





Prime Telecom Primer

VoiceObjects 9.0

To ensure that you are using the documentation that corresponds to the VoiceObjects software you are licensed to use, compare this version number with the software version shown in the Help menu of the VoiceObjects software you are using.

Copyright

Copyright © 2001-2009 VoiceObjects Germany GmbH and its licensors. All rights reserved.

Published in Germany – Legal information January 2009

Information in this document is subject to change without notice and does not represent a commitment on the part of VoiceObjects or any of its subsidiaries. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. You may not copy, use, modify, or distribute the software except as specifically allowed in the license or nondisclosure agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of VoiceObjects or a subsidiary thereof.

Protected by German and European patents. Further patents pending.

Companies, names, and dates used in examples herein are fictitious unless otherwise noted. If such names affect copyrights or trademarks or others, please notify VoiceObjects by e-mail at vo-documentation@voiceobjects.com.

Trademarks

VoiceObjects is a registered trademark of VoiceObjects Germany GmbH. Any other trademarks, trade names or service marks mentioned in this document belong to their respective owners.

The material presented herein is based upon information that we consider reliable, but we do not represent that it is error-free and complete. VoiceObjects is not making any representation or granting any warranty with respect to such material, and the distribution of such material shall not subject VoiceObjects to any liability.

Explicit Copyright Notice

The VoiceObjects software includes software developed by the Apache Software Foundation (www.apache.org). Copyright © 1999-2009 – The Apache Software Foundation. All rights reserved.

Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both.

Specific versions of the VoiceObjects software contain copyright material authorized by the Eclipse Foundation (www.eclipse.org), their contributors and others. All rights reserved.

Specific versions of the VoiceObjects software contain copyright material authorized to copy from Bocaloco Software LLC (www.xmlbuddy.com). All rights to such copyright material rest with Bocaloco Software.

Specific versions of the VoiceObjects software work with Microsoft Excel or make use of copyright material from Microsoft Corporation (www.microsoft.com). All rights to such copyright material rest with Microsoft. Microsoft and Excel are registered trademarks of Microsoft Corporation.

Document Number: E-033-20090604-VO9



Table of Contents

TABLE OF CONTENTS	3
PRIME TELECOM PRIMER	4
Introduction	4
The Application	5
The Main Menu	5
Service Plan Manager	5
Customer Data	6
Billing	7
Support	7
Personalization Features	7
Reporting on layers	8
The Voice Channel – VUI Best Practices	9
Handling voice and DTMF input	9
Navigation shortcuts	10
Avoid annoying the caller	10
Explicit confirmation	10
Implicit confirmation	11
Natural pronunciation of dynamic data	12
Adapt-to-me – the pronunciation feature	12
The Web and Text Channels	13
Basic formatting: Logos and cascading style sheets	13
Formatting output and dynamic data for Web and text	14
Multi-field Web forms	15
Cross-channel dialog strategies	15
Business Tasks	16
Handling business task execution failures	16
Reporting on business tasks	16
Business task data	18
Installation and Configuration	18
Installation Package	18
Configure the Formatting bus	19
Install the Prime Telecom application	19
Adjust resource locators	20
Deploy and test the service	20
Service parameters	21



Prime Telecom Primer

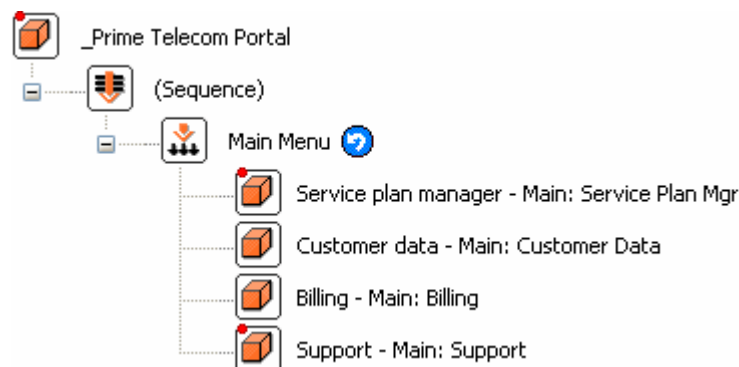
This guide provides an overview of the *Prime Telecom* sample application provided by VoiceObjects.

You may want to use *Prime Telecom* as a reference that you may come back to from time to time during ongoing multi-channel application development work.

Information on how to install and configure the *Prime Telecom* application can be found at the end of this guide.

Introduction

Prime Telecom implements a sample self-service portal for a telecommunication company, supporting three phone channels (voice, text, and Web) in three different languages (US-English, UK-English, and German). Callers may get information on their current tariff (or service plan), order new tariff add-ons, change customer data such as credit card or postal address, get billing information, or ask for support.



As a multi-channel application, *Prime Telecom* presents sample implementations for voice, text and Web interaction using VoiceObjects. For the voice channel, it demonstrates several VUI (voice user interface) best practices. For the text channel, it shows how to use menu- and free input-style patterns to interact with the caller. For the Web channel, it demonstrates the use of cascading style-sheets and other layout capabilities to achieve a compelling Web-based dialog application.

