

Advanced Bundle Management Concepts with ZCM 11 Lecture

ZEN16

Novell Training Services

www.novell.com

ATT LIVE 2012 LAS VEGAS

Novell[®]

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2012 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the [Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Advanced Bundle Management Concepts with ZENworks Configuration Management 11

Jason Blackett

jblackett@novell.com

Novell[®]

Agenda

- Review key bundle concepts
- Change Management
- System Requirements
- Bundle Ordering (11.2.1)
- Windows Bundle Management
 - Credential Vault Access

Bundle Concepts

- Bundle

- A bundle is an object that consists of:
 - Summary meta data
 - Versioned Action Sets
 - Actions
 - Both bundles and actions link to the content system
 - Content status is based on all content associated with a given bundle existing on a given server
 - Other linked settings

Bundle Concepts

- Action Set

- An action set is a container for 1-*n* actions that the bundle needs to perform
- Actions sets are versioned
 - The default behavior of the Install action set is to only execute if the version on the current action set is higher than the one currently installed
- Action sets are processed in the following order:
 - Distribution (can be scheduled and can initiate install)
 - Install
 - Launch (can be scheduled and will initiate install if not previously installed)
 - Verify (if launch fails or manual verification is initiated)
 - Termination (when availability schedule expires on Windows)
 - Uninstall (when user initiates uninstall or app expires)

Bundle Concepts

- Action

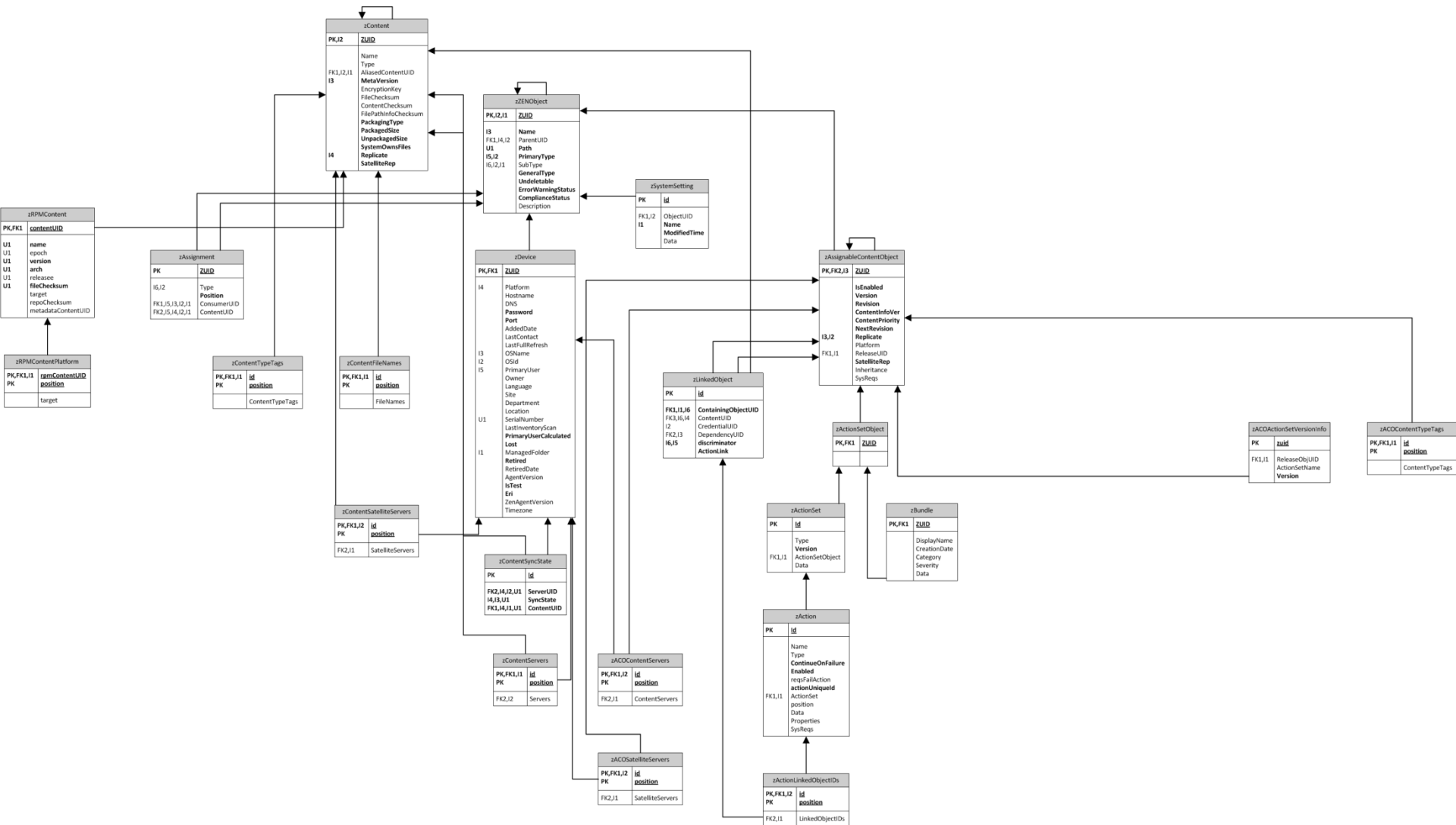
- An action is the smallest amount of work that can be performed by a bundle
- Actions are processed in the order listed in a given action set
- Each action can be flagged continue on failure if appropriate

