

Context-sensitive Help Techniques



August 19-20th, Cambridge, MA

Scott DeLoach
Founding Partner
User First Services, Inc.
scott@userfirst.net

We will discuss how to use:

- JavaScript (JS)
- Active Server Pages (ASP)
- eXtensible Markup Language (XML)

to link context-sensitive Help to a Web-based application

- Stewart Walker, Helpful Technology Pty Ltd

Options...

- Hardcoding links vs mapping
- Tool APIs vs custom approaches
- Client-side vs server-side mapping
- One-to-one and one-to-many mapping

Sample App

01/06/2002 You are logged in as: sss

[Home](#) [Tutorial](#) [Help](#)

- Create project
- View projects
- View schedule
- Search

Create Project

Project Number: ?

Description:

Start Date:

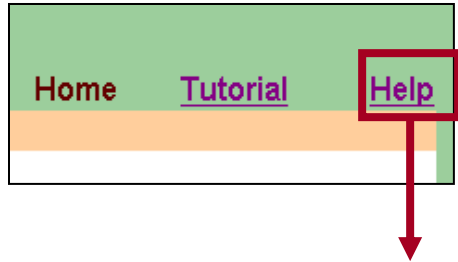
Completion Date:

Requestor:

Priority:

Tasks:

Hardcoding a Help Link

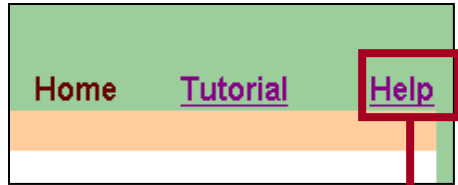


```
<a href="#" onClick="openHelp()">Help</a>
```



```
<script>  
function openHelp() {  
  HelpWin =  
  window.open('help.htm','helpwin','toolbar=0, ↵  
  location=0,directories=0,status=0,menubar=0, ↵  
  scrollbars=1,resizable=1,width=600,height=400');  
  setTimeout('helpWin.focus();',250);}  
</script>
```

Hardcoding a Help Link - a slight improvement



```
<a href="#" onClick="openHelp('thispage.htm')">Help</a>
```

```
<script>  
function openHelp(page) {  
  helpWin =  
  window.open(page,'helpwin','toolbar=0,location=0, ↵  
  directories=0,status=0,menubar=0,scrollbars=1, ↵  
  resizable=1,width=600,height=400');  
  setTimeout('helpWin.focus();',250);  
}  
</script>
```

Pros

- Good for Help link since it doesn't change
- Little work for Help author
- Simple solution for small applications

Cons

- A lot of work for application developer
- Doesn't scale well for large applications

Tool APIs - Pros

- Map file can be automatically created
- Defined process
- Technical support

Tool APIs - Cons

- Requires a tool
- Can be locked into a specific technology
- Can be locked into a process

Custom Approach - Pros

- Completely flexible
- No tool required
- Can choose client-side or server-side mapping

Custom Approach - Cons

- Little to no technical support
- Requires coding knowledge

Client-side Mapping with JS

Sample App Home Tutorial **Help**

01/06/2002 You are logged in as: sss

Create project **Create Project**

View projects

View schedule

Search

Project Number: **Field-level Help**

Description:

Start Date:

Completion Date:

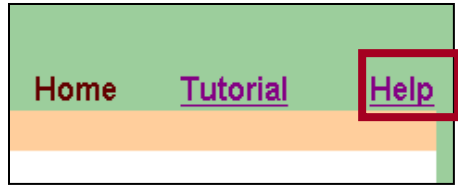
Requestor:

Priority:

Tasks:

Page-level Help

Client-side Mapping with JS for Page-level Help



```
<script>
function openCSHelp() {
  helpurl = location.href;
  begin=helpurl.lastIndexOf('/');
  begin = begin + 1;
  end=helpurl.lastIndexOf('m');
  end=end + 1;
  helpurl = "h_" + helpurl.substring(begin, end);
  helpWin = window.open(helpurl,'CShelpwin', ↵
  'toolbar=0,location=0,directories=0,status=0, ↵
  menubar=0,scrollbars=0,resizable=0, ↵
  width=300,height=200');
  setTimeout('helpWin.focus();',250);
}
</script>
```

Client-side Mapping with JS for Field-level Help

Create Project

Project Number: 

```
<a href="#" onClick="openCSFieldHelp('ProjectNumber')">  
</a>
```

```
<script>  
function openCSFieldHelp(id) {  
  helpurl = "h_" + id + ".htm";  
  helpWin = window.open(helpurl,'CShelpwin','toolbar=0, ↵  
location=0,directories=0,status=0,menubar=0,scrollbars=0, ↵  
resizable=0,width=300,height=200');  
  setTimeout('helpWin.focus();',250);  
}  
</script>
```

