawing free symbols

**Description:**

Create a new symbol from the tree view. Use the drawing tools to draw graphics to the canvas of the free symbol. Use the delete tool to delete different graphical objects. Also try to save the symbol. The test passes if all drawing tools produce graphics.

**Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

### 2.2 PTC\_2: Moving graphical objects

**Description:**

Use the move tool and the commands in the Image menu to move graphical primitive objects. Group objects together, put them on top of each other and clone them. The test passes if all object can be moved.

**Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

### 2.3 PTC\_3: Make free symbol a terminal symbol

**Description:**

Make a free symbol a terminal symbol. The name of the canvas should change from “New symbol” to “Terminal”. Save the terminal. The test passes if the name changes correctly and the terminal can be saved.

**Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.4 PTC\_4: Add terminals to symbol

### **Description:**

Draw a symbol and add terminals to it. After adding a terminal it should be movable. The test passes if terminals can be added.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.5 PTC\_5: Attach symbol to type

### **Description:**

Draw or load a free symbol with terminals. Browse the tree view to component types. Attach the symbol to a component type. The name of the canvas should change from “New symbol” to “Component”. Save to component. The test passes if the name changes correctly and the component can be saved.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.6 PTC\_6: Create new category

### **Description:**

Make a new category to a department or another category in the tree. A new node should be created under the parent node. The test passes if the new category is visible in the tree after the creation.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.7 PTC\_7: Open net

### **Description:**

Open the net of a category. A new canvas should appear to the right named “net”. The test passes if the net opens.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.8 PTC\_8: Add components to net

### **Description:**

Add a new category and open its net. Instantiate component types to this net with the drag and drop technique. Tests 5,6 and 7 must have passed, partially at least, in order to continue with this test. The test passes if the graphics of the component appears on the net and the component appears in the tree below the category that the net represents.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.9 PTC\_9: Move components and graphics in a net

### **Description:**

After adding components to a net, try to move them around. Use the graphics tools to add texts and other graphics and move them around on the net. The test passes if components can be moved.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 2.10 PTC\_10: Delete component

### **Description:**

Delete a component, both starting from the tree and from the net. When deleting a component it should disappear from both the canvas and the tree. The test passes if deletion removes the component from both the tree and the canvas.

### **Test Results:**

<Test results should indicate whether a test passed or not, when the test was run and by who. In case of a failed test a description of each fault or abnormality, along with the corresponding Burana code should be stated.>

## 3. Evaluation and feedback

Evaluate the peer test plan, the user guide, the installation of the system and the defectiveness of the system.

Give feedback and improvement suggestions to the PMoC group