- Content

- A file that has been uploaded into the ZENworks content repository
- The file may be compressed or encrypted depending on the type of content
- Content records in the DB keep track of the files a given bundle references

Database Schema for Bundles

(Subject to Change – DO NOT MODIFY)



Change Management

- ZENworks 11 and later include bundle and policy change management capabilities
- Enforces rigid change control, so that random changes are not deployed in your production environment
- Exposes the Author vs Publisher right so that you can control who can impact production bundles and who cannot

Demo of Change Management UI

What's really happening behind the scenes?

- When you create the sandbox bundle, a zZENObject, zAssignableContentObject and zBundle entry is created
- zAssignableContent version is initialized to -1
- Changes to sandboxes cause the version to decrement (-1, -2, -3 ...)
- When you publish the sandbox it becomes the “Released” bundle and the version becomes the next positive integer
- If an older version existed then the ReleaseUID attribute of the zAssignableContentObject is updated to reflect the new bundle

Change Management Best Practices

- Don't forget to set the devices/users you want to test with as "test" devices/users and make sure those are the only ones you flag as "test"
- Pick a broad enough set of test devices that you can validate your changes
- Periodically clean-up bundles if you don't require infinite history
 - Deletes the older bundles and any content no longer referenced by the current bundle
- Ensure only users that you want to be able to Publish new versions of a bundle have Publish rights
- If you are migrating from ZENworks 7 to ZENworks 11, use the Sandbox migration option

System Requirements

- Bundle system requirements allow you to evaluate conditions on the managed devices before displaying/executing a bundle
 - The more system requirements you have the longer your refresh will take, so make sure you use them appropriately
- Action system requirements allow you to evaluate conditions on the managed device prior to the action set being executed
 - The more system requirements you have the more logic has to be processed during execution, however this may be offset by fewer actions being executed

System Requirements for Custom Patches

- If you have ZENworks Configuration Management and ZENworks Patch Management you can do the following:
 - Define bundle system requirements
 - Used to identify “applicability” of the patch
 - Promote the bundle to a patch
 - Initiate a subscription update to rebuild DAU bundles
 - Updates BundleDetectFile.xml in the DAU bundle
 - Perform DAU
 - Reports back patched/not patched status for any device that meets the system requirements
 - Uses the bundle install status as the patched identifier

Impersonation

- Many actions within bundles provide the following impersonation levels
 - Normal
 - Executes as the logged in user to both the network and local machine
 - Launch MSI action provides the ability to temporarily elevate
 - Dynamic Administrative User
 - In most actions post ZENworks 11SP1 you can specify a credential object that should be used when accessing network resources
 - Script action does not yet have this as there are no variables currently to retrieve credential object data
 - On Windows 7 if the application wants to display a message this will give you an ISD prompt
 - When executing software from a UNC you may need the `SEE_MASK_NOZONECHECKS=1` variable set to suppress the file security dialog
 - System User / root
 - Has access as SYSTEM / root on the local PC and no network rights unless in a domain, in a domain it has the rights of the Computer object in the domain

Chaining

- Bundle is accomplished via actions in bundles
 - Install Bundle action
 - Executes the Install action set of the selected bundle (assuming the Install action set has a higher version or the referenced bundle is flagged to execute the install action set always)
 - Launch Bundle action
 - Executes the Launch action set of the selected bundle
 - Uninstall Bundle action
 - Executes the Uninstall action set of the selected bundle
 - Verify Bundle action
 - Executes the Verify action set of the selected bundle
- If you want to check if a dependent apps has a new install action set, and execute it on each bundle launch put the Install bundle action in the Launch action set, not the Install action set

Ordering (returning soon)

- ZENworks 11.2.1 will “re-introduce” bundle ordering
- Bundle ordering ensures that if several applications are configured to be executed simultaneously that they are executed in the specified order
- Wait on force run ensures that the app2 install action set doesn't execute until app1's install action set completes

Content Enhancements

- With ZENworks 11SP1 and higher
 - Install directory action now has two new options
 - Compress and encrypt content
 - This allows you to disable the encryption and compression of the individual files that are uploaded into the system
 - Upload as single content
 - Compresses the directory to a .zip file
 - This means that only a single entry has to be written to the content db

Assignment Blocking

- When troubleshooting a problem with an application on a specific device you can now “block” an assignment
 - Can block / unblock either direct or inherited assignments
 - When an application is blocked, it is by default just made “unavailable”
 - Optionally, the creator of the bundle may set the Uninstall when blocked flag

