

OnWeb 7.5.0

Samples Guide

Micro Focus (IP) Ltd.
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

Copyright 2010 Micro Focus (IP) Limited. All Rights Reserved.

MICRO FOCUS, the Micro Focus logo and RUMBA are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

All other marks are the property of their respective owners.

Table of Contents

Chapter 1: Overview

- OnWeb samples • 6
 - Sample applications • 6
 - Sample scripts • 7

Chapter 2: NetTelecom Sample Application

- Using NetTelecom sample application • 9
 - Adding the user and the user group manually • 10
- How the NetTelecom application was developed • 12
 - The challenge • 12
- Application architecture • 17
 - NetTelecom Login application • 17
 - NetTelecom application • 21

Chapter 3: Sample Scripts

- Sample Data Source scripts • 42
 - VBScript scripts • 42
 - JScript/JavaScript scripts • 42
 - REXX scripts • 43
- Sample Data Services scripts • 43
 - VBScript scripts • 43
 - JScript/JavaScript scripts • 45
 - REXX scripts • 46
 - SQL scripts • 47
- Sample Business Logic scripts • 48
 - VBScript scripts • 48
 - JScript scripts • 49
 - REXX scripts • 50
- Sample User Services scripts • 51
 - VBScript scripts • 51
 - JScript scripts • 51



Table of Contents

- REXX scripts • 52
- HTML scripts • 52
- XSL scripts • 53
- Sample Presentation File scripts • 53
 - HTML scripts • 53
 - XSL scripts • 53
- Sample HTMLCustomization scripts • 54
 - VBScript scripts • 54
 - JScript scripts • 54
- Sample ReturnToParkingScreen scripts • 54
 - VBScript scripts • 54
 - JScript scripts • 55
- Sample Host Publishing supporting files • 55
- Sample Host Publishing logon scripts • 55
- Sample LiveConnect scripts • 56

Index • 57



OnWeb® gives you the power to build Web applications that take advantage of the existing data and business logic in your enterprise. OnWeb also provides the tools to create new, reusable logic for your organization's Web applications.

An application that works with OnWeb can be a Web page in a browser or a stand-alone executable. You can create the application using any language or development environment such as the Microsoft® Visual Basic® and various HTML editors.

OnWeb applications are stored on OnWeb Server, which can be installed on one of the following platforms: Windows®, Solaris™, Linux®, or AIX® platforms, or the IBM® iSeries™ systems.

OnWeb is part of the Micro Focus® Host Services Platform (HSP), a comprehensive solution for accessing, presentation, and integration of host services. HSP offers a modular, standards-based design that connects Micro FocusRUMBA® and OnWeb products to create unified host access, presentation, and management services. The platform works seamlessly with .NET, Web services, and Java™ applications.

NETMANAGE® Host Services Platform



OnWeb samples

OnWeb comes with two types of samples, sample applications and sample scripts. Sample scripts are automatically installed with the product. With sample application, you must specify that you want to install them.

Sample applications

You can install the following sample applications:

- **NetTelecom** is a sample Host Integration application that allows customers of a fictitious telephone company, NetTelecom, to view and manage their accounts. For more information on this sample application, see [Chapter 2, “NetTelecom Sample Application”](#).
- **EnhancedB** is a sample Host Integration application that illustrates how to use conditional Business Logic objects.
- Two sample Visual Basic source codes:
 - › **VB_Async** uses the IOnWebIObjct ActiveX® control for asynchronous access.
 - › **VB_Sync** uses IOnWebIObjct AciveX control for synchronous access.
- Two sample Visual C++® source codes:
 - › **VC_UseCOM** uses IOnWeb COM object and smart pointers
 - › **VC_UseLib** uses the IOnWeb library files

► To install sample applications

- For a single user installation, select **Complete** on the **Select Installation Type** screen.
- When you install OnWeb Server on a separate machine, select **Sample Applications** on the **Select Features** screen.