Client-side Mapping with JS

- Working with Developers

- Help filenames must match application filenames (h_ + pagename.htm)
- Same code for all page-level Help
- Field Help requires each field to have a name

Client-side Mapping with JS

- Testing

- Avoid "404 Document Not Found" messages
- Watch for application filename changes

Client-side Mapping with ASP

The screenshot displays a web application interface for "Sample App". At the top right, there are navigation links for "Home", "Tutorial", and "Help". The "Help" link is highlighted with a red box and an arrow pointing to it. Below the navigation bar, the date "01/06/2002" and the user information "You are logged in as: sss" are shown. On the left side, there is a vertical menu with options: "Create project", "View projects", "View schedule", and "Search". The main content area is titled "Create Project" and contains a form with the following fields: "Project Number:", "Description:", "Start Date:", "Completion Date:", "Requestor:", "Priority:" (with a dropdown menu set to "1"), and "Tasks:". A "Submit Project" button is located at the bottom of the form. Two red boxes with arrows highlight specific help features: one box around the "Help" link is labeled "Page-level Help", and another box around a question mark icon in the "Description" field is labeled "Field-level Help".

Server-side Mapping with ASP for Field-level Help

The Web Help
JumpStart Conference
August 19-20th, Cambridge, MA

Create Project

Project Number: 

JavaScript Code to Open the Help


```
<a href="#" onClick="openFieldHelp('ProjectNumber')">  
</a>
```

```
<script>  
function openFieldHelp(topic) {  
window.open('fieldhelp.asp?' + topic,'helpwin','toolbar=0,location=0, ↵  
directories=0,status=0,menubar=0,scrollbars=1,resizable=0,width=600, ↵  
height=400');  
}  
</script>
```

Server-side Mapping with ASP for Field-level Help

The Web Help
JumpStart Conference
August 19-20th, Cambridge, MA

Create Project

Project Number: 

VBScript Code to Open the Database

```
Dim objConn  
Set objConn = Server.CreateObject("ADODB.Connection")  
objConn.Open "DRIVER={Microsoft Access Driver (*.mdb)};  
DBQ=" & Server.MapPath("\db\fieldhelp.mdb")
```

```
Dim objRS  
Set objRS = Server.CreateObject("ADODB.Recordset")  
objRS.Open "Tutorial", objConn, , , adCmdTable
```

Server-side Mapping with ASP for Field-level Help

FieldID	FieldLabel	HlpText	HlpTextCustom
ProjectNumber	Project Number	The number	Note: This number is automatically generated.
Description	Description	A brief	
StartDate	Start Date	When the project needs to begin.	
CompletionDate	Completion Date	When the project should end.	
Requestor	Requestor	The person	Requestors must be managers.
Priority	Priority	The importance	
Tasks	Tasks	What should	

- "HlpText" is used to store defaults.
- "HlpTextCustom" is used to store modified Help topics.

Server-side Mapping with ASP for Field-level Help

The Web Help
JumpStart Conference
August 19-20th, Cambridge, MA

Create Project

Project Number:



Code to Pull the Field Help from the Database

```
Do While Not objRS.EOF
If Request.QueryString = objRS("FieldID") Then
If objRS("HlpTextCustom") <> "" Then
    Response.Write "<b>" & objRS("FieldLabel") &
    "</b><br> " & objRS("HlpTextCustom")
Else
    Response.Write "<b>" & objRS("FieldLabel") &
    "</b><br> " & objRS("HlpText")
End If
End If
objRS.MoveNext
Loop
```

Server-side Mapping with ASP

- Working with Developers

- Help filenames can have any name
- Field Help requires Help icon image with appropriate field name to link to topic

Server-side Mapping with ASP

- Testing

- Set up two Help columns in the database: default and custom.
- Make sure that each cell contains some information, even "Help is not available for this field."