Highlights

- **Multi-channel development:** *Prime Telecom* serves as a reference of how to create a single service definition for the voice, text and Web channels.
- **VUI best practices:** The Voice User Interface (VUI) of *Prime Telecom* demonstrates how to implement some important concepts such as different confirmation strategies, natural pronunciation, navigation shortcuts implemented as hyperlinks, and more.
- **Personalization:** The Layer concept is used to implement different service strategies for different customer status and for different caller experience. In addition, specific marketing promotions are presented for each caller segment.
- **Reporting:** The *Prime Telecom* application has been designed to meet various logging and reporting requirements. For example, the application defines several business tasks, and the *Enable Layer state logging* option has been carefully configured for each Layer object.



- **Implementation best practices:** In *Prime Telecom*, almost all object types from the VoiceObjects library have been used to serve as an implementation reference.

The Application

The *Prime Telecom* application contains four different sub-applications accessible from a main menu. The following paragraphs describe each of them and provide information on how to use them.

The Main Menu

After entering *Prime Telecom*, the caller first listens to a welcome prompt before being presented the options of the main menu. This menu (Menu object *Main Menu*) looks and sounds very different in the different channels, while maintaining the same application logic. In the voice channel, it may play the initial prompt “*Please choose one of the following items: service plan manager, customer data, billing or support*”, while it displays a graphical user interface with icons and logos (controlled by cascading style sheets) in the Web interface, and a simple text menu in the text interface, respectively:



The main menu makes use of layers, random prompting, and occurrence levels. Its presentation differs depending on the time of day and it changes when the caller enters it more than once. The main menu offers cascaded error and help handling with increasingly verbose explanations. In the voice channel, there is also a DTMF fallback mode which is explained after successive failed inputs or help requests.

Service Plan Manager



The *Service Plan Manager* module presents the currently selected tariff (or service plan) and offers to add or change tariff add-ons. For certain callers, depending on the state of the layer *Customer Status*, a special add-on offer is presented as a marketing promotion on first entering this module; the caller can choose to order or ignore this offer.

When the caller likes the promotion and wants to subscribe to this tariff add-on, he must authenticate by providing a PIN for verification. Any 4-digit PIN will be accepted.

The *Service Plan Manager* menu has the following menu items:

- *Change rate plan* is just a stub. When selecting this item, you are told that this service is “Under Construction”.
- In *Additional Add-ons* you will find the actual content. You may want to use this menu item for presentations.



- Selecting *Change Add-ons* forwards the caller to an agent (who is supposed to talk the customer out of canceling tariff add-ons).

The Business Task object *Order Tariff Addon* is used to track the task of ordering an add-on. It is finished and counted as *aborted* if the caller chooses not to order the offered add-on.

Customer Data



In the *Customer Data* module, you can view and/or modify customer-related data, such as postal address, email address and payment settings. Changing the postal and email address is only available in the text and Web channel. If this option is selected in the voice channel, the (platinum) caller is transferred to an agent instead. This is an example of free-form input, such as names or addresses, which is easy to do in the text and Web channel, but always a challenge in the IVR.

The Business Task objects *Change Bank Details*, *Enter new Credit Card*, *Update Credit Card Expiration Date*, *Change Postal Address* and *Change Email Address* are used to track the tasks of changing the respective address information.

When callers enter the *Customer Data* module for the first time, they are prompted for their PIN (if not entered before in the same call). This is because *Prime Telecom* does not want to disclose personal contact information to non-authenticated callers. Any 4-digit PIN will be accepted.

The *Customer Data* module has 3 submodules:

Change Postal Address



In the *Change Postal Address* submodule, callers can confirm or change their postal address. This module is available only in the Web and text channels.

The caller is first prompted for the street and number of the postal address.

Second, the caller is asked for the ZIP code if the current language setting is *en-US*; for the postal code in *en-UK*; and for the Postleitzahl (PLZ) in *de-DE*. Here, you may enter any real postal code. The *Prime Telecom* connector code maintains lists of all current US, UK and German postal codes in order to look up the city name from the postal code. Note that in rare occasions, these lists might not be entirely up-to-date, since postal codes may change over time, while we don't maintain these lists in a regular way – after all, it's only a sample application.

So, when you enter the ZIP code *72205* while the active language is *en-US*, the city name and state will automatically be looked up and populated with *LITTLE ROCK, AR*.

For *en-UK*, the postal code *PA63* will be translated to *Croghan, Argyll and Bute*, and for *de-DE*, the PLZ *33729* will be identified as *Bielefeld, Nordrhein-Westfalen*, to give some examples.

The postal code lookup is implemented in the Connector object *Lookup City and State by ZIP code*.

Change Email Address



In the *Change Email Address* module, callers can confirm and change their email address. This module is available only in the Web and text channels.

You can enter any valid email address. The input is validated for compliancy with basic email address rules: The address must include an “@” symbol, and it must be terminated by a dot “.” followed by a 2- or 3-character suffix.



Change Payment Settings

Payment settings mean different things in *Prime Telecom*: In English (both *en-US* and *en-UK*), callers can confirm or change their credit card data. If the language is set to German, they can manage their bank account data. This module is available in all channels, including voice.

Credit Card (en-UK and en-US)

The caller can confirm their present credit card details, just update the expiration date, or enter a new credit card. Depending on the credit card type – Visa, American Express, or MasterCard – the credit card number is validated. By default, only the leading digits and the number of digits are checked according to the following rules:

Card Type	Prefix	Length
Visa	4	16
MasterCard	51 – 55	16
American Express	34, 37	15

Bank account details (German)

When using *Prime Telecom* in German, customers use their bank accounts for automatic bill payment rather than a credit card. Hence, they can review and change their bank account here.

For the account number no validation is performed.