You can examine the code for each application by importing it into OnWeb Designer. If you play with sample applications and change their code, use the same importing procedure to bring back the application to its original state.



Using Visual Basic and Visual C++ sample applications

To examine OnWeb objects used in these applications, you must import them into Designer.

► To import sample applications into Designer

1. Start OnWeb Designer.
2. From the **Tools** menu, choose **Sample Applications**.
3. Select the required application from the list and click **OK**.

To run each application, you must first open source code in the relevant program and then compile the application.

Sample scripts

OnWeb provides a wide variety of sample scripts that you can use as a starting point when creating your own script. Although these scripts will not work “as is” and require customization, they will help create your own custom scripts. For a full list of available sample scripts, see [Chapter 3, “Sample Scripts”](#).



This chapter describes how to use the NetTelecom sample application, the process behind creation of this application, and the architecture of the application.

Using NetTelecom sample application

You can use the NetTelecom sample application in two ways:

- You can examine the application code in OnWeb Designer.
- You can run the application to see how it works.

In order to examine the application code, you must import this application into Designer.

► To import NetTelecom application into Designer

1. From the Designer's **Tools** menu, choose **Sample Applications**.
2. Select the NetTelecom application from the list and click **OK**.

This process installs two applications:

- NetTelecomLogin - this application is used to start the main NetTelecom application.
- NetTelecom - this is the main NetTelecom application

It also creates a user group NetTelecomGroup that has access permissions for the NetTelecom application. This user group contains one user, Sta_VerifiedUser.

► **To run the NetTelecom application**

1. Build both the NetTelecom and the NetTelecom Login applications to the server:
 - › In Designer, right-click the NetTelecom application name in the tree list and choose **Build Item** from the menu.
 - › Right-click the NetTelecomLogin application name in the tree list and choose **Build Item** from the menu.

This step needs to be performed only once.

2. In the browser, enter:

IPaddress:Port/NetTelecomLogin/User_Services/Sta_Login.htm

where *IPaddress* is the IP address of your OnWeb Server, and *Port* is the OnWeb Server port number.

Running the NetTelecom application allows you to see an OnWeb application in action. On each page displayed by the application, there is a “How OnWeb does it” button. Click this button to see how OnWeb was used to develop that page.

Adding the user and the user group manually

If the import process does not install the NetTelecomGroup user group and/or the Sta_VerifiedUser user, or if you delete the content of the Server\information_rules directory, you can manually add the user and the user group required to run the NetTelecom sample application. This process consists of 5 steps, that are performed in OnWeb Administrator:

- **Step 1.** Add user Sta_VerifiedUser
- **Step 2.** Add user group NetTelecomGroup.
- **Step 3.** Add user Sta_VerifiedUser to the NetTelecomGroup user group.
- **Step 4.** Approve actions performed in the preceding steps.
- **Step 5.** Configure the NetTelecomGroup group access to the NetTelecom application.

Note: Before proceeding, make sure that the NetTelecom application has been built to the server, as described above.



► **To add user Sta_VerifiedUser**

1. Log on to OnWeb Administrator and click the **User Management** tab.
2. Click **Add User**.
3. In the **User name** box, type Sta_VerifiedUser.
4. In the **Password** box, type Sta_VerifiedUser.
5. In the **Confirm password** box, re-type the password.
6. Click **Save** to create the user.

► **To add user group NetTelecomGroup**

1. In OnWeb Administrator, click the **Group Management** tab.
2. Click **Add Group**.
3. In the **Group name** box, type NetTelecomGroup.
4. Click **Save** to add the new user group.

► **To add user to the user group**

1. In OnWeb Administrator, click the **User Management** tab.
2. In the **Users** list, select the Sta_VerifiedUser user.
3. Click **Edit User**.
4. In the **Not member of** list, select the NetTelecomGroup group.
5. Click the << button. The NetTelecomGroup name will be moved to the **Member of** list.
6. Click **Save** to save your changes.

