Cherwell GLOBAL CONFERENCE 2016

Innovation with Purpose.

Develop

Consultant, Excalibur Data Systems, Inc.

Jeff Jones

Automating Cherwell User, Team and Team Membership Creation From Active Directory

Automating Cherwell User, Team and Team Membership Creation From Active Directory

Jeff Jones of Excalibur Data Systems is a veteran of multiple ITSM toolsets, just about every programming language known to man and way too much digital technology. He has spent the past 25+ years working with customers to improve their processes, integrate their data and when needed code things into submission. He has been working with Cherwell Service Management since its earliest days on the market and is a Cherwell consultant and instructor. When he isn't working with a customer to make their Cherwell System leap tall building in a single bound, you'll probably find him learning yet another programming language or elbows deep in a new technology he wants to add to his tool bag.

Agenda

- Welcome & Introduction
- Potential Automation Methods
- Users & Teams in Cherwell
  - Cherwell Objects & Backend Tables
  - How They Fit Together
- A Deeper Look
  - Teams
  - Users & Authentications
  - Team Memberships
  - Security Groups
- Automating It
  - Active Directory: The Source of Truth
  - Preparing the Cherwell System
  - Avoiding Conflicts with Manually Entered Data

- Processing the Import
- I Want This
- Q&A
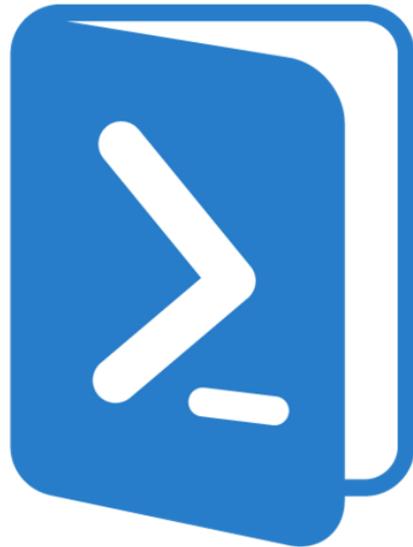
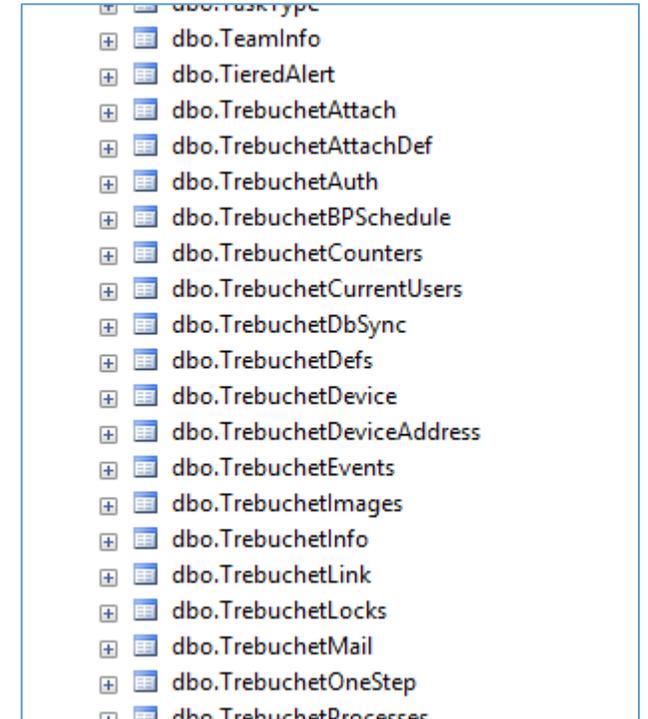# Potential Automation Methods

- SQL Scripting/Triggers
- SQL Service Integration Services
- Web Services
- PowerShell
- One-Steps

