NOKIA

Business Service Tool
Release 19

Functional Description

Disclaimer

The information in this document applies solely to the hardware/software product ("Product") specified herein, and only as specified herein. Reference to "Nokia" later in this document shall mean the respective company within Nokia Group of Companies with whom you have entered into the Agreement (as defined below).

This document is intended for use by Nokia's customers ("You") only, and it may not be used except for the purposes defined in the agreement between You and Nokia ("Agreement") under which this document is distributed. No part of this document may be used, copied, reproduced, modified or transmitted in any form or means without the prior written permission of Nokia. If You have not entered into an Agreement applicable to the Product, or if that Agreement has expired or has been terminated, You may not use this document in any manner and You are obliged to return it to Nokia and destroy or delete any copies thereof.

The document has been prepared to be used by professional and properly trained personnel, and You assume full responsibility when using it. Nokia welcomes your comments as part of the process of continuous development and improvement of the documentation. This document and its contents are provided as a convenience to You. Any information or statements concerning the suitability, capacity, fitness for purpose or performance of the Product are given solely on an "as is" and "as available" basis in this document, and Nokia reserves the right to change any such information and statements without notice. Nokia has made all reasonable efforts to ensure that the content of this document is adequate and free of material errors and omissions, and Nokia will correct errors that You identify in this document. Nokia's total liability for any errors in the document is strictly limited to the correction of such error(s). Nokia does not warrant that the use of the software in the Product will be uninterrupted or error-free.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

This document is Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

Important Notice on Product Safety

This product may present safety risks due to laser, electricity, heat, and other sources of danger.

Only trained and qualified personnel may install, operate, maintain or otherwise handle this product and only after having carefully read the safety information applicable to this product.

The safety information is provided in the Safety Information section in the "Legal, Safety and Environmental Information" part of this document or documentation set.

Nokia is continually striving to reduce the adverse environmental effects of its products and services. We would like to encourage you as our customers and users to join us in working towards a cleaner, safer environment. Please recycle product packaging and follow the recommendations for power use and proper disposal of our products and their components.

If you should have questions regarding our Environmental Policy or any of the environmental services we offer, please contact us at Nokia for any additional information.

1 About This Document

This document describes the functionality of Business Service Tool, an add-on tool for creating provisioning logics to be used with InstantLink.

1.1 Audience

This document is intended for anyone who needs to have an overview of Business Service Tool and its functionality. The reader should be familiar with InstantLink.

1.2 Terms and Concepts

The following abbreviations, terms and concepts are used in the document.

1.2.1 Abbreviations

API	Application Protocol Interface
BST	Business Service Tool
GRC	Global Resource Configuration
NE	Network Element
NEI	Network Element Interface
OSS/BSS	Operations and Business Support System
UI	User Interface
XML	Extensible Markup Language

1.2.2 Terminology

activation	The process of modifying a subscriber's account records to create, modify or remove a subscriber's service.
Business Service Tool	A module that enables provisioning logic design and execution with InstantLink.
GRC parameter	A variable that is used to define configuration settings. GRC parameters are located in a database.
GRC section	A group of parameters for a particular application stored in a database. Each section is identified by the unique name of the application.
InstantLink	A system for subscriber provisioning and service activation from OSS/BSS to the communications network. InstantLink receives requests from the OSS/BSS systems, translates the requests to network-element-specific commands and executes these commands. After execution of a request, InstantLink sends a response to the OSS/BSS.
method	A building block in a provisioning logic. A collection of steps and also other methods used to arrange a set of operations into one logical group in the provisioning logic, making the provisioning logic more readable and structured.
Operations and Business Support System; OSS/BSS	A program that helps an operator monitor, control, analyse or manage usage of a communications network. OSS/BSS systems include, for example, systems for customer care, order management, billing, relationship management, decision support, market analysis, fraud detection, traffic engineering and network planning.
parameter pool	A set of parameter names and values related to a method or step during the execution of one request. The parameter pool is used to pass data in the provisioning logic between steps and methods.
provisioning	The process of configuring network and infrastructure to allow creation, modification or removal of a subscriber's service.
provisioning logic	A set of rules that define how provisioning is performed. A provisioning logic is presented in the Business Service Tool user interface by a flowchart, consisting of steps and methods. A provisioning logic can, for example, manipulate parameter data, execute a provisioning task, fetch information from lookup tables and create a response.
request	A work order that InstantLink receives from an OSS/BSS system and that consists of a required operation which is aimed at one subscriber or a logical service (for example, a VPN tunnel in an IP network) in one or more network elements.
response	In Business Service Tool, a message with the status of provisioning logic execution. A response indicates how the request execution succeeded. After all tasks in the provisioning logic are completed, Business Service Tool sends a response to InstantLink, which then routes it to the OSS/BSS.
step	A building block in a provisioning logic. Steps provide the actual functionality in a provisioning logic. Steps can be used, for example, for string manipulation, arithmetic operations, parameter mapping and checking as well as Boolean operations.

task	A work order for a network element.
task response	A work order (task) response from a network element.
unit	A step or method in a provisioning logic.

1.3 Related Documentation

The following documents provide more information about Business Service Tool:

- Business Service Tool Release Notes
- Business Service Tool Online Help
- Business Service Tool Reference Manual
- Business Service Tool User’s Guide (includes the same information as the Online Help)
- Provisioning and Activation Installation Guide

For more information about InstantLink, see the InstantLink documentation.

2 Introduction

Business Service Tool is an add-on module to InstantLink, which processes requests from Operations and Business Support Systems (OSS/BSS) and creates request responses to them. Business Service Tool enables complex request processing based on provisioning logics, for example, splitting one request into multiple tasks that are executed in one or several network elements.

Figure 1 illustrates the provisioning environment of Business Service Tool.