► **To approve new actions**

1. In OnWeb Administrator, click the **Pending Actions** tab.
2. For each of the three actions (Add User, Add Group, and Add Membership), select the radio button in the **Approve** column.
3. Click **Save** to finalize the approval process.

► **To assign group access to the application**

1. In OnWeb Administrator, click the **Application Access** tab.
2. From the **Applications** list, select the NetTelecom application.
3. Click **Modify Access**.
4. In the **Without access** list, select the NetTelecomGroup group.
5. Click the << button. The name of the group will be moved to the **With access** list.
6. Click **Apply** to save your changes and return to the **Application Access** tab.

How the NetTelecom application was developed

Let's assume that NetTelecom is typical of organizations in several industries — particularly the telecommunications, utilities and financial segments — that are undergoing the changes wrought by deregulation, mergers and acquisitions and competition. As a result of these forces, NetTelecom needs to expand their market and their services, improve customer loyalty, decrease customer churn and digest a newly acquired Internet service provider. Maybe your organization is facing similar challenges. NetTelecom's experience reflects and communicates many of the advantages realized by over 80 Global 2000 organizations that have chosen OnWeb. While NetTelecom may be fictional, the benefits of OnWeb are real.

The challenge

NetTelecom is a US-based telecommunications company that offers residential local and toll services, as well as mobile telephone services. In a bid to take a market lead and improve customer loyalty, NetTelecom committed to a strategic plan to exploit the Internet by providing customer service through an Extranet. NetTelecom also recently acquired Mega-Net Internet Services Inc., an Internet Service Provider (ISP), with headquarters 200 miles away. This acquisition will allow NetTelecom to expand its portfolio of services and add a substantial number of new customers to its installed base.

Challenge #1 - Select the right technologies with which to develop and deploy a customer service Extranet.

Challenge #2 - Integrate Mega-Net's information systems.

NetTelecom planned to exploit the Internet to:

- expand its market
- integrate multiple services into one on-line account status system
- offer new service packages, including a customer loyalty and discount program
- maintain their image as a leading telecommunications company
- improve customer loyalty and reduce churn
- integrate Mega-Net's computing infrastructure with NetTelecom's.

Here is how they did it.

Facing Challenge #1: On-line account status application

NetTelecom wants to keep the customers they have and expand their markets. If a current customer subscribed to residential, mobile and Internet services, they received three different bills, a problem for customers and a continuing source of confusion, customer complaints and cashflow problems.

NetTelecom wanted to provide a more integrated approach, where customers could view the total amount they owe, while also viewing transaction details. By integrating the services, NetTelecom could also offer customer loyalty programs, such as discounts for subscribing to multiple services. This all needed to have one clean look in a simple-to-use service with access to authorized users 24-hours a day.

Computing environment

NetTelecom's billing was a decades-old mainframe-based system with a CICS® application that provided access to local and toll call records and mobile service records. They had an in-house application that printed bills, which was hard-coded and difficult to change. Printing was done through a series of batch processes and access to information was through 3270 screens only. In addition, NetTelecom had an NT Server running MS SQL Server. The database contained information on custom calling features for Residential and Mobile services, with access to data through ODBC.

Requirements

NetTelecom's requirements were to:

- provide integrated account status information to customers who had previously received multiple bills every month
- create new information by calculating such items as discounts without modifying the existing databases or mainframe applications
- leverage their existing 3270 and SQL infrastructure
- scale to support tens of thousands of users across the nation as they grew
- prepare for change, since the company plans more acquisitions
- develop and implement this service in less than six months
- reduce the project risk
- stay within budget

Alternatives

NetTelecom considered doing a roll-your-own application. But every time they had done this before, they took longer than expected and went over budget. Worse, the results did not scale, were difficult to update, were slow and often were not robust. This time, NetTelecom knew they needed a better, more predictable way. Customers were now involved, and mistakes and delays were not an option.