For the bank code ("Bankleitzahl"), enter an 8-digit number. *Prime Telecom* holds a list of all German bank codes and performs a lookup for the name of the bank, using the bank code. E.g., when entering 37050198 for the bank code, the bank name will be automatically set to *Sparkasse KölnBonn* (current as of August 2008).

Billing



The *Billing* module is a mock-up only; no real functionality is implemented here. Instead, the caller is transferred to an agent.

Support



The *Support* module is a mock-up only as well. It is handled in the same way as the *Billing* module.

Personalization Features

In addition to the system layers Language (*English (US)*, *English (UK)*, *German*), and Channel (*voice*, *text*, *Web*), *Prime Telecom* defines a few custom layers that help control the complexity of the implementation.



Layer	How to control
Customer Status	<p><i>Prime Telecom</i> supports a layer to distinguish two levels of customer segmentation, <i>silver</i> and <i>platinum</i>. In a production-level application, the status would be fetched from a CRM database.</p> <p>In <i>Prime Telecom</i>, the Customer Status defaults to <i>silver</i>. To set it to <i>platinum</i>, provide the URL parameter <code>layCustStat=platinum</code> when calling the service.</p> <p>The main differences between <i>silver</i> and <i>platinum</i> customers are</p> <ul style="list-style-type: none"> • Different contact data and product portfolio • Easier access to live agents for <i>platinum</i> customers
Call Frequency	<p><i>Prime Telecom</i> keeps track of recent calls and can hence distinguish between a <i>new</i> call and a <i>repeat</i> call. A call is considered a <i>repeat</i> call if there has been another call from the same ANI within the last 60 minutes.</p> <p>Implementation: Connector object <i>Get Customer Data</i>, Connector item <i>Check for Repeat Call</i>. For an activity interval other than 60 minutes, change the value of the parameter with alias <i>minutes</i>.</p> <p>As an example for how this layer is used in <i>Prime Telecom</i>, <i>repeat</i> callers won't be offered the same marketing promotion again, and they don't have to listen through the "How to get back to the Main Menu" help prompt again.</p>
Caller Authenticated	<p>The status of this layer defaults to <i>Not yet</i>, and is changed to <i>Yes</i> after the caller has successfully authenticated by entering a PIN.</p> <p>Representing this status change by a Layer object has the additional benefit that it will be automatically logged to the Infostore Repository. This means that when generating statistical reports in VoiceObjects Analyzer, you can easily differentiate between calls where callers had authenticated and calls where they had not (yet).</p>
Daytime	<p><i>Prime Telecom</i> distinguishes three different times of day: <i>morning</i>, <i>day</i>, and <i>evening</i>. Depending on the time the caller calls in, different greeting prompts are used.</p>

Reporting on layers

Layers are not only a powerful technical means for personalizing the dialog flow, but they are also very useful for reporting purposes. At the end of a dialog, VoiceObjects Server automatically logs the final status of each layer to Infostore – unless the *Enable Layer State logging* option of the Layer object is deselected.

In *Prime Telecom*, for example, some layers like *Input Phase*, *Single Offer Mode* or *Last Inputmode* have this option deselected, because they are of purely technical purpose and don't tell you much about the caller or the call success.



Other layers like *Call Frequency*, *Customer Status* or *Daytime*, however, do tell you quite a bit about the call – like, was it a *first* or a *repeat* call, about the customer segment, and about the time of day.

VoiceObjects Analyzer provides a standard report *Layer Usage Overview*, which contains a table on layer usage and some basic call statistics:

Layer	Layer State	# Sessions	% Sessions	Avg Duration (sec.)	% Successful Input States
Caller authenticated	Yes	9,177	78.62%	95	80.51%
	Not yet	2,495	21.38%	70	70.89%
Caller authenticated		11,672		83	78.76%
Customer Status	Silver	14,064	77.56%	73	76.98%
	Platinum	4,070	22.44%	52	74.08%
Customer Status		18,134		62	76.46%
Summary		54,594		74	75.05%

You can see here in this example that more *Silver* than *Platinum* customers used the phone portal, but that the average call duration for the higher-valued group was much shorter. This helps to validate that the design which gives *Platinum* callers easier access to live agents actually works.

But more importantly than just looking at the layer usage overview, you can use layer states for filtering or grouping in any report within VoiceObjects Analyzer. So, you could investigate, for certain parts of the dialog, how the recognition success varies with different customer segments or regions, different time of day, different customer preferences, customer gender, customer age etc. – whichever of these information categories are represented by layers.

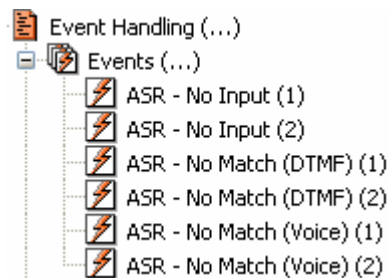
The Voice Channel – VUI Best Practices

The voice channel in *Prime Telecom* features a number of Voice User Interface (VUI) best practices. A few of them will be discussed in this paragraph.

Handling voice and DTMF input

For pretty much every input state in the voice channel of *Prime Telecom*, the caller is first prompted for voice input, and a fallback to DTMF is offered after the second No Match or No Input event.

This is reflected in the prompting and event handling strategy. There are always specific event handlers defined for the No Match (DTMF) and the No Match (Voice) events, respectively.



