

**EXAM BLUEPRINT**

# **PRPC Certified System Architect v6.1**



**BUILD FOR  
CHANGE.**

**© Copyright 2011**  
**Pegasystems Inc., Cambridge, MA**

All rights reserved.

This document describes products and services of Pegasystems Inc. It may contain trade secrets and proprietary information. The document and product are protected by copyright and distributed under licenses restricting their use, copying, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This document is current as of the date of publication only. Changes in the document may be made from time to time at the discretion of Pegasystems. This document remains the property of Pegasystems and must be returned to it upon request. This document does not imply any commitment to offer or deliver the products or services provided.

This document may include references to Pegasystems product features that have not been licensed by your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems service consultant.

PegaRULES, Process Commander, SmartBPM<sup>®</sup> and the Pegasystems logo are trademarks or registered trademarks of Pegasystems Inc. All other product names, logos and symbols may be registered trademarks of their respective owners.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors. This document or Help System could contain technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Pegasystems Inc. may make improvements and/or changes in the information described herein at any time.

This document is the property of:

Pegasystems Inc.

101 Main Street

Cambridge, MA 02142-1590

Phone: (617) 374-9600

Fax: (617) 374-9620

[www.pegasystems.com](http://www.pegasystems.com)

PegaRULES Process Commander

Document: Certified System Architect Exam Blueprint

Updated: March 7, 2011



## Contents

About the Certified System Architect Exam .....	1
About the Exam Blueprint .....	1
Prerequisites .....	1
Professional Experience .....	2
Exam Test Domains.....	2
Exam Format .....	2
Question Format .....	3
Test Topics .....	3



## About the Certified System Architect Exam

The Pegasystems Certified Professional Program has created a worldwide community of hundreds of PRPC Certified System Architects, drawn by a commitment to excel in their ability to deliver world class PegaRULES Process Commander (PRPC) applications.

When a System Architect participates in the design and construction of a PRPC application there are fundamental, essential PRPC skills and knowledge areas that must be applied to ensure success. It is these skills and knowledge areas that form the basis of the certification exam.

Thus, Pegasystems is committed to providing you the training, tools, and knowledge that you need to achieve Certification as a PRPC System Architect.

## About the Exam Blueprint

The purpose of the blueprint is to provide you with a roadmap of the System Architect Certification content to allow you to better prepare for the exam.

The blueprint includes test domain weighting, test objectives, and topical content. The topics and concepts are included to clarify the test objectives.

The exam is based upon the knowledge areas necessary for a System Architect to be able to successfully design and build multi-process PRPC applications.

Candidates are tested on their:

- ▶ Understanding of the phases, the DCO tools used in the Pegasystems' Implementation Methodology
- ▶ Ability to analyze, design and create class and data structures that are built for reusability
- ▶ Ability to understand and/or apply key concepts and techniques in the design and construction of the components of a multi-process application
- ▶ Ability to understand and apply PRPC test and developer tools

## Prerequisites

The suggested training prerequisites for this certification level are the completion of the following Pegasystems instructor-led courses:

- ▶ PRPC: Fast Track
- ▶ PRPC: Bootcamp

The skills and knowledge areas measured by this exam are derived directly from the content of these courses.

## Professional Experience

Additionally, you may find the following profession experience and/or exposure to PRPC helpful in preparing for the exam:

- ▶ Experience in the development of applications using an object-oriented language
- ▶ Experience in the development of applications using PRPC
- ▶ Familiarity with PRPC Version 6.1 (or earlier versions)
- ▶ Programming experience with Object Oriented languages, javascript and relational databases
- ▶ Exposure to PRPC information via product documents, online help files and/or the Pegasystems Developer Network (PDN)

## Exam Test Domains

The table below lists the test domains and the extent to which they are represented as an estimated percentage of test items.

Test Domains	% of Exam
Direct Capture of Objectives	7%
Class structure and Hierarchy Design	10%
Process Flow	18%
Case Management	5%
Data Modeling	7%
User Interface	13%
Activities	5%
Decision Rules	6%
Declarative Rules	6%
Rule Management/Security	11%
Integration	5%
Performance	7%
<b>Total</b>	<b>100%</b>

## Exam Format

The exam consists of 70 multiple choice questions. You are given 90 minutes to complete the exam and the initial exam sign in form. A passing score of 70% is needed to be recognized as a PRPC Certified System Architect (CSA).