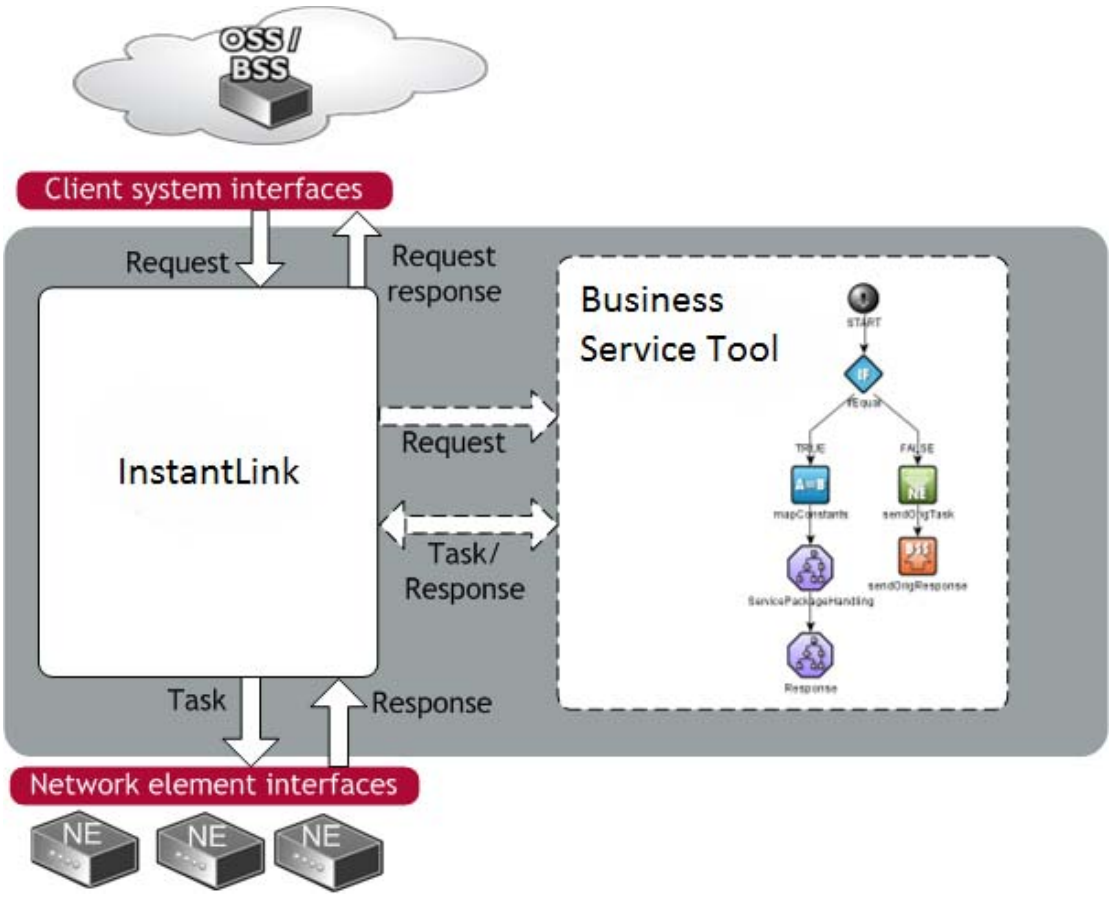

Figure 1. Business Service Tool provisioning environment

The request execution consists of the following phases:

1. The OSS/BSS system sends a request to InstantLink.
2. InstantLink stores the request and starts executing it.
3. InstantLink passes the request to Business Service Tool.
4. Business Service Tool receives the request, chooses the correct provisioning logic and starts to execute the provisioning logic.
5. The provisioning logic creates a task that must be executed in the network.

6. Business Service Tool gives the task to InstantLink.
7. InstantLink sends the task to a network element.
8. The network element performs the task and sends the task response to InstantLink.
9. InstantLink sends the task response to Business Service Tool.
10. Business Service Tool continues executing the provisioning logic, that is, repeats steps until all tasks have been executed.
11. The provisioning logic creates a response message.
12. Business Service Tool sends the response to InstantLink.
13. InstantLink sends the request response to the OSS/BSS system.

A provisioning logic is a set of rules that define how subscriber provisioning or service activation is performed. *Figure 2* illustrates how the UI presents the provisioning logic as a graphical flowchart.

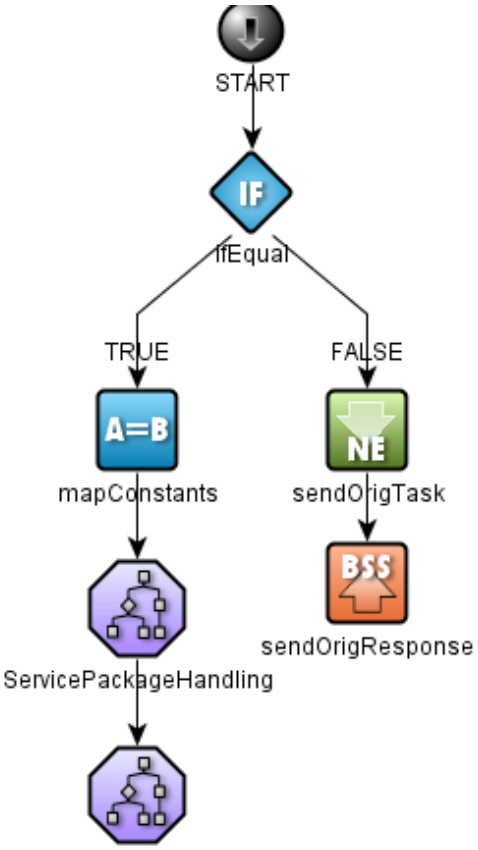

Figure 2. A provisioning logic flowchart

In Business Service Tool (BST), a provisioning logic is first constructed using Logic Editor and then activated to the InstantLink system. In this way, the entire InstantLink request-handling logic is easy to control.

3 BST Pages in the InstantLink UI

Business Service Tool functions are accessed from the InstantLink UI.

Business Service Tool provides different functions based on the user roles that have been granted to the user. For more information on the user roles, see *Business Service Tool Online Help*.

Provisioning logics are managed on the BST pages in the InstantLink UI. You can open the BST pages by clicking the **BST** main menu item (see *Figure 3*).