Solution

Mr. Technicky, the Director of Information Systems, chose OnWeb for the following reasons:

- OnWeb makes a project completely predictable. The product is proven and tested through real customer development and deployment projects making results reliably predicted. You can easily say how long it will take, what it will cost to create and maintain, and how many customers it will support.
- OnWeb allows NetTelecom to integrate multiple data sources into a single application in real time and offer new service and pricing packages not otherwise possible.
- OnWeb's linear scalability will support NetTelecom's growth path.



- OnWeb leverages their existing systems and their Year 2000 work.
- OnWeb fits their security structure and adds a fifth layer of application security.
- OnWeb has been proven for Extranets and customer service solutions worldwide.

Because OnWeb has a true three-tier architecture, end-users never directly touch their mainframes or databases. This also protects their end-users from change since the application is independent of Enterprise platform changes.

Application

NetTelecom's on-line account status application is available for you to view in a browser or download for your own use. The solution gives customers the ability to:

- view monthly account statements, which give integrated account totals as well as detailed transactions
- view and change account information
- view a list of available services and their descriptions
- order or cancel services
- determine discount savings if the customer were to take advantage of offered programs

OnWeb's three-tier architecture places the business logic on the OnWeb Server. OnWeb can therefore create new information out of the data existing in databases and mainframes without changing anything in them. Your mainframe and a database do what they do best, store data, while OnWeb does what it does best, perform business logic functions.

Facing Challenge #2: Integrating Mega-Net

Requirements

Having just acquired Mega-Net Internet Services Inc. (this is a fictitious company), NetTelecom was faced with the challenge of integrating Mega-Net's enterprise data. Mega-Net had an NT Server running relational databases containing customer and services information. The data was distinct and separate from all of NetTelecom's other data sources, and access to the data was through ODBC drivers. The challenge was further compounded by the fact that since NetTelecom had already launched their

on-line account status application, the Internet services would need to be treated as another service in the application. The integration would have to be invisible to customers and cause no disruption in the on-line service.

Solution

Since NetTelecom was already using OnWeb for their on-line account status application, they knew they could take advantage of these features in OnWeb to integrate Mega-Net:

- OnWeb's ability to access almost any data source — 3270, 5250, VTs, RDBMS, and proprietary systems, allowed them to integrate Mega-Net's database, even though it was 200 miles away.
- OnWeb's object-based architecture. With Information Objects®, reusable data components, NetTelecom's developers can change a single object and have that change flow through the entire application, wherever the object was used. Information Objects are also reusable through any number of applications. If the object is changed, the change applies automatically to all the applications.
- OnWeb's fast and efficient development time. Developers were able to develop faster in OnWeb because of its visual GUI development suite. Information Objects are seen as icons that can be dragged and dropped into the applications. Relationships among data objects can be easily seen. As a result, application development is fast, developers can concentrate on creating the best business logic instead of creating code and the application's design can be visually tested against its requirements.

OnWeb has become an integral part of NetTelecom's internal and external communications strategy. As NetTelecom grows, the company will expand its Extranet plan to embrace business partners and other stakeholders as part of an overall e-commerce vision. OnWeb gives them the development path to turn strategy into reality.