- Our Approach
  - Use Cherwell's Web Services via PowerShell to get the data staged into Cherwell Objects
  - Use One-Steps to create the appropriate records for Teams, Users and Team Memberships
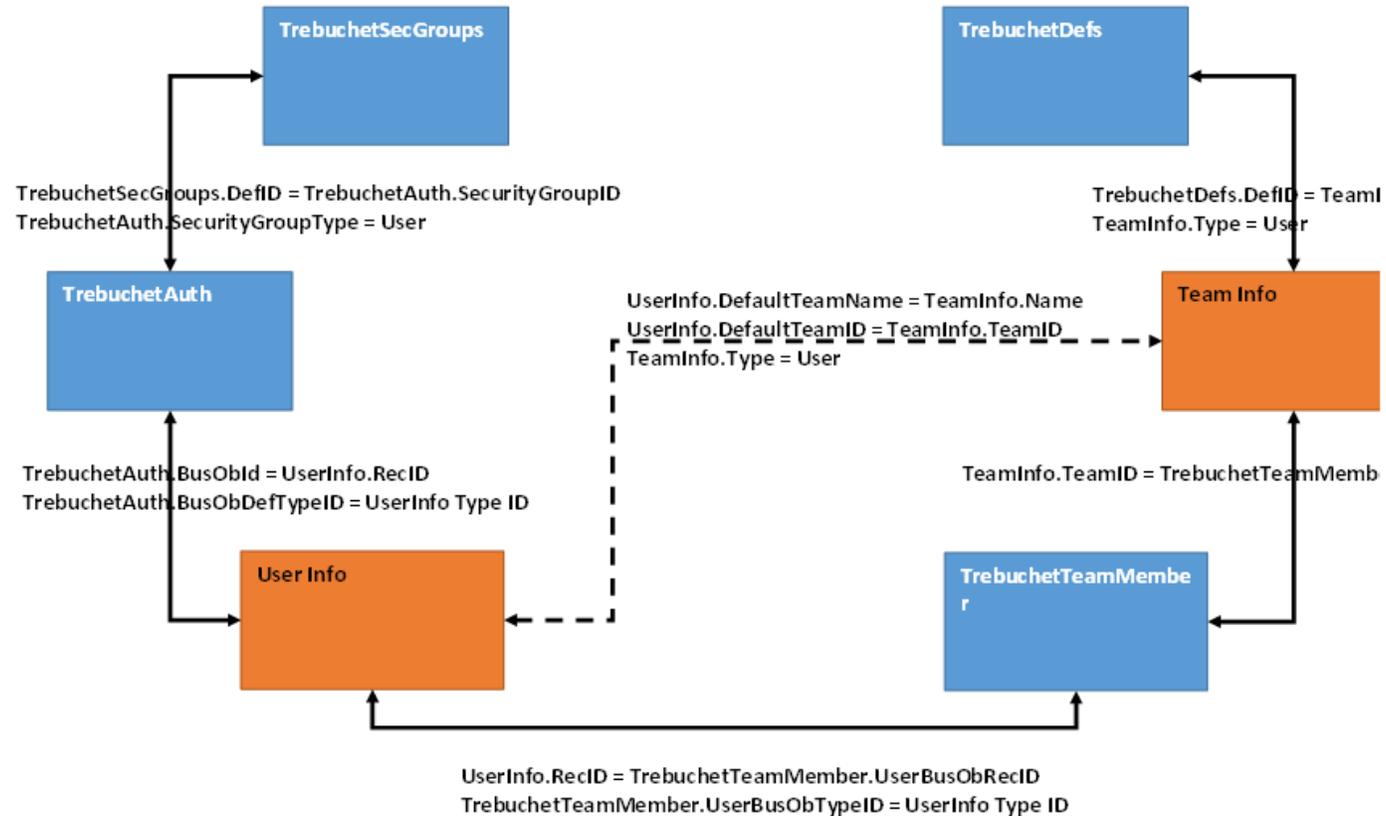 - Wrap this all up in a single One-Step called via the Cherwell Scheduler to perform the entire process

