

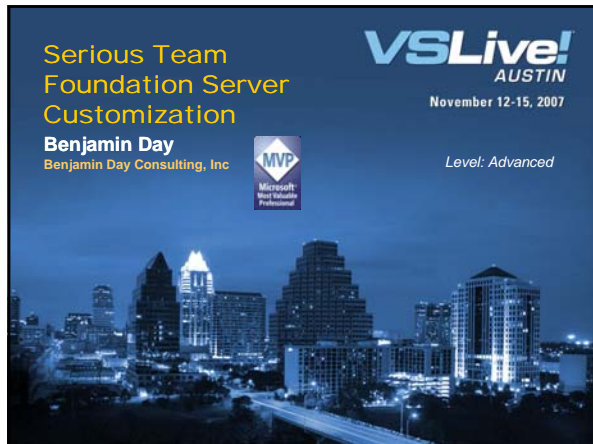
VS Live!
 AUSTIN
 November 12-15, 2007

Level: Advanced

Microsoft MVP
 Microsoft Most Valuable Professional

Benjamin Day
 Benjamin Day Consulting, Inc.

Serious Team Foundation Server Customization



About the speaker

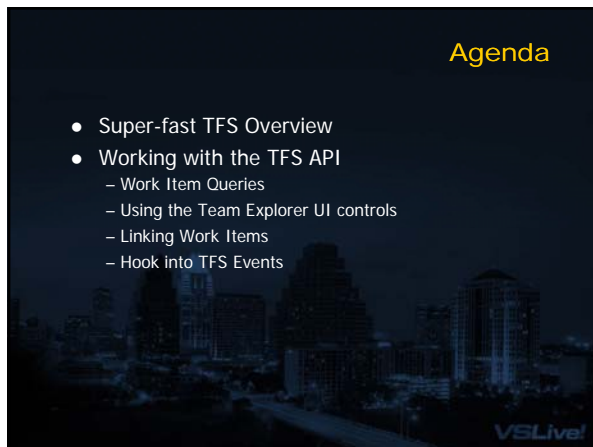


- Owner, Benjamin Day Consulting, Inc.
 - Email: benday@benday.com
 - Web: <http://www.benday.com>
 - Blog: <http://blog.benday.com>
- Trainer
 - Visual Studio Team System, Team Foundation Server
- Microsoft MVP for C#
- Microsoft VSTS/TFS Customer Advisory Council
- Leader of Beantown.NET INETA User Group



Agenda

- Super-fast TFS Overview
- Working with the TFS API
 - Work Item Queries
 - Using the Team Explorer UI controls
 - Linking Work Items
 - Hook into TFS Events



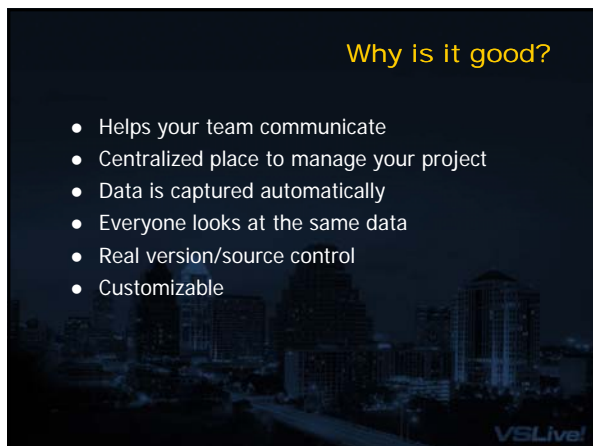
What is Team Foundation Server?

- “Glue” that connects your team
 - What am I supposed to do?
 - What’s my progress?
 - What’s broken?
- Helps manage project artifacts
 - Source Control



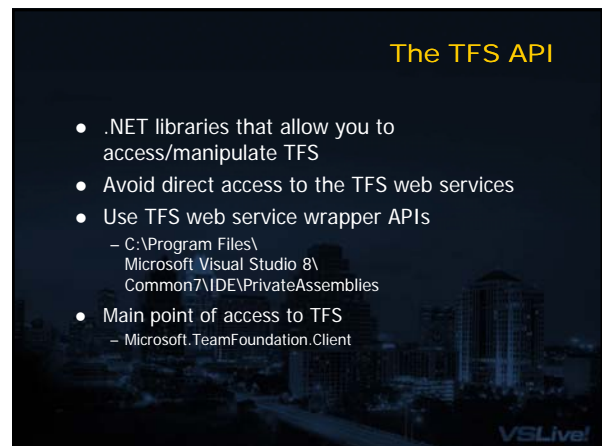
Why is it good?