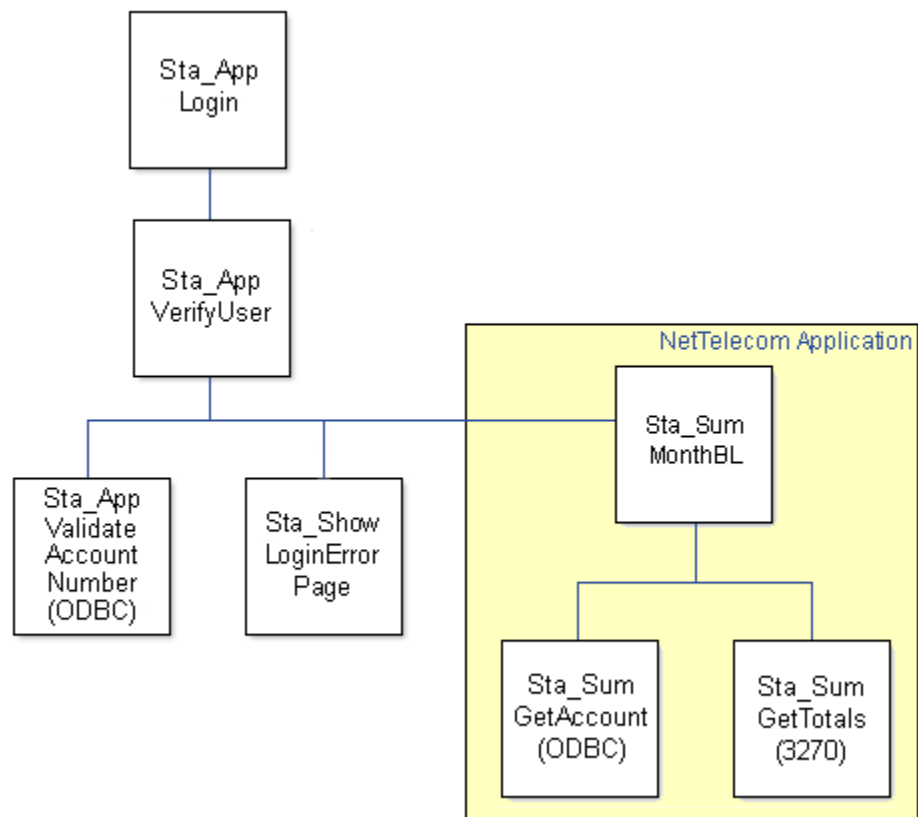


Application architecture

This section describes in details processes in the NetTelecom application. The NetTelecom application comprises two sub-applications, NetTelecom Login and NetTelecom.

NetTelecom Login application

This application logs the user onto the NetTelecom application. It has unrestricted user access.



Sta_AppLogin

This rule is the entry point of the NetTelecom Login application.

Type	User Service
Initiated by	The Sign On button on the Welcome page
Parameters	AccountNumber, Pin

Sets Global Var	None
Uses Global Var	None
Returns	Either an HTML page displaying Customer Summary for current billing period, or if the login fails, the Login page with an error message.

Sta_AppVerifyUser (Refine script)

Acts as a collector object containing one rule, Sta_AppValidateAccountNumber which queries the database for the AccountNumber parameter.

Type	Enhanced Business Logic
Initiated by	Sta_AppLogin
Parameters	AccountNumber, Pin
Sets Global Var	None
Uses Global Var	None
Returns	One table with one row and two columns. The first column contains a string with HTML to display. The second column contains "true" or "false", denoting whether the session should be disconnected or not.

Sta_AppVerifyUser (Plan script)

Verifies that the Account and PIN numbers entered at "Sign On" match those contained in the database and returned by Sta_AppValidateAccountNumber. There is also a check on the number of attempts; after three, the record is locked for the day. Upon successful validation, it initializes session variables, logs on to OnWeb Server as "Sta_VerifiedUser" as part of the "NetTelecom Group" group, and calls Sta_SumMonthBL.

Type	Enhanced Business Logic
Initiated by	Sta_AppLogin
Parameters	AccountNumber, Pin
Sets Global Var	AccountNumber, LoginAccountNumber, CustomerID, ErrorMessage, Month, MonthName



Uses Global Var None

Returns Nothing

Sta_AppValidateAccountNumber

An SQL statement queries the Sta_Customer_Account table in the ODBC Data Source and returns a result set of the correct CustomerID, login date, login number, and PIN for the provided AccountNumber. The PIN is evaluated, by the Plan script of Sta_AppVerifyUser against the PIN provided by the user through the Sign On page to secure the site and records.

Type ODBC Data Service

Initiated by Sta_AppVerifyUser

Parameters AccountNumber

Sets Global Var None

Uses Global Var None

Returns Result set containing the correct CustomerID, login date, login number, and PIN for the AccountNumber provided.