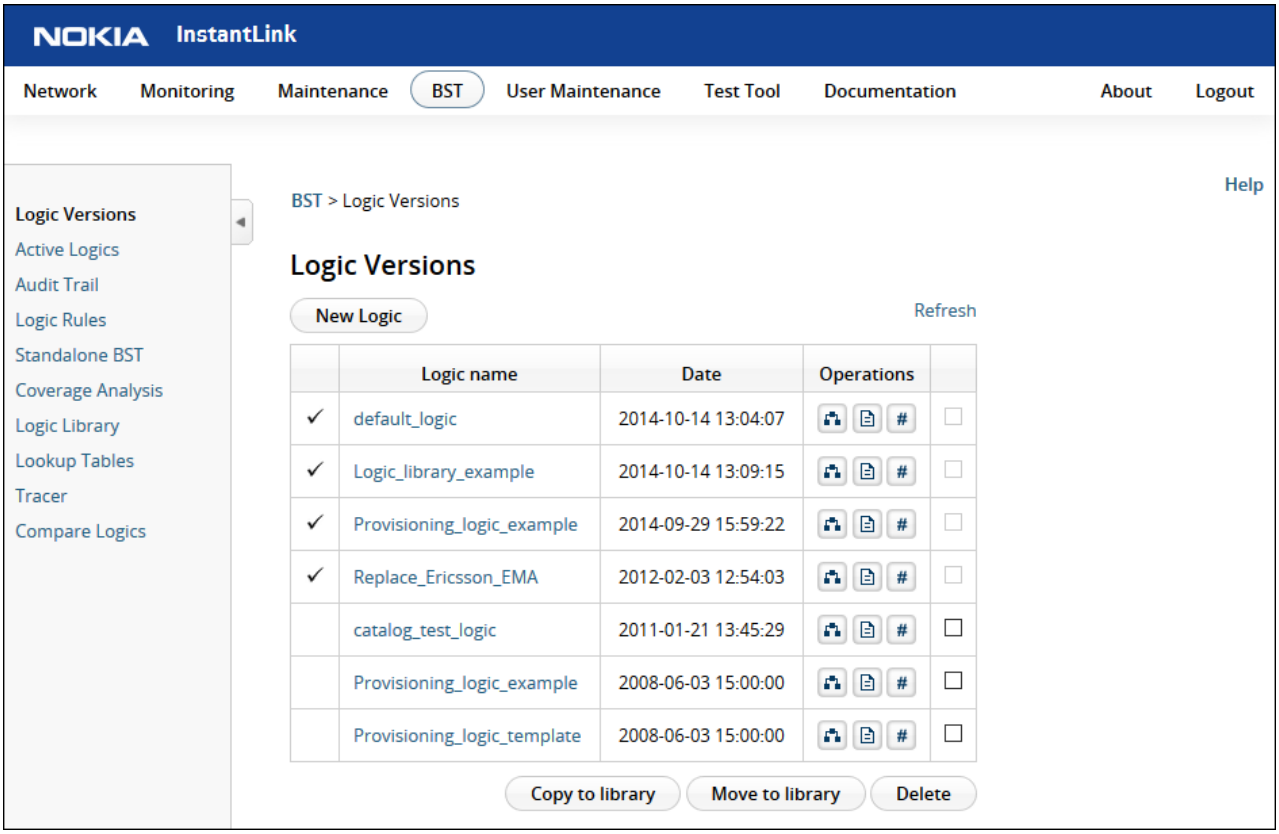

Figure 3. BST pages

On the BST pages you can:

- view all provisioning logics that are saved in the system
- start creating a new provisioning logic
- open provisioning logics

> **Note** The actual editing, that is, creating and configuring of provisioning logics, is done in Logic Editor. For more information, see Section 4 *Business Service Tool Logic Editor*.

- activate provisioning logics
- view configuration audit logs

- create logic rules to allow several logics to be active at the same time
- create a report on the provisioning logic that describes the methods, steps and parameters
- install a standalone version of Logic Editor that you can use without a server connection. For more information, see Section 4.1.3 *Using Standalone Logic Editor*.
- search for unused parts of a provisioning logic with the Coverage Analysis. For more information, see Section 4.3 *Coverage Analysis*.
- use library logics to divide logics into smaller, more functional entities that provide better support to customer's operating environments
- open Number Range Editor to manage number ranges in the NumberRange step. For more information, see Section 4.4 *Number Range Editor*.
- open Lookup Table Editor to manage lookup tables, see Section 4.5 *Lookup Table Editor*.
- compare two logics to find out the differences between them.

The execution path of a request can be followed with BST Tracer, which is a separate UI window that opens from InstantLink UI **Monitoring > Requests** page. For more information, see Section 4.2 *Request Tracing*.

When on the BST pages, click the **Help** link for instructions on using the BST pages.

4 Business Service Tool Logic Editor

The actual configuring of a provisioning logic is done in Logic Editor, which is accessed through the BST pages, either by clicking the **Create New Logic** link, **Edit Active Logic** link or clicking the **Edit** icon next to the provisioning logic that you want to modify (see *Figure 4*).

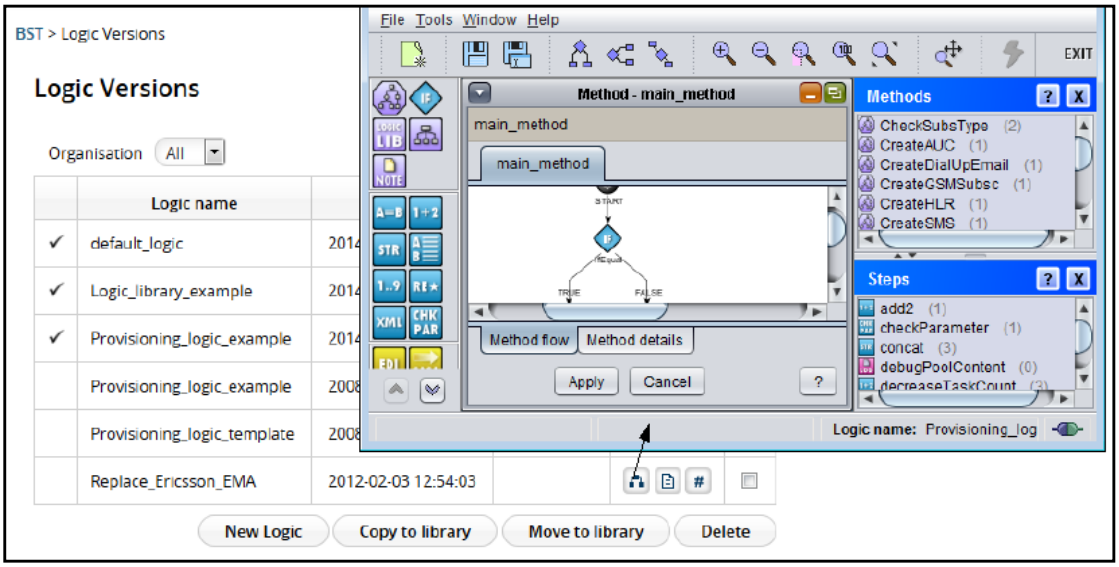

Figure 4. Logic Editor is accessed through the BST pages

The main functions of Logic Editor are:

- configuring a provisioning logic
- browsing a provisioning logic
- activating a provisioning logic
- viewing the change history of a provisioning logic
- uploading and downloading provisioning logics between a workstation and the server file system (This can also be done in InstantLink's UI.)
- searching steps and methods
- testing steps
- creating a report on a provisioning logic that describes the used methods, steps and the parameters

The UI is used with drag-and-drop and point-and-click actions.

For more information on working with provisioning logics, see *Business Service Tool Online Help*. Online help instructions on a specific UI window can be accessed by clicking the Help link. The entire online help can be opened by pressing F1 when Business Service Tool Editor is open.

4.1 Steps and Methods in Provisioning Logic

A provisioning logic consists of individual units. Each unit in a provisioning logic can be either a step or a method.

Steps are the actual building blocks that provide the functionality of provisioning logics. Steps can, for example, be used for Boolean operations (such as comparing parameters), execution branching, parameter mapping, and sending tasks to and receiving responses from network elements.

Methods are for grouping together larger functional parts, steps and other methods, to make a provisioning logic more readable and structured. One method can include several methods and steps. While methods themselves do not provide any functionality to the provisioning logic, they allow you to easily reuse parts of the logic in a different place in the logic.

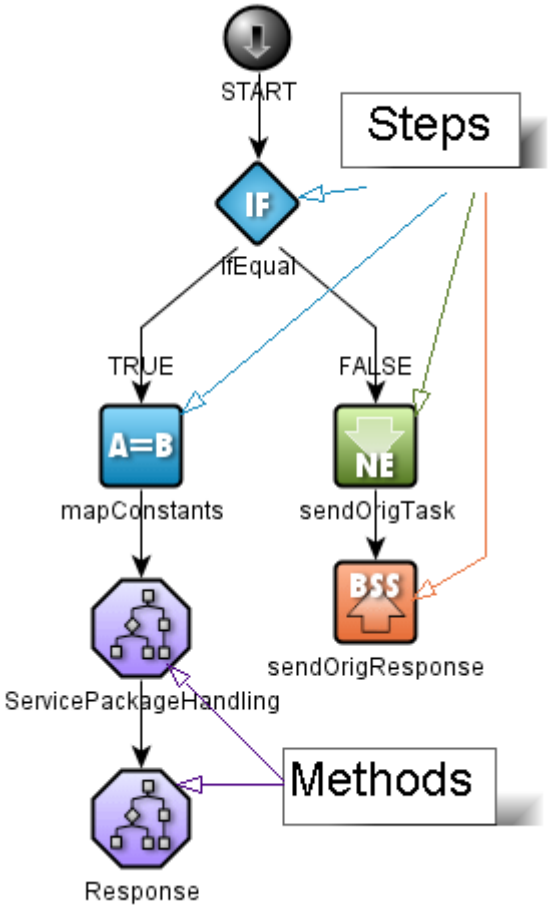

Figure 5. A provisioning logic consists of steps and methods

4.1.1 Business Service Tool Steps

The following steps are available for creating a provisioning logic:

- **ActivityStatus** for manipulating the activity status in Order Management (This step is available only if Order Management is installed)
- **Alarm** for sending an alarm to Alarm Dispatcher
- **AmendOrder** for amending orders from BST logic with configured parameters (This step is available only if Order Management is installed)
- **Branch** for defining several conditions based on which the provisioning logic continues to one of several branches
- **Calc** for performing a simple arithmetic or string operation
- **CheckParams** for checking whether defined parameters are in the parameter pool. For more information on the parameter pool, see Section 5.2 *Storing Parameters in a Parameter Pool*.
- **CreateSchedule** for manipulating the mainline during order execution (This step is available only if Order Management is installed)
- **Database** for executing SQL statements in the defined database
- **Date** for adding and subtracting time, performing date format conversions, comparing two dates and getting the difference between two dates
- **Edifact** for parsing Edifact messages and translating them into parameters
- **If** for choosing on the basis of conditions which of the two possible paths the execution of the logic should continue
- **LoadOriginalTasksParameters** for loading all original task parameters from a Request API request to the parameter pool
- **Log** for writing a message or the contents of the parameter pool to the log file
- **LogicLibrary** for calling library logics
- **LookUp** for loading parameters from the database with lookup rules
- **Manual** for sending a work item task to Workflow Client (This step is available only if Order Management is installed)
- **MapParameters** for creating new parameters and timestamps to the parameter pool and removing parameters from the pool
- **Notification** for sending notifications through the InstantLink for the purpose of external monitoring of the provisioning progress
- **NumberRange** for creating parameters based on number range comparison
- **OrderReschedule** for changing the schedule of an order in-flight by updating the customer required date (This step is available only if Order Management is installed)
- **ParallelSend2** for sending tasks to network element execution in parallel
- **PL/SQL** for executing PL/SQL procedures in the defined database

- **RegExp** for creating parameters based on Regular Expression comparison
- **SendOriginalTask** for sending an original task to be executed in a network
- **SendRequest** for sending a new request to InstantLink
- **SendResp** for sending a request response to OSS/BSS
- **SendSAS4Response** for sending Request API request response to OSS/BSS
- **SendToNe** for sending a task to be executed in a network element
- **String** for performing string operations
- **Switch** for creating parameters based on switch cases
- **UpdateBaseline** for updating baseline schedule at runtime (This step is available only if Order Management is installed)
- **UpdateSchedule** for updating start date, duration and name for activities and for changing the Point of No Return in an order (This step is available only if Order Management is installed)
- **XML_Parser** for parsing parameters from XML strings

A provisioning logic can be configured to include useful rollback operations in case a task fails. This means that you can define in the provisioning logic that when a specified task in a request fails, reverse operations are performed to cancel the changes done in the network. This way, the request does not remain partially executed in the network and the executed tasks need not be cancelled manually.