One-to-Many Mapping with JS

```
<script>
function openCSHelp() {
  helpurl = location.href;
  begin=helpurl.lastIndexOf('/');
  begin = begin + 1;
  end=helpurl.lastIndexOf('m');
  end=end + 1;
  helpurl = "h_" + helpurl.substring(begin, end);
  helpWin = window.open('help.htm?' + helpurl,'CShelpwin', ↵
  'toolbar=0,location=0,directories=0,status=0, ↵
  menubar=0,scrollbars=0,resizable=0, ↵
  width=300,height=200');
  setTimeout('helpWin.focus();',250);
}
</script>
```

One-to-Many Mapping with JS

```
<script>
if ((location.search).indexOf('print_mod') != -1) {
location.href = 'nohelp.htm';
}
else {
location.href = location.search;
}
</script>
```

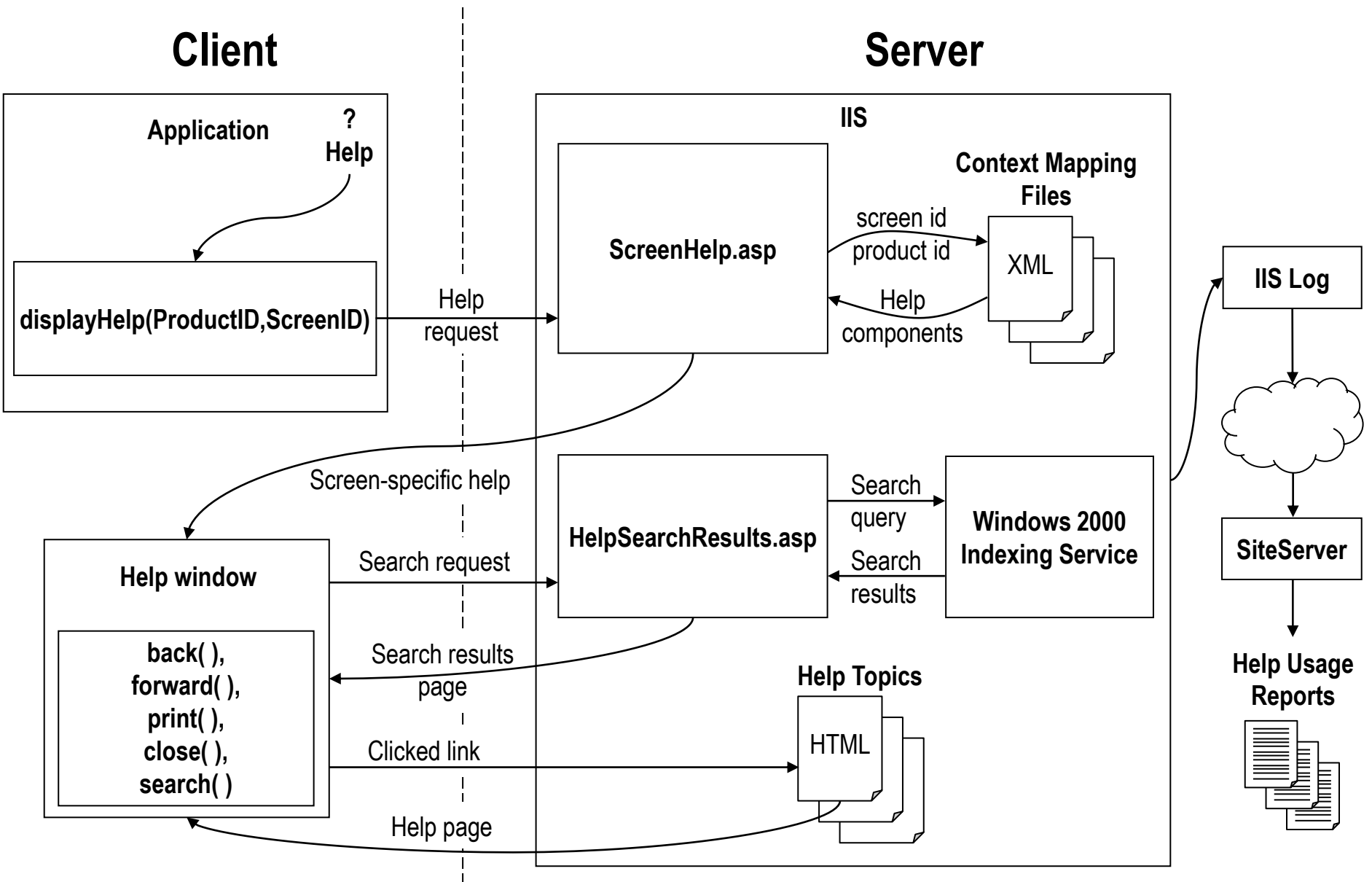

- Independent Consultant: Helpful Technology Pty Ltd, Melbourne, Australia
- Bachelor of Science (Physics), Graduate Certificate in Applied Science (Scientific and Technical Writing)
- Technical writer since 1987, Help author since 1994
- Demonstrating Help for a corporate Internet banking application
- helpfulau@yahoo.com

- What we did
- Why we did it
- How we did it
- What's next?