Sta_ShowLoginErrorPage

Displays the Sta_Login.htm page with the login error message and disconnects the user from the application and server.

Type Data Service

Initiated by Sta_AppVerifyUser

Parameters None

Sets Global Var None

Uses Global Var ErrorMessage

Returns One table with one row and two columns. The first column contains a string with the HTML to display. The second column contains "true" to denote that the session should be disconnected.

Sta_SumMonthBL

See the description in “Sta_SumMonthBL” on page 22.

Sta_SumGetAccount

See the description in “Sta_SumGetAccount” on page 22.

Sta_SumGetTotals

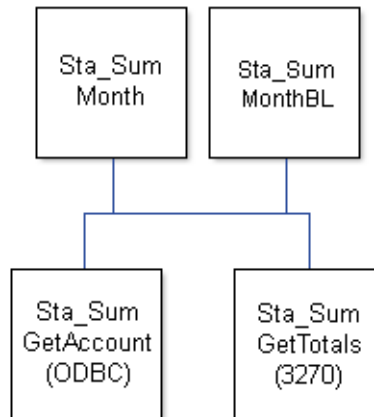
See the description in “Sta_SumGetTotals” on page 23.



NetTelecom application

The access to this application is restricted to the members of the NetTelecom Group user group. By default, this group contains one user, Sta_VerifiedUser.

View Customer Summary for selected month



Sta_SumMonth

This User Service calls its two component rules, which assemble all the necessary account information specific to a given month (in the same way it is done during the application's logon for the current billing period), and inserts its data into the appropriate parts of the Customer Summary HTML page.

Type	User Service
Initiated by	The Enter button below option list of months on the Customer Summary page or from the Back to Customer Summary button located on several pages.
Parameters	Month
Sets Global Var	ispAccountNumber
Uses Global Var	MonthName, Month, Year
Returns	HTML displaying Customer Summary page for the selected month.

Sta_SumMonthBL

Generates a table containing two strings, an HTML string created from the Customer Summary page Sta_CustomerSummary.htm, and a string, "false", for the disconnect value. The Customer Summary page displays a listing of the various accounts a given client has and the totals when all these accounts' billing information is tabulated for the specified billing period (as indicated by drop-down list of months selection on the Customer Summary page).

Type	Business Logic
Initiated by	Sta_AppVerifyUser
Parameters	Month
Sets Global Var	ispAccountNumber
Uses Global Var	MonthName, Month, Year
Returns	The updated Sta_CustomerSummary.htm contents and disconnect is set to false.

Sta_SumGetAccount

Based on the CustomerID stored in the Session variable pool, an SQL statement is issued to get all the records matching it. This SQL query will generate a Result set of Account numbers associated with the CustomerID.

Type	ODBC Data Service
Initiated by	Sta_SumMonth Sta_SumMonthBL Sta_AccountDiscount
Parameters	ChangeMonth
Sets Global Var	Month, MonthName, Year
Uses Global Var	Month, MonthName, CustomerID
Returns	Result set containing a list of Account numbers associated with the global variable CustomerID

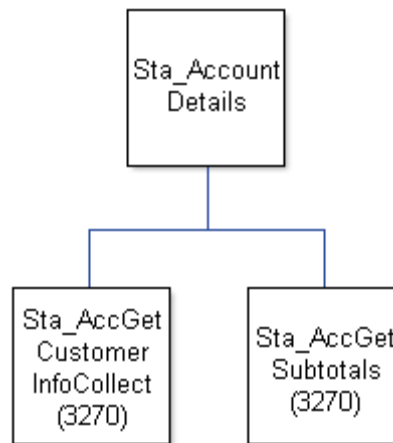


Sta_SumGetTotals

For each account number in the Sta_SumGetAccount Result set, OnWeb will use these account numbers and the MonthName stored in the Session variable pool to locate the account on the mainframe and get the type of account (Mobile or Residence) and the total due. OnWeb will generate a table to store the data gathered.

