

Orca User's Guide

Orca User's Guide

Copyright 2005-2008, Sun Microsystems, Inc.

Table of Contents

Foreword.....	iv
1. Introduction.....	1
2. Running Orca.....	2
3. Customizing Orca	4
3.1. orcaModifierKeys - Override "Insert" as Orca Modifier Key.....	6

Foreword

Orca is a flexible, extensible, and powerful assistive technology that provides end-user access to applications and toolkits that support the AT-SPI (e.g., the GNOME desktop). With early input from and continued engagement with its end users, Orca has been designed and implemented by the Sun Microsystems, Inc., Accessibility Program Office.

NOTE: Orca is currently a work in progress. As a result, this and other books in the Orca Documentation Series are under continuous modification and are also in various states of completeness.

This book provides a guide for the installation, configuration and use of Orca. It also includes details on how users can customize the keyboard and braille input mappings of Orca.

Chapter 1. Introduction

TODO: To be written. This currently contains just the most important stuff you need to know. In addition, you should always visit <http://live.gnome.org/Orca> (<http://live.gnome.org/Orca>) for the latest information and hints on using Orca.

Chapter 2. Running Orca

You can run Orca in many ways:

- Using the `orca` command from a terminal or virtual console. For many users, this is the easiest and most accessible way to launch Orca.
- Via the "Run Application" dialog that you can bring up by pressing `Alt+F2`. In this dialog, type `orca` and press `Return`.
- Via the "Orca Screen Reader and Magnifier" menu item available from the "Accessibility" submenu off of the "Applications" menu on some systems.
- For GNOME 2.16 and later, via the "Assistive Technology Support" preferences dialog that you can launch from the "Preferences" menu on some systems. This will enable accessibility in your environment and tell Orca to automatically launch when you log in.
- Via the "Startup Programs" tab in the "Sessions" dialog that you can launch from the "Preferences" menu on some systems. This will automatically launch Orca when you log in if you add `orca` as a startup program.

When you run the `orca` command from a terminal window, you have the following options:

Usage: `orca [OPTION...]`

<code>-?, --help</code>	Show this help message
<code>-v, --version</code>	0.9.0
<code>-s, --setup, --gui-setup</code>	Set up user preferences
<code>-t, --text-setup</code>	Set up user preferences (text version)
<code>-n, --no-setup</code>	Skip set up of user preferences

If Orca has not been previously set up by the user, Orca will automatically launch the preferences set up unless the `-n` or `--no-setup` option is used.

Report bugs to orca-list@gnome.org.

If you have not run Orca before, it will automatically query you for preferences, such as whether you want to use speech, braille, or magnification. These preferences are saved in `~/.orca/user-settings.py` (see material in the following sections for more information).

NOTE: The GNOME accessibility environment requires accessibility to be enabled for your login session, which is not the default configuration. The first time you run Orca, Orca will detect this and will enable accessibility for you. In order for this to take effect, however, you still need to log out and log back in.

To get help while running Orca, you can press `Insert+F1` to enter learn mode. In learn mode, you can type any key combination and Orca will tell you the effects of that key combination. To exit learn mode, press `Escape`.

To quit Orca, you can do any of the following:

- Press `Insert+Q`
- Enter `killall -TERM orca` in a terminal window. *NOTE:* if you enter `killall -HUP orca` in a terminal window, Orca will restart.

NOTE: the main shell script to start Orca, `/usr/bin/orca`, will detect failures in Orca and restart it if necessary. In addition, it will stop any existing Orca-related processes before starting Orca again. As such, if something bad seems to have happened with Orca, you can force it to clean up and restart by merely launching Orca again via any of the methods listed previously.

Chapter 3. Customizing Orca

To be written. Include verbosity, speech rate, braille, braille monitor, orca modifier keys, voices, key/word echo, read table line, etc.

To configure Orca, you can do any of the following:

- Run `orca --setup`
- Press `Insert+Space` while Orca is running

When you configure Orca, it will create `~/.orca/user-settings.py`. You can edit this file using a text editor, but be aware that it will be overwritten the next time you configure Orca. If you wish to have more persistent settings for Orca, you can create `~/.orca/orca-customizations.py`. If this file exists (you need to create it if you want one), `~/.orca/user-settings.py` will import it each time the settings are reloaded. You can force Orca to reload the settings by pressing `Ctrl+Insert+Space`.