- Project background/requirements
- Architecture
- Demo
 - Accessing Help from the application
 - Context-sensitive elements of initial Help page
 - No Help topics found page

- One of the world's top 100 banks
- Corporate internet banking application
- 4 modules
- 80 windows
- Browsers: IE 5+ / Netscape 6+
- Minimum bandwidth: 56k modem
- 6 UI developers, 6 business analysts
- 1 Help architect/author (100 d)
- 1 Help ASP/JavaScript developer (35 d)

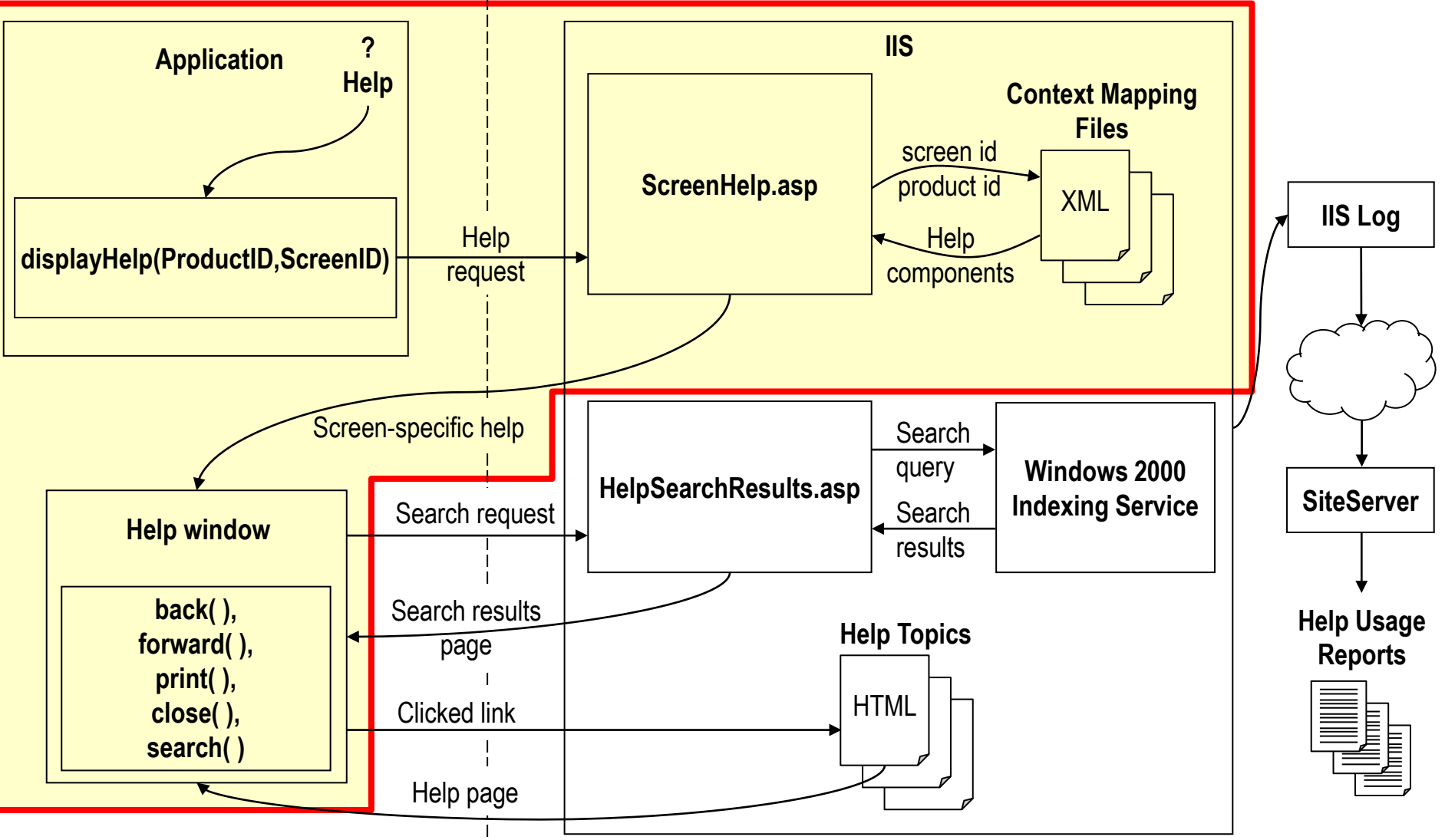
Help System Architecture



Focus of Presentation

Client

Server



Demo – Accessing Help 1

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "Sample Application - Microsoft Internet Explorer". The address bar contains "http://user/HelpTest/sample_home.asp". The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar contains icons for Back, Forward, Stop, Refresh, Home, Search, Favorites, History, Mail, Print, Edit, Discuss, and Real.com. The main content area displays a web page titled "Sample App" with a green header. In the top right of the page, there are three navigation links: "Home", "Tutorial", and "Help". The "Help" link is circled in red. Below the header, the page shows a login status "07/24/2002 You are logged in as: Stewart Walker" and a "Create Project" form with various input fields and a "Submit Project" button. The status bar at the bottom shows "javascript:displayHelp('ACCT_Reporting','Balances and Transactions (Balances)');" and "Local intranet".

Demo – Accessing Help 2

The screenshot shows a Microsoft Internet Explorer browser window with the address bar set to `http://user/HelpTest/sample_home.asp#`. The main content area displays a 'Sample App' interface with a green header and a sidebar menu containing 'Create project', 'View projects', 'View schedule', and 'Search'. The main content area shows a 'Create Project' form with fields for 'Project Number', 'Description', 'Start Date', 'Completion Date', 'Requestor', 'Priority' (set to 1), and 'Tasks', along with a 'Submit Project' button.