Type	3270 Terminal Data Service
Initiated by	Sta_SumMonth Sta_SumMonthBL
Parameters	AccountList – Result set from Sta_SumGetAccount
Sets Global Var	None
Uses Global Var	MonthName, Year
Returns	Two tables. The first table will contain one row and three columns, PreviousBalance, PaymentReceived, and LatePaymentCharges. This first table provides summary information, combining all the totals from all clients' accounts. The second table will have as many rows as the client has accounts and will have three columns, PhoneNumber, PhoneType, and Total.

View Account Details



Sta_AccountDetails

Calls two components, Sta_AccGetCustomerInfoCollect and Sta_AccGetSubtotals, to assemble the necessary information and perform the bulk of the work and, then, displays the resulting data in the Account Details Page.

Type	User Service
Initiated by	The Details button associated with particular accounts on the Customer Summary page or the Back to Account Details button located on several pages.
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	Month, MonthName, Year
Returns	HTML of Account Details page



Sta_AccGetCustomerInfoCollect

Navigates the mainframe application and, using the AccountNumber, gets the Customer's personal information, such as their name, address and phone.

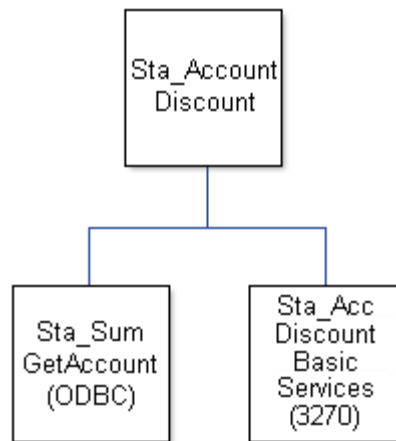
Type	3270 Terminal Data Service
Initiated by	Sta_AccountDetails Sta_AccGetCustomerInfo
Parameters	AccountNumber
Sets Global Var	AccountNumber
Uses Global Var	MonthName, Year
Returns	Result set containing the Customer's preferred Salutation, FirstName, LastName, PhoneNumber, Address, City, State/Prov. and ZIP/Postal Code.

Sta_AccGetSubtotals

Navigates the mainframe application to get Subtotals and Totals for Basic Services (for a Residential line) or Usage (for a Mobile), Long Distance calling and Custom Calling Features. It will also include information such as what the Previous Balance was, whether Payment was received, if Late Payment Charges should be incurred, etc. All this information is returned to the Sta_AccountDetails User Service and used to perform the necessary calculations to display the Account's detailed information page.

Type	ODBC Data Service
Initiated by	Sta_AccountDetails
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	MonthName, Year
Returns	Result set containing Totals, SubTotals, PreviousBalance, LatePaymentCharge, etc. for the Account in question.

Discount



Sta_AccountDiscount

Calls two components, `Sta_SumGetAccount` and `Sta_AccDiscountBasicServices`, to assemble the necessary information and perform the bulk of the work and, then displays the resulting data in the New Discount Program page.

Type	User Service
Initiated by	The Check Out the Super Discount Program – Starting Next Month button on the Customer Summary page for the current billing period.
Parameters	Month
Sets Global Var	None
Uses Global Var	None
Returns	HTML of Next Month – New Discount Program page

Sta_SumGetAccount

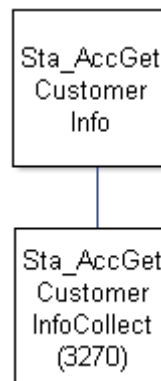
See the description in “`Sta_SumGetAccount`” on page 22.

Sta_AccountDiscountBasicServices

Navigates the mainframe application and sums the totals and discounts of the basic services for each account registered against a particular Customer.

Type	3270 Terminal Data Service
Initiated by	Sta_AccountDiscount
Parameters	AccountList – Result set from Sta_SumGetAccount
Sets Global Var	None
Uses Global Var	ErrorCode, MonthName, Year
Returns	Two sets of results. The first contains the Subtotal, Discount and subsequent Totals for each account. The second is used to store the cumulative discounts and totals for all the accounts registered for that particular Customer.