## Question Format

The examinee selects from one or more response options to answer a question. A response is considered correct when it accurately completes the statement or answers the question. Distracters or incorrect answers are plausible response options that examinees with incomplete knowledge are likely to choose.

Test item formats used in this examination are:

- ▶ **Multiple Choice** — Select one option that best answers the question or completes a statement.
- ▶ **Multiple Responses** — Select more than one option that best answers the question or completes a statement. The text states how many options are correct, such as Choose two.
- ▶ **Sample Directions** — Read the statement or question. From the response options, select the option(s) that represent the most correct or best answer(s) given the information provided.
- ▶ **True/False** — Read the statement or question. Select either true or false as the answer.

## Test Topics

### Direct Capture of Objectives (DCO)

- Phases of the Project Methodology
  - Name the project phases
  - Describe the goals and deliverables of each phase
- Directly Capturing Objectives
  - Describe the inputs to the process
  - Identify the steps followed by the Application Profile Wizard and Application Accelerator
  - Identify the generated documents
  - Identify what information needs to be captured

### Class Structures & Hierarchy Design

- Reusability and Hierarchy Design
  - Define reusability and its impact on application design
  - Describe the difference between a framework layer and an implementation layer
  - Design and create an organizational structure comprised of an organization, divisions, and business units
  - Understand the concept and benefits of building on a framework layer

- Design and implement a reusable class structure with appropriate top level classes, class groups and work classes
- Classes and Class Groups/Work Pools
  - Create classes and class groups
  - Understand the purpose of a class group and its relation to the application's database and a user's work pool
  - Describe the difference between abstract and concrete classes
  - Understand the difference between pattern and directed inheritance
- Units of Work
  - Understand the inheritance structure of work classes
  - Describe the importance of the Work- class and it's children
  - Explain how a work object ID is defined
  - Locate work objects on the clipboard

### Process Flow

- Visio Editing Tool
  - Explain how SmartShapes are used in a flow diagram
  - Create and modify flow diagrams
- Flow Design
  - Design, build and test a multi-process flow
  - Describe how a flow can support reusability
  - Call one flow from another flow using SmartShapes and activities
  - Design and implement a screen flow
  - Understand how to divide a large flow into several smaller flows
  - Understand the purpose of – and how to create – worklists and workbaskets
  - Describe the benefits of placing a flow into Draft mode
  - Use the Where-Am-I feature during flow processing
- Flow Actions
  - Define and create flow action rules
  - Describe various methods used to define HTML in a flow action
  - Call activities before and after a flow action executes
  - Understand the difference between connector flow actions and local flow actions
  - Understand how to use and modify standard flow actions
- Work Status
  - Describe standard work statuses and how they are categorized
  - Apply work statuses to work objects
  - Understand how work is resolved

- Service Levels
  - Understand the time intervals on which service levels operate
  - Execute an activity based on reaching a goal or deadline
  - Understand the concept of Urgency as it relates to the work object and an assignment
- Correspondence
  - Identify the different correspondence types supported by Process Commander
  - List activities used to generate correspondence
- Work Parties
  - Explain the purpose of a work party
  - Describe how Data-Party classes determine the rule inheritance of work parties
  - Add a party rule and associate it with a work object

### Case Management

- Objects, Covers and Folders
  - Define the differences between the basic units of work and their associated work classes:
    - Work-Object-
    - Work-Cover-
    - Work-Folder-
  - Describe how flow execution moves between a work object and its cover or folder during processing
  - Explain when to use a work cover and when to use a work folder

### Data Modelling

- Data Classes
  - Understand how and where application data is stored
  - Create a data branch in an application
  - Use data classes to store information related to work
  - Use data classes as the basis of embedded pages, page lists and page groups within a work object
- Properties
  - Identify the correct class in which a property should be defined
  - Identify and describe the standard property prefixes
  - Describe the property modes that can be selected in a property rule
  - Understand the difference between the single, list and group type for the value and page modes
  - Define an embedded page
  - Name the syntax used to reference pages and properties