You can add **notes** to the provisioning logic by dragging the note icon from the step palette to the desired place in the flow. You can edit the note text at any time and move the note in the method flow.

4.1.2 Searching Steps and Methods

Logic Editor and BST Tracer provide a tool for searching steps, methods and parameters by name or type. Also parameter values can be searched and the search criteria can include regular expression. For more information, see *Business Service Tool Online Help*.

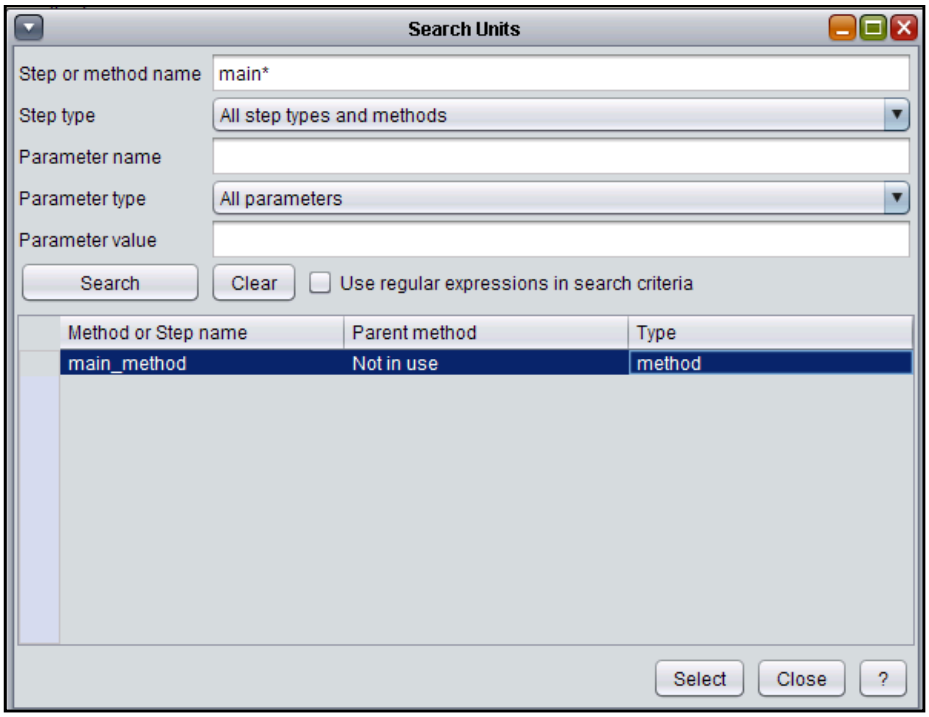

Figure 6. Searching steps and methods

4.1.3 Using Standalone Logic Editor

Logic Editor can be used as a standalone version, which allows you to develop provisioning logics on your workstation without a connection to the server. Standalone mode also improves stability: if connection to the server is lost unexpectedly, Logic Editor automatically switches to standalone mode and no data is lost. For instructions on how to install the standalone UI, see *Business Service Tool Online Help*.

4.1.4 Testing Provisioning Logic Steps

Logic Editor provides tool for testing steps before provisioning logic is saved and activated. The advanced test functionality makes the design and implementation of provisioning logic more efficient and improves service delivery to customers.

As shown in *Figure 7*, the user can select one ore more steps for testing and then supply the parameters used for testing. The test data can be either retrieved from a file or supplied manually.

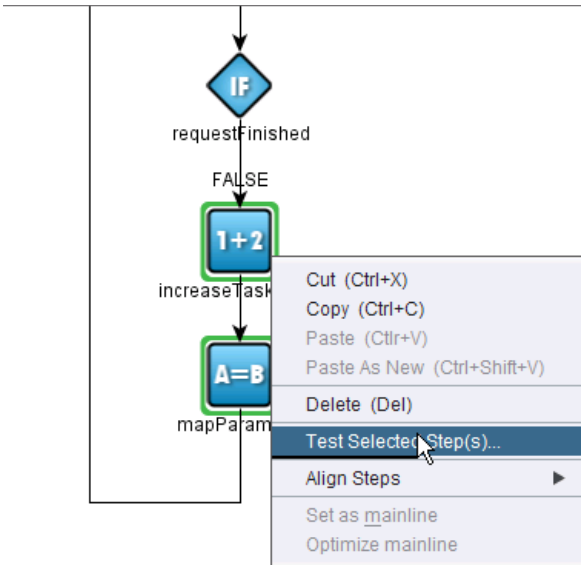

Figure 7. Testing steps

Some of the steps cannot be tested because they have external connections. These steps are: Alarm step, Database step, LogicLibrary step, LookUp step, ParallelSend2 step, PL/SQL step, SendRequest step, SendToNe step, SendOriginalTask step, SendResp step and SendSAS4Response step.

For more information, see *Business Service Tool Online Help.*

4.2 Request Tracing

With the BST Tracer, it is possible to follow the execution trace of a request and the request status as the request goes through the provisioning logic. The BST Tracer is a useful graphical tool for debugging purposes in a development environment when configuring a provisioning logic. Request tracing is not turned on by default, as it should not be used in a production environment.

In addition to following the execution trace of one request, you can also compare the execution traces of two requests run with the same logic version to see how they differ.

The BST Tracer can be accessed through InstantLink UI's **Request Details** page as well as through InstantLink's Test Tool. For more information, see *Business Service Tool Online Help*.

Figure 8 shows an example of BST Tracer. The execution trace is highlighted in green. For each step it shows step configuration and the parameters before and after the step.