Users & Teams in Cherwell

- **Cherwell Backend Tables**
  - TrebuchetSecGroups – Holds security group definitions
  - TrebuchetDefs – Holds team definitions
  - TrebuchetAuth – Holds user and customer authentication records
  - TrebuchetTeamMember – Holds user and customer team membership information
- **Cherwell Objects**
  - User Info – Holds the client accessible information about a user
  - Team Info – Holds the client accessible information about a team, v6.0 and later only

dbo.TaskType
dbo.TeamInfo
dbo.TieredAlert
dbo.TrebuchetAttach
dbo.TrebuchetAttachDef
dbo.TrebuchetAuth
dbo.TrebuchetBPSchedule
dbo.TrebuchetCounters
dbo.TrebuchetCurrentUsers
dbo.TrebuchetDbSync
dbo.TrebuchetDefs
dbo.TrebuchetDevice
dbo.TrebuchetDeviceAddress
dbo.TrebuchetEvents
dbo.TrebuchetImages
dbo.TrebuchetInfo
dbo.TrebuchetLink
dbo.TrebuchetLocks
dbo.TrebuchetMail
dbo.TrebuchetOneStep
dbo.TrebuchetProcesses

Users & Teams in Cherwell

How They Fit Together

A Deeper Look - Teams

TrebuchetDefs

TrebuchetDefs contains the definition of almost every object the user interacts with in Cherwell, one of these is the definition of a team. The definition is stored as XML in the DefDetails field and looks like this:

```
<Trebuchet>
  <TeamDef ID="93d5f985f19b1e6dbb50f24942866662053c13e88f"
          Name="IT Management" Version="1.0" SubType=""
          Scope="Core" Culture="Invariant" View="(None)"
          TeamType="User">
    <Alias />
    <Description />
  </TeamDef>
</Trebuchet>
```

TeamInfo

Fortunately with v6.0 Cherwell introduced the TeamInfo object, which is directly tied to the backend definition. There are several interesting fields available for our use:

TeamID – This contains the ID of the team that Cherwell uses internally
Name – This contains the Name of the team
Description – This contains a description of the team
Image – This points to an image displayed for the team
Type – This contains the team type , either User or CustomerWorkgroup
EmailAlias – Contains the email address to send messages to

To create a new team you just create a TeamInfo record filling in the Name and Type and Cherwell will create the definition for you.

A Deeper Look – Users & Authentications

Users are split into two parts in the Cherwell backend, the authentication details stored in TrebuchetAuth and the User details stored in UserInfo. To correctly create a user both of these pieces must be created.

Starting with UserInfo, these are some important fields:

RecID – Contains the system generated record id.
Full Name (and associated name fields) – Contains the user's name
Default Team Name – contains the name the user's default team
Default Team ID – contains the ID of the above
SAMAccountName – contains the user's network ID
System_LDAPPath – contains the user's LDAP distinguished name

After the UserInfo record is constructed we need to create an authentication record in the TrebuchetAuth table. Here are a few important fields to set here:

WindowsIdentifier – Contains the user's AD login id (domain\username)
SecurityGroupType – Will always be set to User
SecurityGroupID – Contains the DefID of the user's security group
BusObDefTypeID – Contains the DefID of the UserInfo table
BusObID – Contains the RecID of the user's UserInfo record

Note that we can only create AD/Windows accounts as we don't know the hash function to create passwords for internal accounts, although with v8's REST API there is now a web service call to set the password.

A Deeper Look – Team Memberships

Team Memberships in Cherwell are implemented as a simple join table that relates the UserInfo object to the TeamInfo (or more specifically the TrebuchetDefs) object in a many to many arrangement, with the addition of a single field to indicate if the user is a manger of the team or not. Here is what we need to set in the table:

TeamDefID – Contains the team's TeamID from the TeamInfo table
UserBusObRecID – Contains the user's RecID from the UserInfo table
UserBusObTypeID – Contains the ID for the UserInfo object
TeamMgr – Contains 1 if the user is a team manger, 0 otherwise
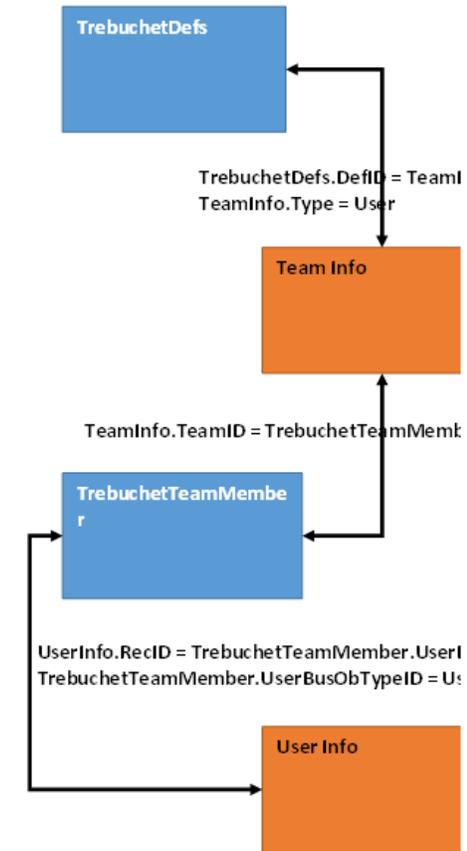
A Deeper Look – Security Groups