- Define a property that prompts a user to select a value
- Use the Define Properties Wizard to create multiple properties
- Use the Data Table Wizard to create a data table and its associated instances
- Models
  - Define models to initialize data
  - Describe how superclass model chaining works

## User Interface

- User Interface Rules
  - Define how different HTML rule types are used to construct the user interface:
    - Harness
    - Section
    - Flow Action
    - HTML Property
  - Identify the standard harnesses and their functions
  - Describe how sections can be embedded in other HTML rule types
  - Use the Rules Inspector to identify HTML rule types and properties
  - Understand the benefit of automatically generating HTML and the consequences of overriding this feature
- Property Display
  - Use HTML property rules to define field display and input formats for calendars, text fields, and select lists
  - Use Edit Input rules to convert data
- Validation
  - Understand how and when validation rules are used
- Portals
  - List the standard portals and their default users
  - Describe how to associate users and their portals

## Activities

- Activity Concepts
  - Define how and where activities are used in flows
  - List some of the standard activities and where they are used
  - Identify standard methods and other keywords
  - Describe the purpose of iterations, preconditions and transitions
  - Describe how to call an activity from a separate activity
  - Pass parameters between activities
- Step Pages

- Understand the purpose of the Pages and Classes tab and the relationship to the step page
- Understand when to use a step page and its scope
- Understand when to use a primary page and its scope
- Access embedded pages using the correct step page syntax
- Methods
  - Understand the behavior of Page-\*, Obj-\* and Property-Set methods
- Testing Activities
  - Run an activity from the rule form
  - Understand when to create a temporary test page
  - Explain which tools can be used to test activities

### Decision Rules

- Decision Rule Types
  - Describe the following types of decision rules and their function
    - Decision Tables
    - Decision Trees
    - When Conditions
  - Compare different decision rule types and identify when to use them
- Fork and Decision Shapes
  - Explain the difference between fork shapes and decision shapes in a flow

### Declarative Rules

- General Functionality
  - Understand the difference between procedural and declarative programming models
  - Describe the benefits of declarative processing
  - Use the Dependency Network Analysis (DNA) tool
- Rule Types
  - Identify the various computations types for Declare Expressions
  - Describe how Declare Expressions can be dynamically called
  - Understand how to use the Expression Builder to create complex expressions
  - Understand how to use functions in Declare Expressions
  - Describe the difference between thread and node Declare Pages
  - Understand when and how Declare Pages expire and can be refreshed

### Rule Management/Security

- RuleSets and Rule Resolution
  - Understand the basic concepts of RuleSet hierarchy and its prerequisites


- Understand the relationship between classes and RuleSets
- Understand RuleSet versioning and why it is important to application development and configuration management
- Describe when and how RuleSets are locked
- Explain the function of a personal RuleSet
- Describe rule resolution, how it works, and how it impacts RuleSet hierarchy design
- Describe how a RuleSet list is compiled
- Application Rule
  - Describe the function of an Application rule
  - Explain how Application rules can be built on each other
- Access Groups
  - List the various sections in an access group instance and their functions
  - Understand how access groups define an operator's access to the system
- Documenting the Application
  - Understand how to use the Documentation wizard and Application Inventory wizard
  - Understand how to use Rule Import and Export wizards

### Integration

- Services and Connectors
  - Understand the difference between a service and a connector
  - Add an integrator SmartShape to a process rule
  - Create a connector to a relational database
  - Create a connector to a SOAP service
  - List the most common service and connector protocols

### Performance

- Tracer
  - Setup and run the Tracer to debug a flow or activity
    - Set breakpoints
  - View saved state information in the tracer output
  - Understand how to review and analyze output at different steps in the execution and on parameter pages
- Clipboard
  - Understand how to navigate and use the clipboard to view existing pages
  - Identify standard clipboard pages and their functions, including:
    - pyWorkPage
    - pyWorkCover
- PAL

- 
- Understand when and how PAL is used in the construction of an application
  - Run PAL and interpret results to benchmark performance
  - Additional Tools
    - Understand the purpose and use of various monitoring tools, including:
      - Preflight
      - DB Trace
      - My Alerts