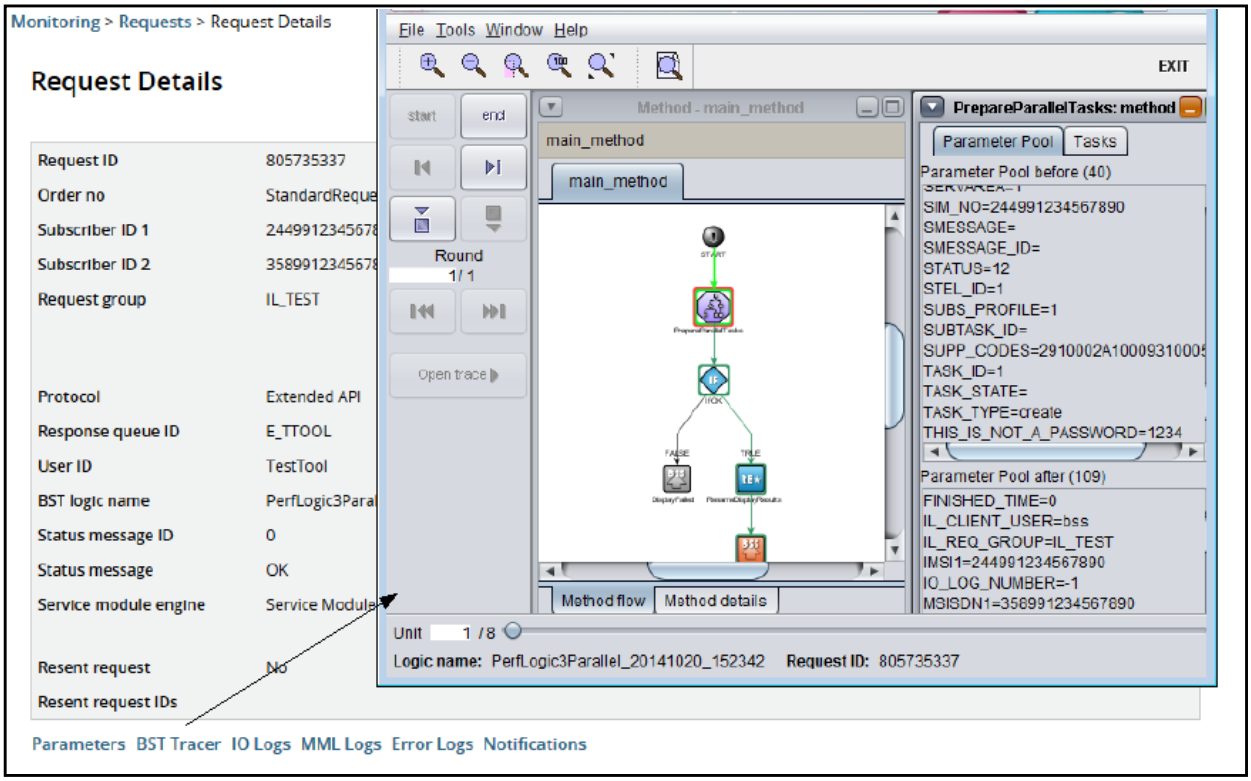

Figure 8. BST Tracer is opened from InstantLink Request Details page

4.3 Coverage Analysis

With the Coverage Analysis, you can search a provisioning logic for unused, potentially removable parts of the logic. For example, you can send 100 different requests and with the Coverage Analysis it is possible to see which steps and methods have been used in the requests and how often they have been used.

The coverage analysis is based on request tracing information, so the `request_trace` GRC parameter has to be set to `EXECUTION_PATH`, `EVERYTHING` or `MIXED`, or the request parameter `REQUEST_TRACE` must exist with value `EVERYTHING`, `EXECUTION_PATH` or `MIXED`.

.