Security Groups are stored separately from all other definitions in Cherwell's backend in a table named TrebuchetSecGroups. Similar to the team definitions stored in the TrebuchetDefs table the actual definition of a security group is stored as XML, but, unlike the simple XML of a team definition, the security group definition is extremely complicated and intertwined with other definitions.

Creating Security Groups outside of the standard Cherwell Administrator tool is not recommended and will not be covered in this session.

```
<Trebuchet>
<SecurityGroup ID="93373402da59635ba9e6164cf89364774a3a2bebb1"
Name="Admin" Version="1.0" SubType="" Scope="Core"
Culture="Invariant" View="(None)" Disabled="FALSE"
DontAllowToRemove="TRUE" Admin="TRUE"
DefaultSecurityGroup="FALSE" SecurityGroupType="User">
<LastModBy>CSDAdmin</LastModBy>
<LastModDateTime>2015-12-03T17:54:32</LastModDateTime>
<Alias />
<Description>Security group for system administrators.  Typically, these
Cherwell users can delete groups of records.</Description>
<StandardRightList DefaultRight="Yes,View,Add,Edit,Delete">
  <StandardRight Name="Queues.SuspendQueueItems.Global"
Right="Yes,View,Add,Edit,Delete" />
  <StandardRight Name="OneStep.CanSetToRunUnderAnyGroup"
Right="Yes" />
  <StandardRight Name="Dashboards.Dashboards.Global"
Right="Yes,View,Add,Edit,Delete" />
  <StandardRight Name="Admin.SystemBlueprint.FieldDetails"
Right="Yes,View,Add,Edit,Delete" />
  <StandardRight Name="Reports.ReportStyles.Global"
Right="Yes,View,Add,Edit,Delete" />
…
```

Automating It – Active Directory: The Source of Truth

- Mapping Active Directory Objects to Cherwell Objects
  - Users are Users
  - Teams are AD Security Groups
    - Which Groups to pull in?
  - Team Memberships
    - Limitations with the built in LDAP import
  - Which Team is the Default?
    - Name based?
    - AD attribute based?
    - The first/last found?
   - Or just ignore it?
  - Cherwell Security Groups

Commonly Used AD Convention
- User OU
  - John Smith
- Security Group OU
  - CSM_SG_Admin
  - CSM_DEF_TEAM_ITSM Support
  - CSM_TEAM_ITSM Support
  - CSM_TEAM_IT Management
  - CSM_TEAM_ITSM Support
  - …