Overlaid on the right side of the browser is a 'GOLD Help' window titled 'GOLD Help -- Web Page Dialog'. The help window has a blue header with 'corpbank.com' and a search bar. The main content of the help window is titled 'Balances and Transactions (Balances tab)' and includes a 'print' button. The text explains that the tab shows end-of-day balances and that the 'Transactions' and 'Balance History' tabs use the selected account. Below this, there are sections for 'Help topics' (with links like 'Export displayed balances', 'Find particular accounts or balance records', 'Sort the listed balance records', 'View additional details for an account', and 'View data from a different day'), 'Frequently asked questions' (with links like 'Can I view latest balances rather than close of business balances?', 'Can I view the balance history of an account?', and 'Why can't I see an account?'), and 'Key screen items' (with links like 'Additional Account Details' and 'Total Debits and Credits'). At the bottom, there is a link for 'More help, training and support'.

Context-sensitive Topic Elements

The screenshot shows a web browser window titled "GOLD Help -- Web Page Dialog" displaying a help page for "Balances and Transactions (Balances tab)". The page includes a search bar, a brief description, a "Help topics" section with procedure links, a "Frequently asked questions" section with FAQ links, and a "Key screen items" section with field help links. Callout boxes highlight these elements: "Screen title" points to the page title, "Brief description" points to the introductory paragraph, "Procedure links" points to the list of help topics, "FAQ links" points to the frequently asked questions, and "Field Help links" points to the key screen items.

Brief description

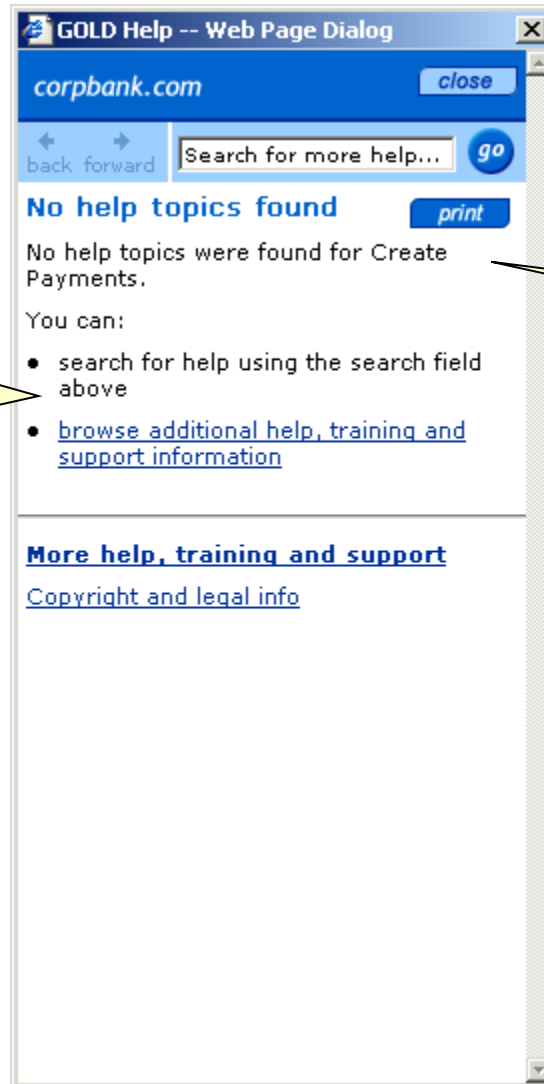
Screen title

Procedure links

FAQ links

Field Help links

No Help Topics Found Page



Other Help options

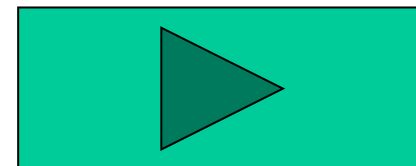
Expected Screen ID

- Great for debugging!