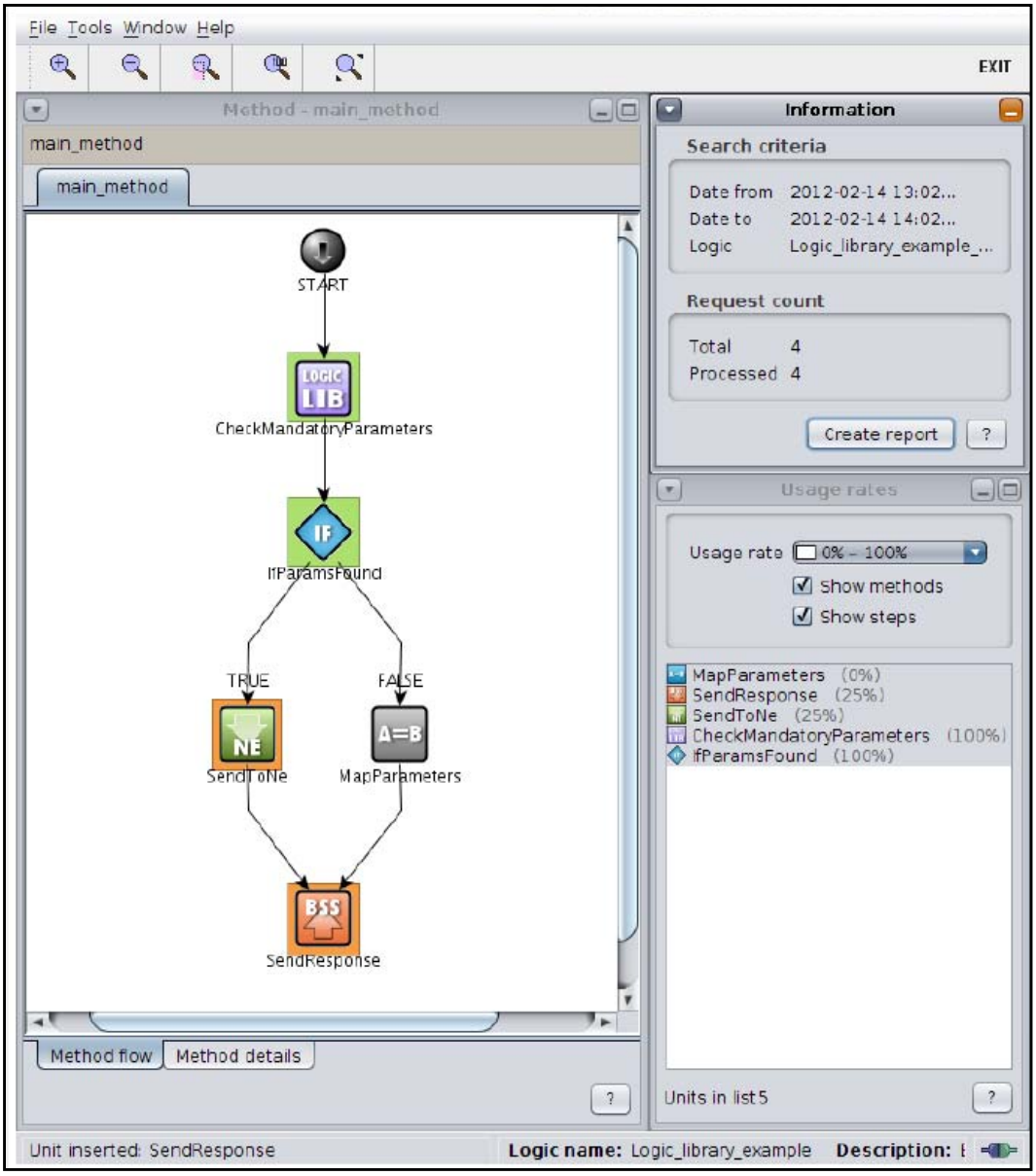

Figure 9. Coverage Analysis UI

For more information on the Coverage Analysis, see *Business Service Tool Online Help*.

4.4 Number Range Editor

Number Range Editor is designed to accurately and efficiently manage large number range definitions used in NumberRange steps. In case new number ranges need to be defined frequently, User Management allows a separate Maintain Number Ranges role to be granted to a user who can edit and update number ranges.

Additional number range validity check in Number Range Editor improves the number range management as all entered data is validated to make sure that, for example, there are no overlapping ranges within the same NE type.

The Number Range Editor is accessed through the BST pages by clicking the **NumberRange** # icon next to the provisioning logic.

For more information on the Number Range Editor, see *Business Service Tool Online Help*.

4.5 Lookup Table Editor

Lookup Table Editor is designed to accurately and efficiently manage lookup tables used in Lookup steps. User Management allows a separate Maintain Lookup Tables role to be granted to a user who can edit and update lookup tables.

Lookup Table Editor allows you to

- search for lookup tables based on lookup table name
- create new lookup tables
- create new lookup tables by copying an existing lookup table
- edit data in the lookup table
- delete lookup tables

The Lookup Table Editor is accessed through the BST pages in the InstantLink UI by clicking the Lookup Tables submenu item.

5 Functionality

5.1 Provisioning Logic Example

A provisioning logic always starts with the **Main** method. In the UI, it is called **main_method**. In the example below (see *Figure 10*), there are two submethods under the **Main** method: the **Create new subscriber** method and the **Modify subscriber** method. The **Create new subscriber** method contains two methods, **Silver** and **Gold**. These methods activate different services for the subscriber, depending on the parameters of the request.

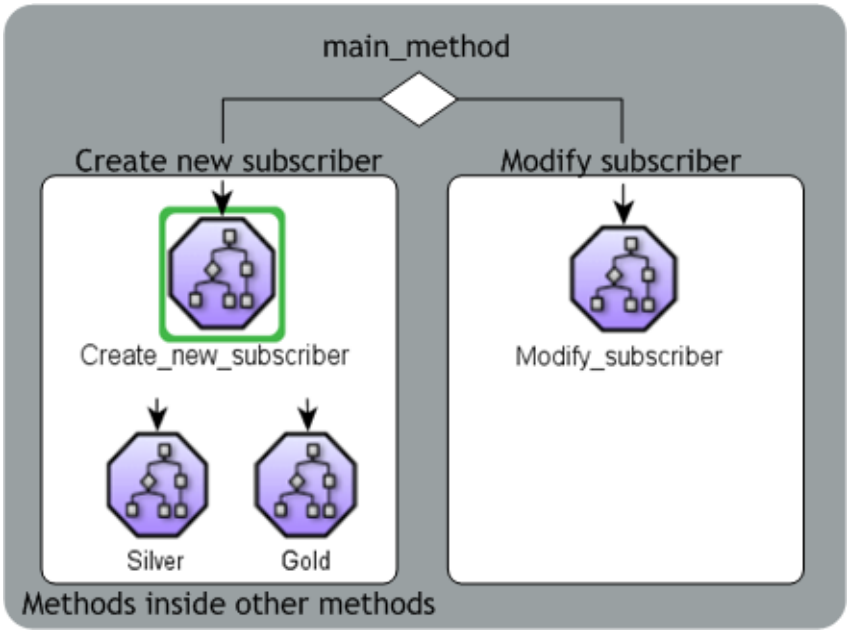

Figure 10. Example method hierarchy in provisioning logic

In this example, when a request arrives from InstantLink, Business Service Tool uses a provisioning logic to check the request's parameters to see whether the request is for creating a new subscriber. If that is the case, Business Service Tool performs the methods and steps as defined in the **Create new subscriber** method and sends a response to InstantLink. Otherwise the request is processed as defined in **Modify subscriber**.

In the example, the **Create new subscriber** method contains common rules for creating a new subscriber. The **Silver** method contains the basic set of services and the **Gold** method activates, for example, business user services.

Figure 11 illustrates the contents of the **Silver** method, a sub-method of the **Create new subscriber** method.