Automating It – Preparing the Cherwell System

- Staging Tables
  - Exposing Backend Tables
    - External Data Connection
    - Exposing TrebuchetAuth
    - Exposing TrebuchetTeamMembers
  - ImportedUser
    - Related To UserInfo
    - Related To TrebuchetAuth
  - ImportedTeam
    - Related To TeamInfo
  - ImportedTeamMembership
    - Related To TeamMemberMapping
  - TeamMemberMapping
    - Related To UserInfo
    - Related To TeamInfo
    - Related To TrebuchetTeamMembers
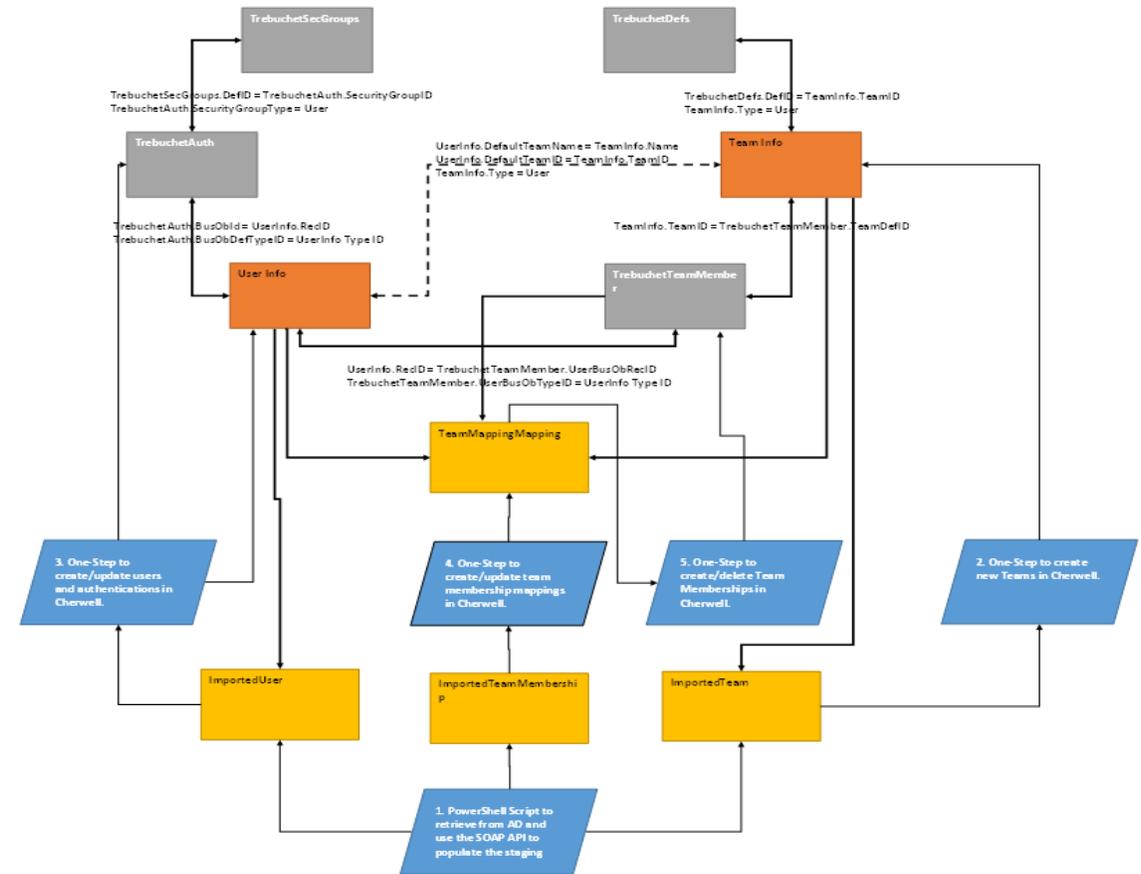
Automating It – Avoiding Conflicts with Manually Entered Data