- Helps your team communicate
- Centralized place to manage your project
- Data is captured automatically
- Everyone looks at the same data
- Real version/source control
- Customizable



The TFS API

- .NET libraries that allow you to access/manipulate TFS
- Avoid direct access to the TFS web services
- Use TFS web service wrapper APIs
 - C:\Program Files\Microsoft Visual Studio 8\Common7\IDE\PrivateAssemblies
- Main point of access to TFS
 - Microsoft.TeamFoundation.Client



Speaker Name



Connect to TFS

- Reference
Microsoft.TeamFoundation.Client.dll
- Namespace =
Microsoft.TeamFoundation.Client

```
TeamFoundationServer server =  
    TeamFoundationServerFactory.GetServer(  
        "http://bendaytfs:8080");  
  
server.Authenticate();
```

Other TFS API DLLs

- Microsoft.TeamFoundation.Build.Client
- Microsoft.TeamFoundation.VersionControl.Client
- Microsoft.TeamFoundation.WorkItemTracking.Client

Access the TFS services

- Call the GetService(Type) method on
TeamFoundationServer instance
– Type is the type of the service API
- Example:

```
- VersionControlServer server =  
    m_server.GetService(  
        typeof(VersionControlServer)) as  
    VersionControlServer;
```

Some Other TFS Service Interfaces

- TFS Services & Configuration Info →
RegistrationProxy
- Work Items → typeof(WorkItemStore)
- Events → typeof(IEventService)
- Links → typeof(ILinking)
- Area & Iteration Info →
typeof(ICommonStructureService)

Projects & Queries

- Get Team Projects
– ProjectCollection projects =
 m_store.Projects;
 Project project = projects[0];
- Get Team Projects
– StoredQueryCollection queries =
 project.StoredQueries;
 StoredQuery query = queries[0];
– QueryScope.Private Or QueryScope.Public;

Run Stored Query

```
StoredQueryCollection queries = project.StoredQueries;  
StoredQuery storedQuery = queries[0];
```

```
WorkItemStore store =  
    server.GetService(typeof(WorkItemStore)) as  
    WorkItemStore;
```

```
Dictionary<string, string> values =  
    new Dictionary<string, string>();
```

```
values.Add("project", projectName);
```

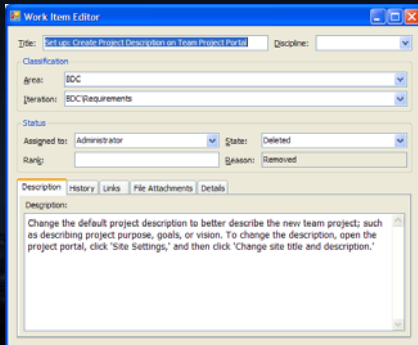
```
Query query = new Query(  
    store, storedQuery.QueryText, values);  
WorkItemCollection workItems =  
    store.Query(query.QueryText);
```

```
WorkItem workItem = workItems[0];
```

Speaker Name



Re-using the Team Explorer UI Controls



UI Controls

- WorkItemResultGrid
 - grid.LoadFromWiql(m_store, query.QueryString);
- WorkItemFormControl
 - control.Item = myWorkItem;
- LinksControl
 - control.WorkItemDatasource = myWorkItem;
 - control.Invalidate();

Code Demo

- Write a "Query Analyzer" for TFS Work Item Queries
- Pop up the detail view of a work item
- Edit and save the work item

Linking work items

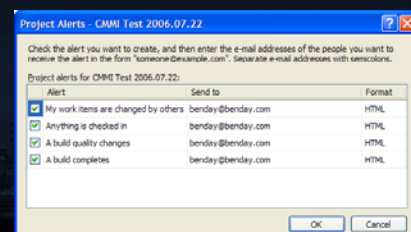
```
WorkItem fromWorkId =  
store.GetWorkItem(282);  
  
fromWorkId.Links.Add(new RelatedLink(283));  
fromWorkId.Links.Add(new RelatedLink(284));  
fromWorkId.Links.Add(new RelatedLink(285));  
  
fromWorkId.Save();
```

Code Demo

- Create a "quick link" interface

TFS Events

- There are more events than you can see in "Project Alerts"



Speaker Name

VSLive!
BOSTON

How do events work?

- Piece of TFS code needs to fire event
- Calls web method on EventService.asmx
 - <http://localhost:8080/services/v1.0/EventService.asmx>
 - FireAsyncEvent() or FireBulkAsyncEvents()
- Events gets temporarily stored in db
- TFS reads the “to do” events queue
- Reads the event subscription configurations
- Processes the events
 - Sends email
 - Calls other web services

TFS events configuration

- Configuration data stored in “TfsIntegration” database
- Get a list of all (well, most) events
 - Select * from tbl_event_type
- “Hidden” events
 - BuildCompletionEvent
 - WorkItemChangedEvent
 - DataChangedEvent
- Get a list of all event subscriptions
 - Select * from tbl_subscription

What does an event look like?

- Event is XML
 - Core info about the event
 - Changed data
- Event Xsd's are stored in
 - C:\Program Files\Microsoft Visual Studio 2005 Team Foundation Server\Web Services\Services\v1.0\Transforms

Event XML, Part 1

```
<WorkItemChangedEvent
  xmlns:xsl="http://www.w3.org/2001/XMLSchema-Instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <PortfolioProject>Yahoo Music Engine PlugIn
</PortfolioProject>
  <ProjectModelId>5e3e6d51-b3c2-4fca-a94d-9b2d14bbd1bc
</ProjectModelId>
  <AreaPath>Yahoo Music Engine PlugIn/AreaPath
<Title>Yahoo Music Engine PlugIn Work Item Changed:
  Bug 160 - Test bugs1</Title>
  <WorkItemTitle>Test bugs1</WorkItemTitle>
  <Subscriber>BENDAYTFS\Administrator</Subscriber>
  <ChangeSID>S-1-5-21-1136986177-3019447445-4132693080-
1021</ChangeSID>
  <DisplayURL>http://bendaytfs:8080/WorkItemTracking/
  WorkItem.aspx?artifactMoniker=160</DisplayURL>
  <TimeZone>Eastern Daylight Time</TimeZone>
  <TimeZoneOffset>-04:00</TimeZoneOffset>
  <ChangeType>Change</ChangeType>
  ...