These event handlers play different intro prompts such as

"I'm sorry, this was an invalid input. ..."



for a first No Match (DTMF) event, or

“Sorry, I still didn’t get that. ...”




for a second No Match (Voice) event, for example.

In addition, the second No Match or No Input prompt usually tells the caller how to use DTMF keys for their input as an alternative:

“Sorry, I missed that again. Please tell me the brand of your credit card or press the corresponding key on your phone: 1-Visa, 2-MasterCard, or 3-American Express.”



Navigation shortcuts

Power users like shortcuts. Hence, the main menu has a few hyperlinks defined in the *Custom Navigation* section which allow the expert caller to bypass intermediate-level menus. So, calls like this are possible:

Object – Caller	Dialog Flow
	<i>How can we help you? Please choose: service plan manager, customer data, billing, or support.</i>
 Caller	<i>I want to browse for new addons.</i>
	<i>We have five new addons for you. ...</i>

Avoid annoying the caller

Well-intended, verbose help prompts tend to annoy the caller when they are repeated too often. As an example for how to avoid this, take a look at the Output object *GEN Info prompt - back to main menu* which is used as welcome prompt in each of the four main sub-dialogs. It plays this text:

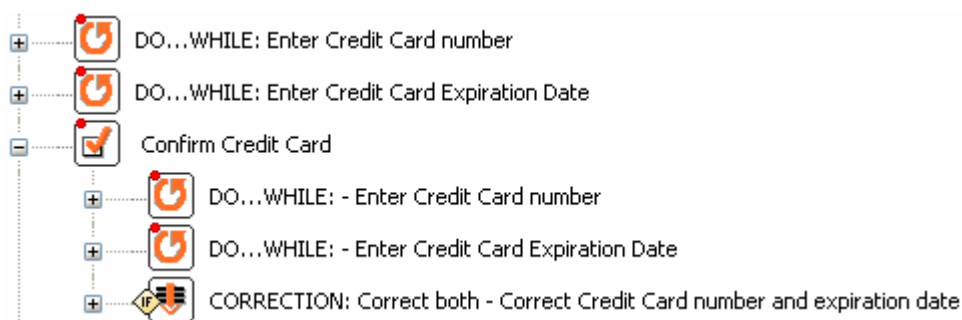
 (Bee-beep) *“By the way: You can get back to the main menu anytime by saying ‘main menu’.”*  (Bee-beep)

By setting the occurrence to *Only Once*, it is ensured that it is played only once per call.

And on top of that, we have configured the layer condition *Call Frequency = New Call*, so callers who call in repeatedly in a short time frame will not be bothered with this help prompt again after hearing it in the first call.

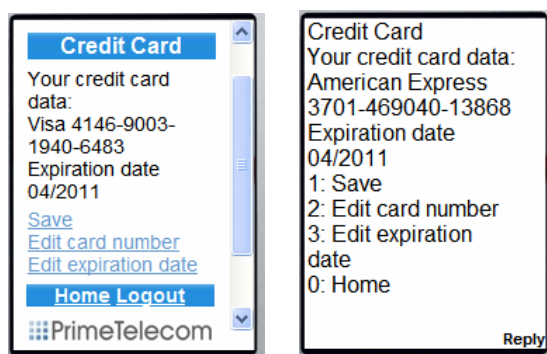
Explicit confirmation

Prime Telecom demonstrates different confirmation strategies – in particular, explicit and implicit confirmation. For the explicit confirmation, inspect the Confirmation object *Confirm Credit Card*, which is part of the *Enter new Credit Card* module.



In the dialog flow above – which is part of the *Enter new Credit Card* module – you see how the caller is first prompted for the credit card number, then for the expiration date. Finally, the full set of credit card data is played back for confirmation in a single prompt, and the caller can choose to confirm, correct the number, correct the date, or correct both.

Note how this concept of explicit confirmation works in all channels – not only in the voice channel, but also in the text and Web channels:



The option to correct both items in one go is only presented in the voice channel, which is displayed through the If icon in the dialog flow above.

Implicit confirmation

After providing the credit card type (such as *American Express*), the caller is prompted for the credit card number. Now, try the following dialog:

Object – Caller	Dialog Flow
	<i>What credit card do you have – Visa, MasterCard, or American Express?</i>
 Caller	<i>I have an AmEx Card.</i>
	<i>And what’s the number of your MasterCard?</i>
 Caller	<i>No, it’s an American Express!</i>
	<i>Sorry, American Express. And what’s the number?</i>



Note how the input – the credit card type – is implicitly confirmed, allowing the caller to correct the input in a single step.

Natural pronunciation of dynamic data

The *Prime Telecom* application uses the Formatting bus for natural pronunciation of data types such as date, currency, credit card numbers etc.

Call into *Prime Telecom* and enter a new credit card. Listen to the confirmation prompt that plays back your credit card number and expiration. The credit card number will be grouped into digit groups matching the number pattern of the respective credit card type, and the prompt will be assembled from a set of audio files containing 5 different recordings per digit.

If you want to see how it is implemented, open the Variable object *Credit Card Number* and inspect the **Formatting** section. You will find two Format items, one for the voice channel (“Format TTA - ...”), and another one for the Web and text channels (“Format text - ...”).

Other places where the Formatting bus is used include the Variable objects *Bank Account Number* and *Bank Code (Digits)*, *Base Tariff Rate (Currency)*, the Expression objects *Get Total monthly rate (Currency)* and *Number of Remaining Add-Ons (Cardinal number)*, and the Expression object *First Day Next Month (formatted) (Date)*.