- **Simplicity**
 - For users
 - For authors
 - For developers
- **Get maximum benefit from first click**
- **Minimize impact on app & developers**
- **Scalability/modularity**
- **Extensibility (platform for customization)**
- **Automate mapping file maintenance as much as possible**

- Content mapping files
- Requesting Help from the application
- Identifying relevant Help topics
- Building the Help page
- Displaying the Help page
- Maintaining the mapping files
- Working with the developers

- Akin to Winhelp or HTML Help map files
- Define **one-to-many** relationships between app pages and Help topics
- Includes link text
- One mapping file per “product” folder
- Additional mapping file for common topics
- XML format

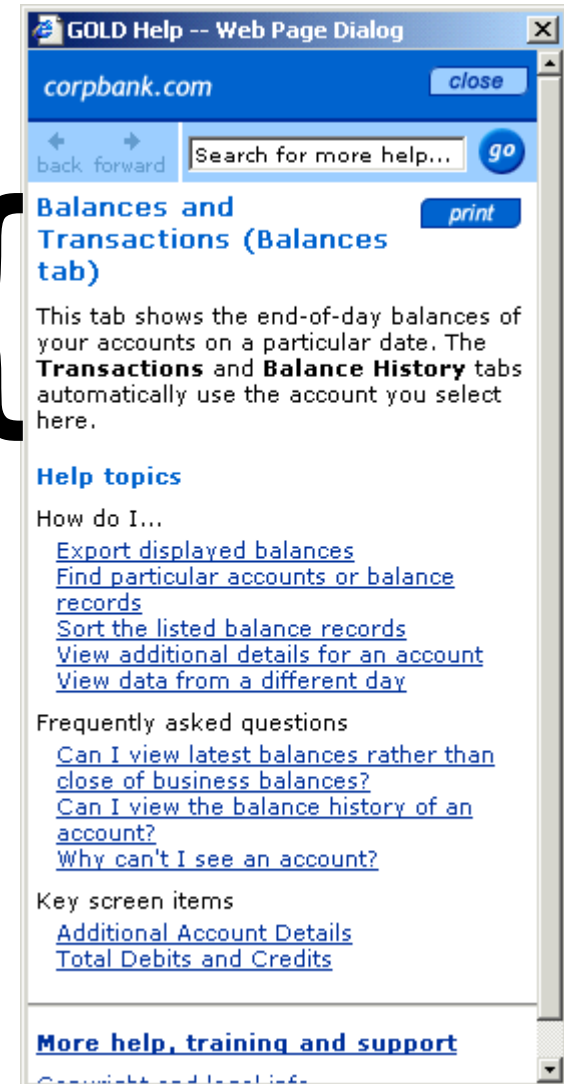


Content Mapping Files – Example

```
<Screen>  
  <ID>  
    <![CDATA[Balances and Transactions  
    (Balances)]]>  
  </ID>  
  <DescriptionURL>  
    <![CDATA[ACCT_Reporting/screens/  
    balances.htm]]>  
  </DescriptionURL>  
  ...  
</Screen>
```

Screen ID

URL of screen
description topic



Content Mapping Files – Example

<Screen>

...

<HowTo>

...

<TopicLink>

<ID>19</ID>

<LinkText>

<![CDATA[Sort the listed
balance records]]>

</LinkText>

</TopicLink>

...