Best Practices

- Use change management to ensure that bundle changes are well tested before deploying them
- Avoid scheduling applications to launch immediately on refresh/login as much as possible
 - This will result in faster login times
- For applications with large content
 - Use UNC content source to avoid caching the install files OR
 - At least use the Upload as single content if many files are required
- Document your bundles via Administrator Notes so that you can see what happened

Best Practices

- Use bundle chaining for real dependencies
- Use bundle ordering for controlling install order
- Create application specific users for UNC based apps and store them in the credential vault when using DAU impersonation
- Use action level system requirements to increase flexibility and reduce number of bundles you need to maintain

Real World Example

- Microsoft Office 2010
 - Best installed from a UNC source
 - Performance as well as disk space usage on the agent
 - Best installed using Microsoft setup program
 - Use the content repo to distribute the administrative setup file
 - Use DAU impersonation to execute the install
 - Use the `SEE_MASK_NOZONECHECKS` to prevent ISD
 - Use custom actions for Install, Verify and Uninstall
 - Chain launcher applications to install application

Lab Exercises

- Exercise : Deploy Office 2010 with ZCM11SP2

Appendix

Linux Bundle Management

Linux Bundle Management Concepts

- Linux Bundle
 - A bundle that is used to deploy changes to the managed device
 - Made up of Action Sets, Actions and Content
- Linux Dependency Bundle
 - A bundle that is used for dependency resolution
 - Optionally can be used for manual package installation
 - Contains only packages
- YUM Repository
 - A standard Linux repository type that provides advanced dependency resolution

Subscriptions

- A subscription is a connection to an external repository
 - Supported repo types: YUM, YAST, ZLM, RHN, Static
- When ZENworks subscribes to a repository the following objects are created during execution
 - A subscription folder
 - Stores all of the bundles related to that subscription
 - 1-*n* Linux or Linux Dependency bundles
 - Bundle types are based on subscription configuration
 - Bundle groups
 - Optionally created as defined in the subscription configuration
- Bundles created by subscription can be optionally created as Sandbox bundles to allow for testing and certification

Install RPM Package Actions

- If you need to install one or more packages use an RPM Package action
- When creating bundle you can select multiple platforms or “All”
 - This ends up creating multiple platform specific actions in the bundle
- Full RPM meta data is stored in an XML file associated with each bundle
 - Agents retrieve all content metadata XML files associated with the device on refresh
- Key RPM metadata is stored in the zRPMContent table of the database
 - Useful for creating reports
- When bundle is distributed all RPMs are cached
- When bundle is transacted the required RPMs are installed

Exporting a bundle as a YUM Repo

- If you have “unmanaged” devices that you want to extend ZENworks content to
 - Make sure that the bundle you want to export is published
 - Go to the details of a bundle or bundle group
 - Choose to create a YUM repository
 - Select one or more primary servers
 - If necessary you can digitally sign the YUM repo using the directions described here:
 - <http://www.novell.com/communities/node/13593/how-digitally-sign-yum-repository-created-zcm11>
 - Now add the YUM repository to the unmanaged device to install packages from the repository
 - Remember that only Packages are a part of a YUM repo, not any other actions

Best Practices

- To avoid duplicating packages you can
 - Export a bundle as a YUM repo
 - Use a policy to assign an exist YUM/ZYPP repository to managed devices for dependency resolution
- Set the package platform at bundle creation time if a package is platform specific
 - Allows better reporting
 - Ensures the package only gets deployed to appropriate operating systems
- Use the cool solution to sign repositories
 - Novell is investigating productizing this in a future release so that repos are signed with a zone certificate

Macintosh Bundle Management

Macintosh Bundles

- Macintosh bundles allow you to deploy software to your Intel based Mac OS-X devices
- Key information for deploying software with Mac Bundles
 - To mount a dmg silently
 - `/usr/bin/hdiutil attach <path to dmg> -readonly -private -noverify -nobrowse`
 - To install a Mac package
 - `/usr/sbin/installer -pkg <path to pkg.mpkg> -target /`
 - To launch a Mac app in the foreground
 - `/usr/bin/open -a <path to .app>`
 - To install a .app
 - Transfer it to a Windows box or network share and then use an Install Directory action

Enabling User Based Access to Bundles on Linux and Mac

Cool Solution for User Based Access to Xplat bundles

- This solution requires that the device be able to talk LDAP directly to the directory server
- Xplat agent on Linux and Mac checks for only Device assignments
- Xplat agent looks for `~/.zmd/zengroups` and `zendn` on each refresh and adds the content of these files to a system variable
- Using the `Idapsearch` tool included with Mac and Linux you can populate these files with the group memberships and dn of the logged in user
- Now on your applications you can use an Environment Variable Value system requirement to make the bundle only appear when `zengroup` or `zendn` contains a specific value

Novell®

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Novell, Inc. may make improvements in or changes to the software described in this document at any time.

Copyright © 2011 Novell, Inc. All rights reserved.

All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