Adapt-to-me – the pronunciation feature




Next, let us have a closer look at the credit card expiration date. The grammar that is used in the Input object *Get Credit Card Expiration Date* allows for utterances such as

- *December two thousand and nine*
- *December owe nine*
- *twelve owe nine*

all of which will be interpreted as the expiration date “12/09”.

Call into *Prime Telecom*, navigate to the *Payment Settings* menu, and choose *Update expiration date*. Try these different utterances and listen to the confirmation prompt and see how the confirmation prompt adapts to the way you chose to speak the expiration date.

Test call 1:

Object – Caller	Dialog Flow
	<i>Then tell me the new expiration date!</i>
 Caller	<i>July two thousand eleven.</i>
	<i>So the new expiration date is July two thousand eleven. Correct?</i>

**Test call 2:**

Object – Caller	Dialog Flow
	<i>Then tell me the new expiration date!</i>
 Caller	<i>Owe seven, eleven (or DTMF: 0-7-1-1).</i>
	<i>So the new expiration date is owe seven, eleven. Correct?</i>

With the confirmation prompts playing the data in the way the caller spoke them, the chance that the caller successfully recognizes their input increases considerably. The same concept can be applied for instance to account numbers or phone numbers, where proper grouping of digits and numbers – repeating the numbers in the same way as the caller said them – is crucial for a successful confirmation step.

This feature makes use of the “pronunciation” value that is automatically assigned to the Variable object *Credit Card Expiration Date* in the Input object *Get Credit Card Expiration date*. In order to populate this pronunciation value with meaningful input, it is necessary to provide a grammar that fills slots not only with the interpretation, but also with pronunciation information.

In this case, the TTG grammar is created dynamically by the Connector object *Create Expiration Date TTG Grammar*. The grammar generated by this connector consists of lines such as

```
...
may two thousand eleven (052011||mmm yyyy),
may eleven (052011||mmm yy),
oh five two thousand eleven (052011||mm yyyy),
oh five eleven (052011||mm yy),
...
```

The double pipe symbol “||” separates the interpretation (*052011*) from the pronunciation information. “mmm” means that the month was referenced by name, “mm” by number. Accordingly, “yyyy” stands for a 4-digit year, “yy” for a 2-digit year reference.

This pronunciation information is leveraged in our implementation of the formatting type *Format TTA – Credit Card Expiration Date*.

For further information on the pronunciation feature, refer to *Input* in the *Object Reference*.

The Web and Text Channels

The text and Web channels in *Prime Telecom* represent a number of capabilities some of which are described in this paragraph.

Basic formatting: Logos and cascading style sheets

In the Web channel, the *Prime Telecom* application has a consistent look-and-feel based on a blue color scheme and the *Prime Telecom* logo displayed on the bottom of every page.

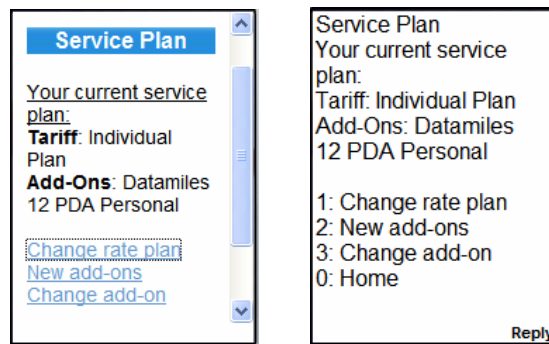


These formatting resources are configured in a central location: In the **Tuning** section of the Module object *Prime Telecom Portal*. All other Input and Menu objects inherit the tuning properties from this Module object.

Open the Module object *Prime Telecom Portal* to see how the *Presentation – Bottom Logo URL* and *Presentation – Style Sheet URL* properties are defined, along with other tuning parameters that apply only to the voice channel, like the *Input - Incomplete Timeout* property.

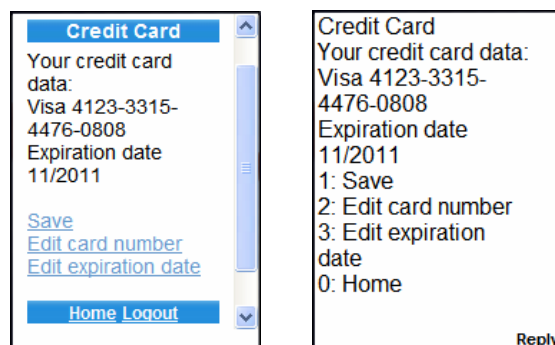
Formatting output and dynamic data for Web and text

When navigating from the main menu to the *Service Plan Manager*, the application first presents the customer's current product portfolio. Note how, in the Web channel, the *Prime Telecom* application makes use of basic HTML formatting, such as using bold face and italic fonts, underlines and more.



Open the Output object *Present Current Products (Text/Web)* to see how these formatting tags are defined inline in the Output item. Note also that this Output item is used both for the Web and the text channel – that is possible because inline HTML tags such as `` or `` are simply ignored by VoiceObjects Server when rendering markup for the text channel. In effect, you will be able to re-use most of your Web channel service implementation very easily in the text channel.