The contents of a typical `~/.orca/user-settings.py` look similar to the following:

```
# user-settings.py - custom Orca settings
# Generated by orca. DO NOT EDIT THIS FILE!!!
# If you want permanent customizations that will not
# be overwritten, edit orca-customizations.py.
#
import re
import time

import orca.debug
import orca.settings
import orca.acss

#orca.debug.debugLevel = orca.debug.LEVEL_OFF
orca.debug.debugLevel = orca.debug.LEVEL_SEVERE
#orca.debug.debugLevel = orca.debug.LEVEL_WARNING
#orca.debug.debugLevel = orca.debug.LEVEL_INFO
#orca.debug.debugLevel = orca.debug.LEVEL_CONFIGURATION
#orca.debug.debugLevel = orca.debug.LEVEL_FINE
#orca.debug.debugLevel = orca.debug.LEVEL_FINER
#orca.debug.debugLevel = orca.debug.LEVEL_FINEST
#orca.debug.debugLevel = orca.debug.LEVEL_ALL

#orca.debug.eventDebugLevel = orca.debug.LEVEL_OFF
#orca.debug.eventDebugFilter = None
#orca.debug.eventDebugFilter = re.compile('[\S]*focus|[\S]*activ')
#orca.debug.eventDebugFilter = re.compile('nomatch')
#orca.debug.eventDebugFilter = re.compile('[\S]*:accessible-name')

#orca.debug.debugFile = open(time.strftime('debug-%Y-%m-%d-%H:%M:%S.out'), 'w', 0)
#orca.debug.debugFile = open('debug.out', 'w', 0)
```



```

#orca.settings.useBonoboMain=False
#orca.settings.debugEventQueue=True
#orca.settings.gilSleepTime=0

if False:
    import sys
    import orca.util
    sys.settrace(orca.util.traceit)
    orca.debug.debugLevel = orca.debug.LEVEL_ALL

orca.settings.orcaModifierKeys = ['Insert', 'KP_Insert']
orca.settings.enableSpeech = True
orca.settings.speechServerFactory = 'orca.gnomespeechfactory'
orca.settings.speechServerInfo = ['Fonix DECTalk GNOME Speech Driver', 'OAFIID:GNOME_Speech']
orca.settings.voices = {
    'default' : orca.acss.ACSS({'average-pitch': 5.0,
    'family': {'locale': 'english', 'gender': None, 'name': 'Paul'},
    'gain': 9,
    'rate': 85.0}),
    'uppercase' : orca.acss.ACSS({'average-pitch': 6}),
    'hyperlink' : orca.acss.ACSS({'average-pitch': 2}),
}
orca.settings.speechVerbosityLevel = orca.settings.VERBOSITY_LEVEL_VERBOSE
orca.settings.readTableCellRow = True
orca.settings.enableSpeechIndentation = False
orca.settings.enableEchoByWord = True
orca.settings.enableKeyEcho = True
orca.settings.enablePrintableKeys = False
orca.settings.enableModifierKeys = False
orca.settings.enableLockingKeys = True
orca.settings.enableFunctionKeys = False
orca.settings.enableActionKeys = False
orca.settings.enableBraille = False
orca.settings.enableBrailleGrouping = False
orca.settings.brailleVerbosityLevel = orca.settings.VERBOSITY_LEVEL_VERBOSE
orca.settings.brailleRolenameStyle = orca.settings.BRAILLE_ROLENAME_STYLE_LONG
orca.settings.enableBrailleMonitor = False
orca.settings.enableMagnifier = False
orca.settings.enableMagCursor = True
orca.settings.enableMagCursorExplicitSize = False
orca.settings.magCursorSize = 32
orca.settings.magCursorColor = '#000000'
orca.settings.enableMagCrossHair = True
orca.settings.enableMagCrossHairClip = False
orca.settings.magCrossHairSize = 16
orca.settings.magZoomerLeft = 512
orca.settings.magZoomerRight = 1014
orca.settings.magZoomerTop = 0
orca.settings.magZoomerBottom = 758
orca.settings.magZoomFactor = 4
orca.settings.enableMagZoomerColorInversion = False
orca.settings.magSmoothingMode = orca.settings.MAG_SMOOTHING_MODE_BILINEAR

```

```
orca.settings.magMouseTrackingMode = orca.settings.MAG_MOUSE_TRACKING_MODE_CENTERED
orca.settings.verbalizePunctuationStyle = orca.settings.PUNCTUATION_STYLE_MOST

try:
    __import__("orca-customizations")
except ImportError:
    pass
```

3.1. `orcaModifierKeys` - Override "Insert" as Orca Modifier Key

Orca defines keystrokes for the keypad and also makes special use of the `Insert` key as the Orca modifier key. Orca also provides an `orcaModifierKeys` customization setting to allow the user to override `Insert` as the Orca modifier key. This setting contains a list of key strings that represent the key to be used as the Orca modifier key. For example, the following example specifies that the keyboard `Insert` key (typically near the `Delete` key) and the keypad `Insert` key (typically the `0` key) are to both act as the Orca modifier key:

```
orca.settings.orcaModifierKeys = ["Insert", "KP_Insert"]
```

You can override this setting in your `~/.orca/orca-customizations.py` module if you wish.