```

Event XML, Part 2

```
...
<CoreFields>
  <IntegerFields>
    <Field>
      <Name>ID</Name>
    </Field>
  </IntegerFields>
  <ReferenceName>System.ID</ReferenceName>
  <OldValue>160</OldValue>
  <NewValue>160</NewValue>
  </Field>
  </IntegerFields>
  <StringFields>
    <Field>
      <Name>Title</Name>
    </Field>
  </StringFields>
</CoreFields>
<ChangedFields>
  <Field>
    <Name>Title</Name>
    <OldValue>Bug 160 - Test bugs1</OldValue>
    <NewValue>Test bugs1</NewValue>
  </Field>
</ChangedFields>
</WorkItemChangedEvent>

```

Events have subscription filters

- Restrict subscription to events you're actually interested in
- Uses Xpath-like syntax to address data in the event xml
- Example: “My work items are changed by others”

```
"PortfolioProject" = 'CMI Test 2006.07.22'
AND
(
  "CoreFields/StringFields/Field[
  ReferenceName='System.AssignedTo']/OldValue" =
  'Benjamin Day'
OR
  "CoreFields/StringFields/Field[
  ReferenceName='System.AssignedTo']/NewValue" =
  'Benjamin Day'
)
AND
"CoreFields/StringFields/Field[
  ReferenceName='System.AuthorizedAs']/NewValue" <>
  'Benjamin Day'

```

Speaker Name



Event Processing

- Best thought of as “notifications”
- Notification methods
 - Emails
 - Web service calls
- Emails
 - Event Xml + Xsl transform → Email
 - C:\Program Files\Microsoft Visual Studio 2005 Team Foundation Server\Web Services\Services\v1.0\Transforms
- Web Service calls

Learn Notification Events: Write a Tracing Service

- Write a web service that accepts any event
- Writes the event XML to a log file
- Copy the web service code to TFS server
 - C:\Program Files\Microsoft Visual Studio 2005 Team Foundation Server\Web Services
- Subscribe this web service to all events
- Enjoy

Basic event web service

```
[WebService(Namespace =
    "http://schemas.microsoft.com/TeamFoundation/" +
    "2005/06/Services/Notification/03")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class NotificationService : System.Web.Services.WebService
{
    public NotificationService() { }

    [SoapDocumentMethod(
        "http://schemas.microsoft.com/" +
        "TeamFoundation/2005/06/Services/Notification/03/Notify",
        RequestNamespace =
            "http://schemas.microsoft.com/" +
            "TeamFoundation/2005/06/Services/Notification/03")]
    [WebMethod]
    public void Notify(string eventXml, string tfsIdentityXml)
    {
    }
}
```

How to subscribe to events

- BisSubscribe.exe
 - Subscribes & unsubscribes to events
 - C:\Program Files\Microsoft Visual Studio 2005 Team Foundation Server\TF Setup
- Calls EventService.asmx via TFS APIs
- Parameters
 - /eventType = name of the event
 - /deliveryType = what kind? (text email, html email, SOAP)
 - /address = email address or web service URL
 - /server = which TFS instance?
 - /tag = friendly event subscription name

How to unsubscribe from events

- Connect to the TFS database
- Go to TfsIntegration database
- select * from tbl_subscription
- Find your event subscription id
- bissubscribe.exe /unsubscribe <id >

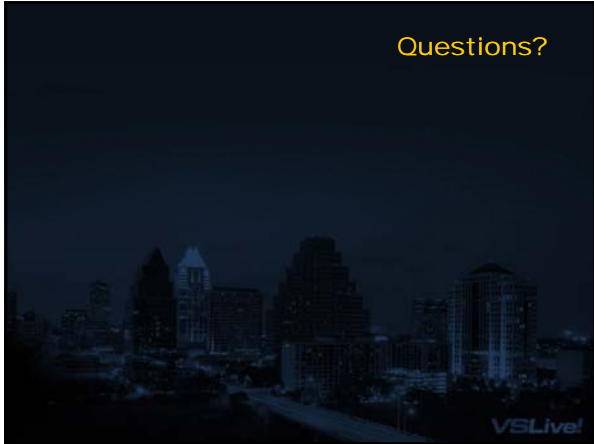
Other Customization Points

- MSBuild
 - Custom tasks
- Static Code Analysis Rules
- Work Item Template Customization
- Reporting Services
 - New reports against the warehouse

Speaker Name

VSLive!
BOSTON

Questions?



Speaker Name