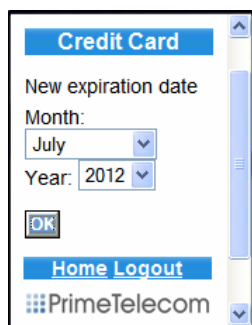
While the voice channel uses the Formatting bus to implement special rules for playback of dynamic data such as credit card expiration date, number, or dates, the text and Web channel also make use of special formatting algorithms that convert raw data into an optimized form for display on the screen. As an example, the formatting type *Format Text - Credit Card Number* adds hyphens after a group of 4 digits:





Multi-field Web forms

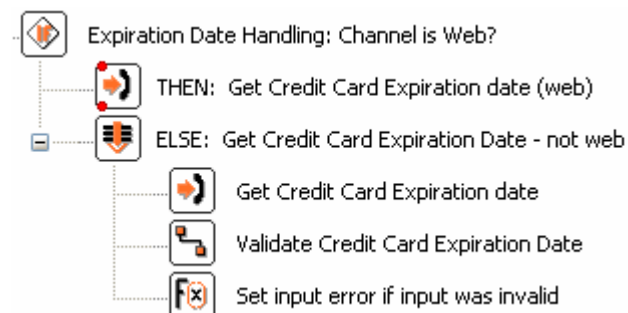
The Input object *Get Credit Card Expiration Date (web)* shows how to implement a multi-field Web form. It features two slots – *month* and *year* – which are both mapped to drop down input fields. Inspect this Input object to see how it is configured with tuning parameters (defining the field labels and field types) and a TTG grammar.



Note that the TTG grammar is created on the fly by the Script object *Create Expiration Date TTG Grammar (web)*.

Cross-channel dialog strategies

While most of the input states and menus in *Prime Telecom* work in all supported channels, sometimes it is preferred to have different, channel-specific implementations of certain parts of the dialog. As an example, take the Module object *Update Credit Card Expiration Date*, and more specifically the input step where the caller is prompted for the expiration date.



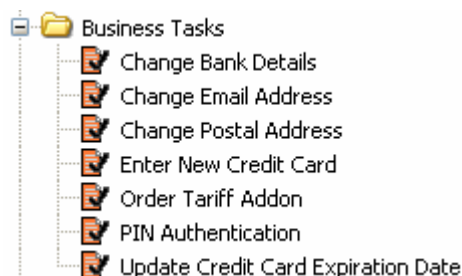
The part of the dialog represented in the dialog flow above starts with an If object that checks a channel condition. In the Web channel, it calls a two-slot input state which displays two drop downs – one for the month and one for the year of the expiration date (cf. screenshot in the paragraph above). Since the caller can only choose valid data from the drop downs, there is no extra validation step necessary here.

In the text and voice channels, on the other hand, the caller input is managed by an input state with a single slot only, followed by an explicit validation step.



Business Tasks

The *Prime Telecom* application defines a few Business Task objects as well as expressions for starting and stopping them.



Note, for example, how the Business Task object *Update Credit Card Expiration Date* is started first thing in the Module object *Update Credit Card Expiration Date*, and stopped only after the caller successfully entered and confirmed the expiration date, and after it was written back to the *PaymentSettings* Collection object.

Handling business task execution failures

Now, what happens if the caller hangs up, not having finished the confirmation step? Or if they navigate back to the main menu before completing this task? Or if a back-end error occurs while the business task is still active?

The behavior of *Prime Telecom* in these and similar cases is defined again on the top-level Module object *Prime Telecom Portal*. Have a look at the following configuration items of this object:

- **Events:** *ASR NoInput/NoMatch (3)*, *Error Connector (1)*
- **Custom Navigation:** Hyperlinks *Main Menu*, *Logout*, and *Talk to Agent*

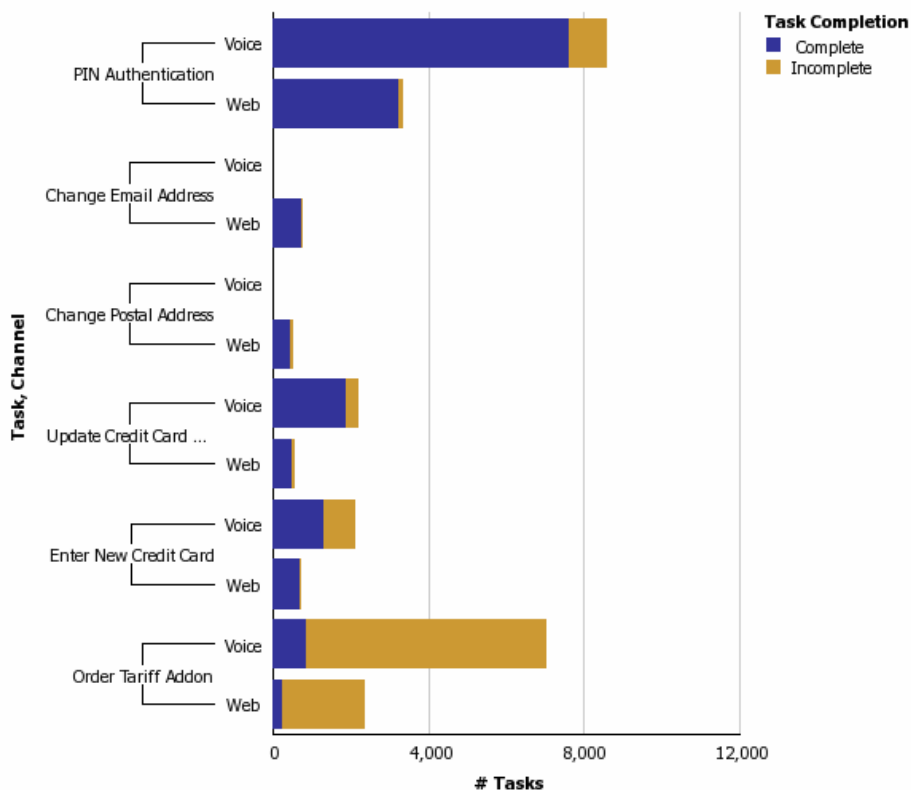
All these items have a configuration field called **Finish Tasks**, which is set to *All*. This tells VoiceObjects Server that whenever one of these events or hyperlinks is triggered, any currently active business task must be finished immediately and marked as *Incomplete*. On top of that, the status of these business task executions will be set to *Recognition failure* (for ASR related events), *Backend Error* (for the *Error Connector* handler), or *Caller abort* (hyperlinks).