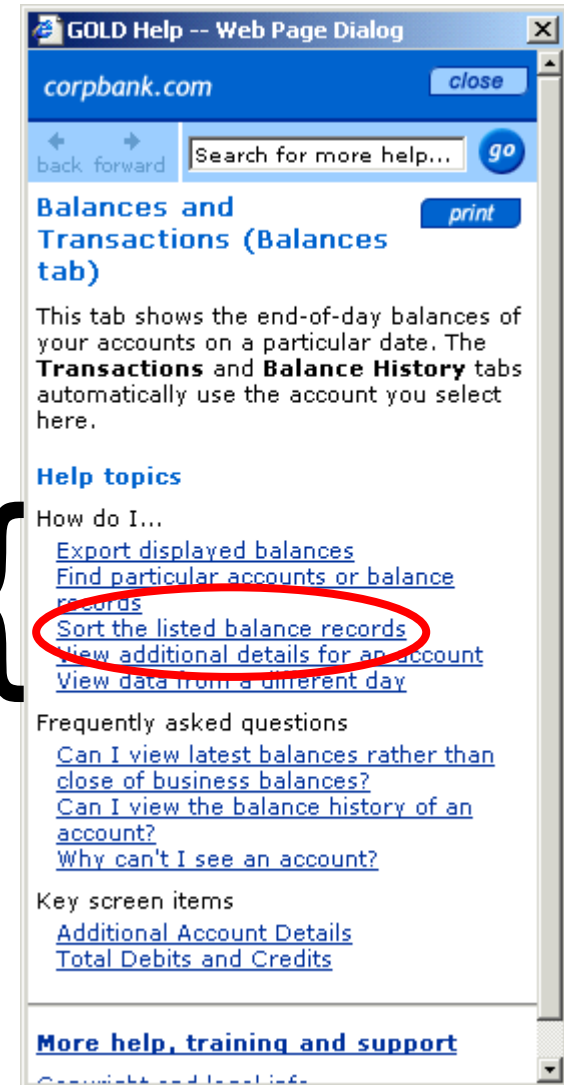
</HowTo>

...

</Screen>

Topic ID (pointer to
topic details)

Link text to use when
linking from this
screen (optional)



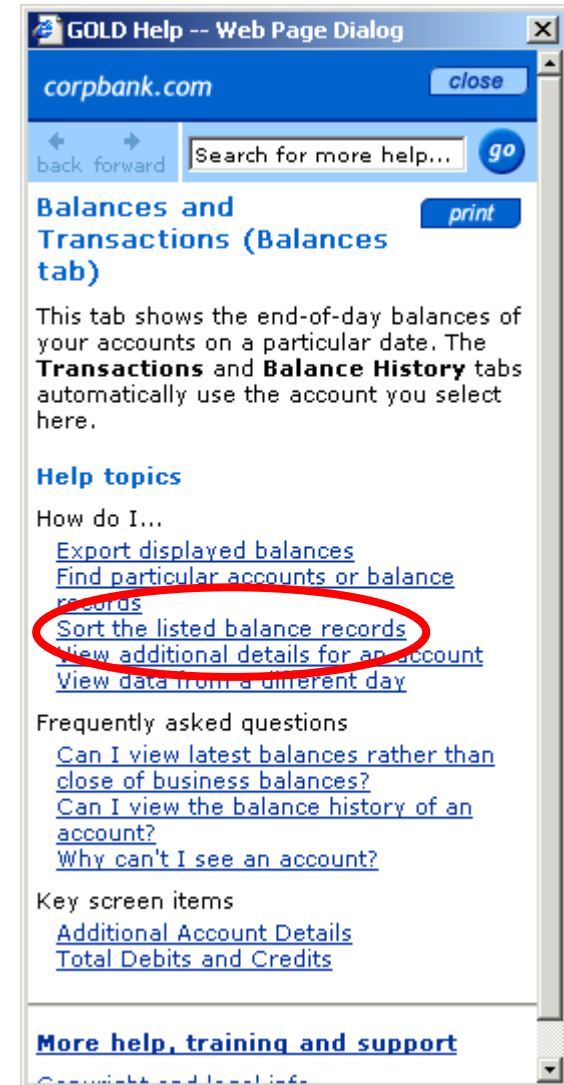
Content Mapping Files – Example

```
<Topics>
...
<Topic>
  <ID>19</ID>
  <URL>
    <![CDATA[ACCT_Reporting/howto/
      sort_table.htm]]>
  </URL>
  <LinkText>
    <![CDATA[Sort the listed data]]>
  </LinkText>
</Topic>
...
</Topics>
```