Invariably some users, teams and team memberships will need to be manually created and we need to make sure that our automated import doesn't touch these items, and doesn't duplicate these items. To facilitate this we add a marker field to the items which are automatically imported for the User and Team items and avoid those items if the marker is not set. For team memberships the Team Mapping structure is used to facilitate this since we cannot change the structure of the TrebuchetTeamMembers table itself.
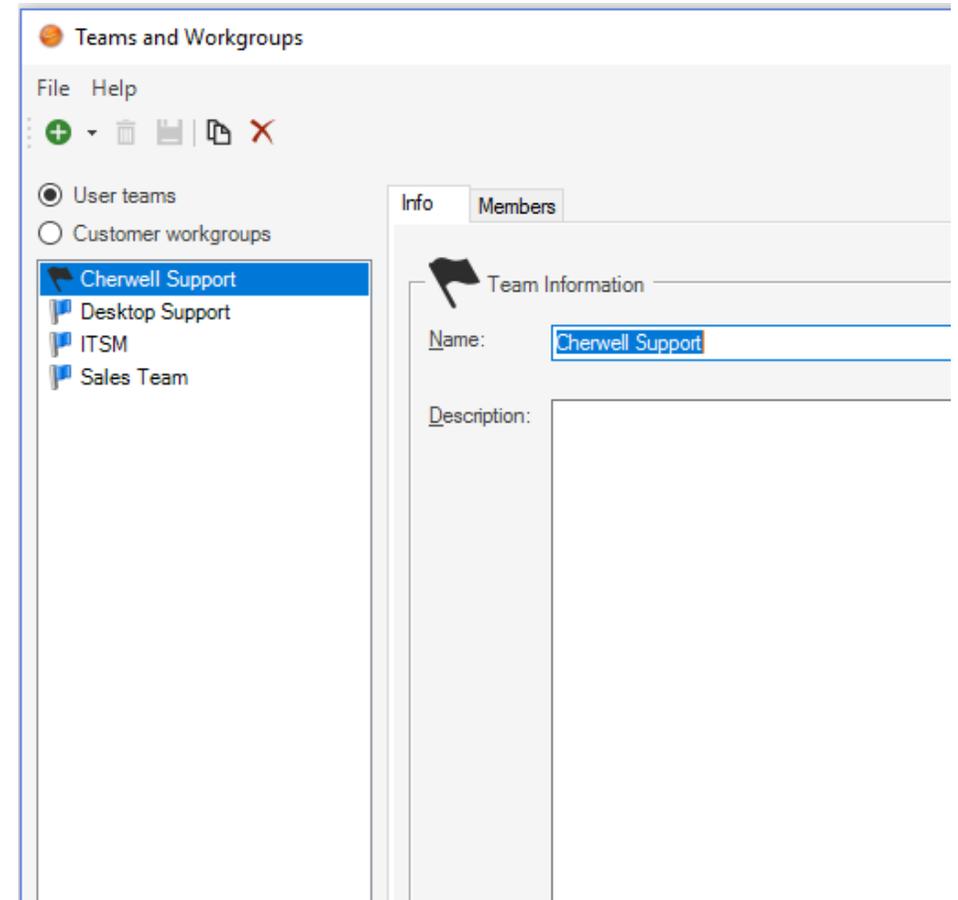
Automating It – Preparing the Cherwell System

- How It Fits Together
  1. Import the Data via PowerShell
     - A. Connect To SOAP API
     - B. Read AD & Create User Objects
     - C. Read AD & Team Objects, if:
       - i. Security Group: update User Objects
       - ii. Default Team: update User Objects
       - iii. CSM Team: create Team Object
       - iv. CSM Team: Add Membership Objects
  2. Create TeamInfos
  3. Create UserInfos/TrebuchetAuths
  4. Create Team Member Mappings
  5. Create Team Memberships
  6. Clean Up

```powershell
81  Try {
82      # Connect to Cherwell
83      $csm = New-WebServiceProxy -Uri $cherwellservice
84      $csm.CookieContainer = New-Object System.Net.CookieContainer
85      if ($csm.Login($cherwelluser, $cherwellpassword))
86      {
87          $root = [adsi]"LDAP://$ldapServer/$searchBase"
88          $userRoot = [adsi]"LDAP://$ldapServer/$userBase"
89
90          # Get the groups from Active Directory which match our filter for
91          $searcher = ""
92          $groups = ""
93          $group = ""
94          $searcher = [adsisearcher]"$secgroupfilter"
95          $searcher.SearchRoot = $root
96          $searcher.PropertiesToLoad.AddRange(( 'cn', 'member' ))
97          $groups = $searcher. FindAll()
98          if ($groups.Count -gt 0) {
99              ForEach ($group in $groups) {
100                 # Get the members of the group
101                 $sgName = ""
102                 $members = ""
103                 $member = ""
104                 $sgName = $group.Properties['cn']
105                 $members = $group.Properties['member']
106                 if ($members.Count -gt 0) {
107                     ForEach ($member in $members) {
108                         # Lookup each member in Active Directory
109                         $usersearcher = ""
110                         $aduser = ""
111                         $usersearcher = [adsisearcher]"(&(distinguishedNa
112                         $usersearcher.SearchRoot = $userRoot
```

Cherwell GLOBAL CONFERENCE 2016

Innovation with Purpose.

I Want This

At the end of the Cherwell Global Conference 2016 a mApp will be available in the mApp Exchange on the community site with a basic implementation of this method to automate the creation of Users, Teams and Team Memberships in your Cherwell platform.

Any Questions?