Reporting on business tasks

Based on these business tasks, cross-channel reports on automation success can be generated in VoiceObjects Analyzer. As an example, the following screenshot shows how the standard report *Business Task Completion Rates* could look like. It shows how often each business task was executed per channel, distinguishing between *Complete* and *Incomplete* executions:



VoiceObjects Analyzer
Business Task Completion Rates



This chart is meant to answer the most important questions from a management perspective in a nutshell: What is the bottom line – how successful are our self services? What is the automation rate, and what are the transaction completion rates?

The same standard report features a table for each channel with more detailed information on the different reasons for failure of the business tasks. You will be able for instance to see at a glance the percentage of callers that didn't get through the *Enter New Credit Card* task because of recognition failures or because of premature hang-ups (session termination):

Voice

		Complete		Incomplete				# Tasks		
		Complete		Caller abort		Recognition failure			Session termination	
		# Tasks	%	# Tasks	%	# Tasks	%		# Tasks	%
Authentication	PIN Authentication	7,640	89%	119	1%	280	3%	552	6%	8,591
Transaction	Enter New Credit Card	1,300	62%	131	6%	327	16%	343	16%	2,101
	Order Tariff Addon	883	13%	5,751	82%	245	4%	115	2%	6,994
	Update Credit Card Expiration Date	1,900	87%	37	2%	96	4%	147	7%	2,180
Summary		11,723		6,038		948		1,157		19,866

Note that with VoiceObjects Analyzer, you will also be able to filter for individual sessions by searching for example for all sessions with a certain business task having failed because of *Recognition failure*. You will be able to analyze these sessions in all necessary detail, down to the level of listening into utterance recordings, in order to find out where and why callers struggled with a certain task.



Business task data

Now, take a closer look at the Business Task objects that are defined in *Prime Telecom*: Some of them have parameter sets defined. For example, the Business Task object *Order Tariff Addon* has the Expression object *Add-On Monthly Rate* referenced in its parameter set *Complete*.

Again, this has been done for reporting purposes. Business task parameter sets accomplish what used to be implemented with custom logging: Application-specific data is being logged to a database in the context of the standard session detail log data. Technically, these parameters are automatically evaluated and stored with the business task execution result once the task is terminated – regardless of whether it was complete or incomplete.

When generating the standard report *Business Task Data*, you select a business task to report on. The following image shows a typical result for the *Order Tariff Addon* business task:

VoiceObjects Analyzer

Business Task Data

Task Completion	Task Data Key	Total	Count	Count Distinct	Min	Max	Avg
Complete	AddonMonthlyRate	15,778.77	1,123	3	9.99	39.99	14.051

Report Filter

Date: Year (All)
Service: Prime Telecom
Task: Order Tariff Addon

The report shows how often this task was being finished successfully by a caller, what the total additional revenue from the up-sell option was (i.e., the sum of all individual *Addon Monthly Rates*), plus some basic statistics on minimum, average, and maximum of this value.

In short, this is a simple, straightforward way to measure the success of up- or cross-sell initiatives in the self-service channels.

Installation and Configuration

Installation Package

The installation package consists of two archives:

- *TTABus.zip*
Contains the configuration files *Format-Text.xml* and *Format-TTA.xml* as well as the Java archive *ttalib.jar*.
- *PrimeTelecom.zip*
Contains all resources (project export file; audio, grammar, image resources, etc.) for the *Prime Telecom* application.

Note that for the voice channel, we currently only provide audio files for *en-US* and *de-DE*. The prompts for *en-UK* will be added in a future release of *Prime Telecom*.



Configure the Formatting bus

Before installing the actual *Prime Telecom* application, the Formatting bus must be configured.

The implementation of the Formatting bus that comes with *Prime Telecom* features formatting algorithms for ordinal numbers, date, time, currency values, digit sequences etc., both for the voice channel and the Web/text channels.

The installation and configuration involves 3 steps. When following this installation description, set

[base] = *VoiceObjects\Platform\WEB-INF*

for installations of VoiceObjects Server and VoiceObjects Desktop, and

[base] = *VoiceObjectsDesktop\plugins\com.voiceobjects.eclipseDesktop_9.0.0*

for VoiceObjects Desktop for Eclipse (DfE).

1. Unzip the archive *TTABus.zip* to the *[base]* folder. This will copy *ttalib.jar* to *[base]\lib\TTALibrary*, and *Format-Text.xml* and *Format-TTA.xml* to *[base]\config*.
2. Open *[base]\config\Eclipse_Configuration.xml* in a simple text editor and add the lines

```
<TTALibraries>
    <TTALibrary>Format-Text.xml</TTALibrary>
    <TTALibrary>Format-TTA.xml</TTALibrary>
</TTALibraries>
```

right before the final “</config>” line.