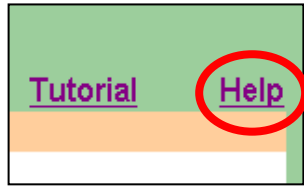
Topic ID

URL of Help topic

Default link text for this topic



Requesting Help from the App



```
<a href="javascript:displayHelp('ACCT_Reporting',  
'Balances and Transactions (Balances)')">Help</a>
```

```
<script>
```

```
function displayHelp(strProductID, strScreenID){
```

```
...
```

```
helpURL='helpFrameset.asp?txtPage=' + 'screenHelp.asp?  
txtProductID=' + escape(strProductID) + '&txtScreenID=' +  
escape(strScreenID);
```

```
window.objOnlineHelpWindow =  
eval('window.showModelessDialog(helpURL,window,"help:0;  
resizable:1;dialogLeft:525px;dialogTop:30px;scroll:0;  
dialogWidth:275px;dialogHeight:543px;status:0" )');
```

```
... }
```

```
</script>
```

screenHelp.asp

VBScript code (not JavaScript!)

```
strProductPath = "/Help/" & strProductID
```

```
strProductMapFile = strProductPath &  
"/ContentMap.xml"
```

```
createDOM objProductXML
```

```
booLoadSuccess = objProductXML.load  
(Server.MapPath (strProductMapFile))
```

```
Set objTopics = objProductXML.  
selectSingleNode("/Product/Topics")
```

Displaying the Screen Topic

```
Set objScreen =  
objProductXML.selectSingleNode("/Product/  
Screens/Screen[ID $ieq$ '" & strScreenID &  
"']")
```

```
Set objDescriptionURL =  
objScreen.selectSingleNode("DescriptionURL")
```

```
strDescriptionPath = "/Help/" &  
objDescriptionURL.firstChild.nodeValue
```

```
Server.Execute strDescriptionPath
```

```
Response.Write "<h2>Help topics</h2>"
```

```
Set objHowToList =  
objScreen.selectNodes("HowTo/TopicLink")
```

```
Response.Write "<P CLASS='text1'>How do  
I...</p>"
```

```
OutputTopicLinks objHowToList, objTopics
```

FAQ and field topic links use the same approach
but with different paths and headings

OutputTopicLinks Function 1

Function OutputTopicLinks(objTopicLinkList, objTopics)

...

For Each **objTopicLink** In objTopicLinkList

strTopicID =

objTopicLink.selectSingleNode("ID").firstChild.nodeValue

Set **objTopic** = objTopics.selectSingleNode("Topic[ID \$ieq\$ " & **strTopicID** & "]")

strTopicURL =

objTopic.selectSingleNode("URL").firstChild.nodeValue

strLinkText =

objTopic.selectSingleNode("LinkText").firstChild.nodeValue

OutputTopicLinks Function 2

```
Response.Write "<P CLASS='linkpara'>"
```

```
Response.Write "<a href='" & strPath & "/Help/" &  
strTopicURL & "'>"
```

```
Response.Write strLinkText
```

```
Response.Write "</a></p>"
```

...

Next

...

End Function

- HAMSTER (Help to Application Mapping Structure Editor) – custom VB app
- Requires the following folders for each product:
 - faq
 - fields
 - howto
 - screens
- HAMSTER checks folders and updates mapping file with new and deleted topics and screens
- Help author defines relationships between screens and topics, and adds link text

The screenshot shows the Hamster tool interface. At the top, the window title is "Hamster" and the menu bar includes "File", "Tools", and "Help". The "Product" dropdown menu is set to "ACCT_Reporting".

The left pane displays a "Content Map" tree structure:

- Content Map
 - Screens
 - Account Profile
 - HowTo
 - FAQ
 - Fields
 - Balances and Transactions (Balance History tab)
 - Balances and Transactions (Balances tab)
 - Balances and Transactions (Transactions tab)
 - Export Balances
 - Export Statement
 - Export Transactions
 - Print Statement
 - Search for Balance History
 - Search for Balances
 - Search for Transactions
 - Transaction Details
 - Topics
 - Common Topics

The right pane is titled "Edit Topic Links" and "Add Topic Links". It contains the following controls:

- "Select A Topic To Add:"
 - Product Topics [Dropdown]
 - Common Topics [Dropdown]
- Default Link Text: [Text Input]
- Screen-specific Link Text: [Text Input]
- Buttons: "Add" and "Clear"

At the bottom of the window, a status bar displays the message: "Loading content map: H:\GOLD Project\Documents\Help\Content\ACCT_Reporting" followed by "Finished loading content map."

- System Architect and Senior UI Developer included in review process for Help design documents
- Help integration conventions agreed upfront
 - ASP code inserts correct Help link into each app page (developer's responsibility)
 - Everything else is Help team's responsibility
- Robust documentation set
 - Functional and Technical Design docs
 - Maintenance Guide for Help authors
 - Deployment Guide for administrators

- Personalized Help based on user permissions
 - Broadening our definition of “context” – making context-sensitive Help sensitive to the user’s context and not just the system’s context
- XCML (eXtensible Context Mapping Language)?

- Choosing a technology
- Working with developers
- Testing
- "Gotchas"