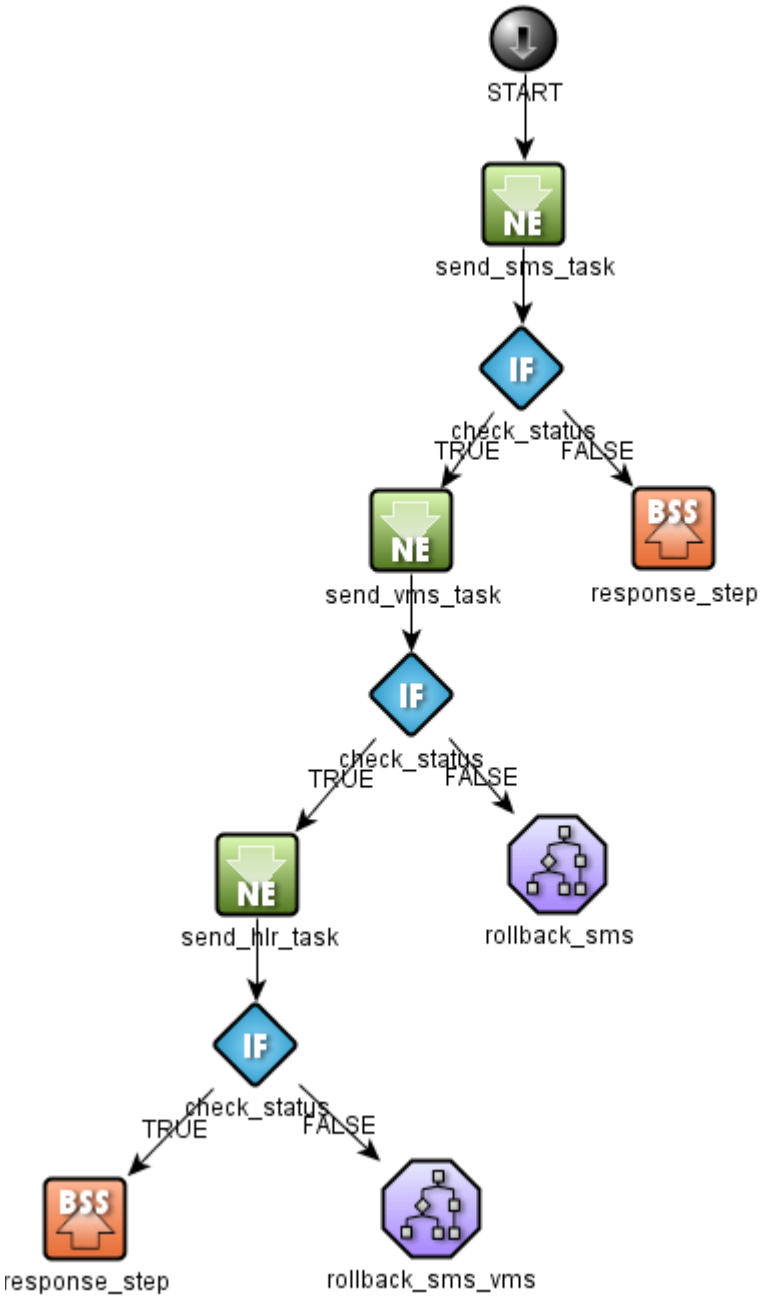

Figure 11. Content of the Silver method

5.2 Storing Parameters in a Parameter Pool

Provisioning logic parameters are stored in an entity called the parameter pool. Each request has its own parameter pool in the provisioning logic. The parameter pool acts as a communication channel between units (steps and methods) in a provisioning logic and it is used for storing request data during the execution of the provisioning logic. The parameter pool includes:

- incoming request parameters
- parameters generated during the execution of a provisioning logic
- response parameters for tasks sent to network elements

Parameters are stored in the parameter pool as ‘name’=‘value’ string pairs, for example, MSISDN=123456789123

The parameter pool is created when Business Service Tool receives a request from OSS/BSS through InstantLink, and is deleted when the request response is passed back from Business Service Tool, through InstantLink, to OSS/BSS.

When executing a provisioning logic, each unit during the execution of the request accesses the parameter pool content on its execution turn. Each unit executes its specific assignment and updates the pool parameters accordingly. The units can:

- create new parameters in the pool
- read existing parameters in the pool
- modify existing parameters in the pool
- delete existing parameters in the pool

The parameter pool content for each request is unique, meaning that when Business Service Tool executes several requests simultaneously, parameters are never mixed between requests.

5.3 Business Service Tool and System Refresh in InstantLink

When you make configuration changes to Business Service Tool, the changes are not applied until InstantLink system refresh is performed. At system refresh, Business Service Tool initialises itself and changes made to its configuration become effective.

During system refresh, database connections are initialised, GRC (global resource configuration) parameters, which define the behaviour of Business Service Tool, are read, and the active provisioning logics are loaded. This means that to activate a new provisioning logic to Business Service Tool, you only need to perform system refresh; restarting InstantLink or Business Service Tool is not necessary.

5.4 Guaranteed Correct Request Execution after Provisioning Logic Changes

Business Service Tool guarantees that all incoming requests and requests that are in execution while the system refreshes are executed with the original provisioning logic even if the active provisioning logic version is changed at system refresh. This means that requests are processed with the provisioning logic version that was active when the request arrived. This function can be set on or off with a GRC parameter.

5.5 Access Control

Based on different roles granted for UI users in the InstantLink, a UI user can access and operate the BST pages in the InstantLink UI, open and use Logic Editor, Number Range Editor, BST Tracer, and Lookup Table Editor through the respective add-on UIs.

The UI user's authorisation level is determined by granting roles to the UI user to define which UI tasks the UI user can perform. To grant a role to a UI user means to associate the UI user with a UI profile that contains the role. When creating a UI user, a user name and password is also defined to identify and authenticate the UI user when logging in. For more information on passwords and UI profiles, see InstantLink documentation. All UI users who need to access the BST pages in the InstantLink UI must have at least View BST Configuration role included in their UI profile.

5.6 OSS/BSS Interfaces

Business Service Tool supports the same OSS/BSS interfaces as InstantLink. There are special steps, such as LoadOriginalTasksParameters and SendSAS4Response, that are used with the Request API interface, but all other interfaces are handled as though they were Extended API interfaces.

5.7 Error Recovery

Business Service Tool stores data on its runtime activities in a database schema and log files for troubleshooting and monitoring purposes. Business Service Tool uses the same database as InstantLink and reads information from the database as well as writes data to the database via the connection it receives from InstantLink.

Business Service Tool writes the runtime information of a request to the database, including information about the processing of the request, for example, the current phase of the provisioning logic and the request’s parameters available to Business Service Tool.

When Business Service Tool is started after an unexpected system failure, Business Service Tool checks the data about previous activities in the database. If the execution of a provisioning logic was not finished before the failure occurred, Business Service Tool continues the execution of that logic.

5.8 Support for Workflow Client

Workflow Client can be used together with a Business Service Tool provisioning logic to provide manual interventions and new or modified parameters for the execution of a single request or ticket batches in the provisioning logic. For more information, see *Workflow Client Functional Description.*

5.9 Support for Order Management

Installing Order Management product adds new features to Business Service Tool. Among others, new steps are added. Changes to requests are enabled by sending tickets/work items to Workflow Client. For more information, see *Order Management Functional Description*.