For VoiceObjects Server/Desktop:

Instead of *Eclipse_Configuration.xml*, open and edit

[base]\config\VO_Server_Configuration.xml and

[base]\config\VODesktop_Configuration.xml accordingly.

Note that if you are using Desktop for Eclipse in network mode, you must configure the Formatting bus both on the Server (full installation) and on your Desktop for Eclipse client (here, *ttalib.jar* is not required).

Install the Prime Telecom application

The *Prime Telecom* package consists of the application itself, which is provided as an XML export file, and a set of resources: Connector code, audio files for the voice channel, and graphics and cascading style sheets for the Web channel.

To install the application, follow these steps:

1. Extract the archive *PrimeTelecom.zip* to the *Resources* folder in your VoiceObjects installation. This will generate the subfolders *VoiceObjects\Platform\Resources\Samples\PrimeTelecom* (VoiceObjects Server), or *VoiceObjectsDesktop\plugins\com.voiceobjects.eclipseDesktop_9.0.0\Resources\Samples\PrimeTelecom* (VoiceObjects Desktop for Eclipse), respectively.
2. Create a new project, e.g. *Prime Telecom*. In Desktop for Eclipse, you create a new project by right-clicking the *Projects* folder in the **Repository Browser** and selecting **New** from the context menu. Open the new project.



3. Import the XML export file *PrimeTelecom.xml* (located in *Resources\Samples\PrimeTelecom*) containing the definition of the *Prime Telecom* application. (In Desktop for Eclipse, first open the new project, then open the **VoiceObjects** menu and select **Import**.)

Adjust resource locators

Before deploying *Prime Telecom* as a service, make sure to adapt the following two resource locators to your system setup:

The **Base Locator** needs to point to the base URL of the *Prime Telecom* resources. By default, the URL points to *Resources/Samples/PrimeTelecom/*, i.e. it references the resource base folder relative to the VoiceObjects Server base URL. Usually, you don't need to change this. The physical path, on the other hand, defaults to *C:\VoiceObjects\Platform\Resources\Samples\PrimeTelecom*. The physical path is used when browsing for files within VoiceObjects Desktop. Also, it is used for storing the recordings generated by the Recording object *Record a Message (Voice)*. The path must be valid and VoiceObjects Server must be allowed to write to that path.


The **Connector Locator** needs to point to the base URL of the Java and JSP connectors. This Locator references the variable *Connector Locator URL* which, by default, points to

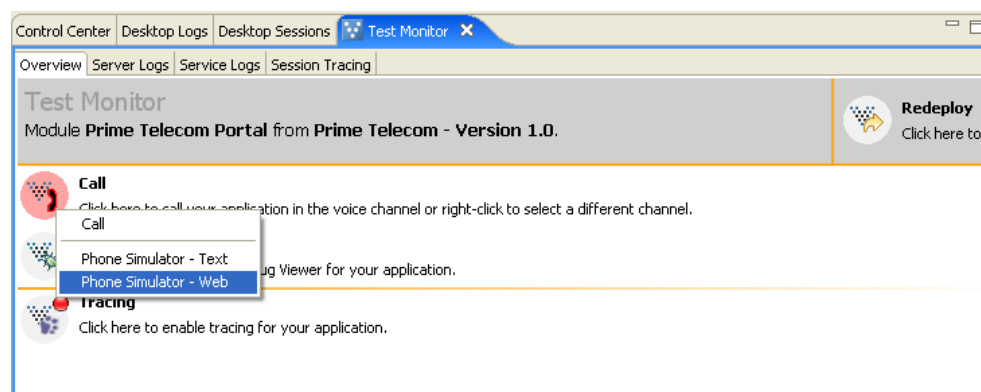
```
http://localhost:8070/VoiceObjects/Resources/Samples/PrimeTelecom/  
Connector/
```

which is appropriate for a default installation of Desktop for Eclipse with the embedded instance of VoiceObjects Server listening on port 8070; note that the default port for a full VoiceObjects Server installation is 8099.

Deploy and test the service

Finally, deploy *Prime Telecom* as a service, using the Module object *Prime Telecom Portal* as start object.

1. In Desktop for Eclipse, right-click the *Prime Telecom Portal* Module object in the **Object Browser** and select **Test Application** from the context menu. The current application, with *Prime Telecom Portal* Module as the start object, will be deployed automatically as the internal service *testService* and show up in the **Test Monitor**.
2. Right-click the **Call** button  and from the context menu select *Phone Simulator - Web* to test the application in the Web channel.





This will launch the Phone Simulator and call your test service. In the Phone Simulator, you see the *Prime Telecom* start page displaying the main menu with its four icons (see [The Main Menu](#) above). Now you can test the *Prime Telecom* service in the mobile Web channel.

Refer to the *Desktop for Eclipse Tutorial* or to the *Deployment Guide* for further information on service deployment.

Service parameters

When you call the *Prime Telecom* service either as a voice, Web or text service, you may initialize your session with optional parameters.

1. Setting the customer status:

Syntax: `layCustStat=platinum`

With this setting, you simulate a *Platinum* customer calling in. By default, the customer status is *Silver*.

2. Setting the Language:

Syntax: `vsLanguage=de-DE`

Supported languages are `en-US`, `en-UK` and `de-DE`.