Get/Change Customer Information



Sta_AccGetCustomerInfo

The Account owner's preferred Salutation, FirstName, LastName, PhoneNumber, Address, City, State/Prov. and ZIP/Postal Code is returned by Sta_AccGetCustomerInfoCollect and presented to the user in editable text fields where they can make the changes required.

Type	User Service
Initiated by	The Change of Information button on Account Details page
Parameters	AccountNumber

Sets Global Var	None
Uses Global Var	None
Returns	HTML of Change of Information page

Sta_AccGetCustomerInfoCollect

See the description in “Sta_AccGetCustomerInfoCollect” on page 25.

Save Customer Information



Sta_AccSaveCustomerInfo

Passes the newly submitted values for the Customer Information fields to Sta_AccSaveCustomerInfoCollect. That component performs the update and returns a table containing the new values which is then shown in a refresh display of the Change of Information page.

Type	User Service
Initiated by	The Save Changes button on Change of Information page
Parameters	AccountNumber, FirstName, LastName, Street, City, State, Zip, Phone, Salutation
Sets Global Var	None
Uses Global Var	None
Returns	HTML Page of Change of Information page



Sta_AccSaveCustomerInfoCollect

Navigates the mainframe application, updating its Customer Information fields with the information submitted through the Change of Information page. It then creates a new table containing this same information (and, therefore, reflecting the changes just made) and returns it to Sta_AccSaveCustomerInfo for display.

Type	3270 Terminal Data Service
Initiated by	Sta_AccSaveCustomerInfo
Parameters	AccountNumber, FirstName, LastName, Street, City, State, Zip, Phone, Salutation
Sets Global Var	None
Uses Global Var	MonthName, Year
Returns	Result set containing preferred Salutation, FirstName, LastName, PhoneNumber, Address, City, State/Prov. and ZIP/Postal Code of Customer.

Long Distance



Sta_AccLongDistance

Displays the current account's Long Distance Details using the table returned from Sta_AccLongDistanceCollect to fill in the details.

Type	User Service
Initiated by	Long Distance Details button on Account Details page

Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	None
Returns	HTML display of Long Distance Details

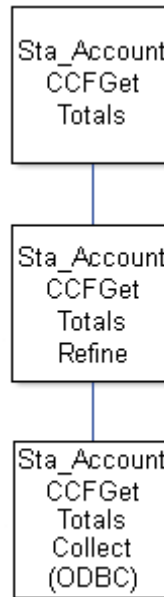
Sta_AccLongDistanceCollect

Navigates the mainframe application and uses the AccountNumber to get the long distance information for the current account.

Type	3270 Terminal Data Service
Initiated by	Sta_AccLongDistance
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	MonthName, Year
Returns	Result set containing Long Distance Details associated with the AccountNumber



Display Current Custom Calling Features



Sta_AccountCCFGetTotals

Indicates which Custom Calling Features the customer either currently subscribes to or did subscribe to during the month being queried.

Type	User Service
Initiated by	Custom Calling Features Details button on Account Details page or the Back to Custom Calling Features button on the Add Custom Calling Features page.
Parameters	AccountNumber, Month
Sets Global Var	None
Uses Global Var	None
Returns	HTML of Custom Calling Features page

Sta_AccountCCFGetTotalsRefine

This rule adds more information, such as the AccountNumber, MonthName and Year, to the table returned by Sta_AccountCCFGetTotalsCollect and passes it back to Sta_AccountCCFGetTotals for display in the HTML.

Type	Business Logic
Initiated by	Sta_AccountCCFGetTotals
Parameters	AccountNumber, Month
Sets Global Var	None
Uses Global Var	MonthName, Year, AccountNumber
Returns	Result set containing Custom Calling Feature names, their prices, the AccountNumber, MonthName and Year requested.

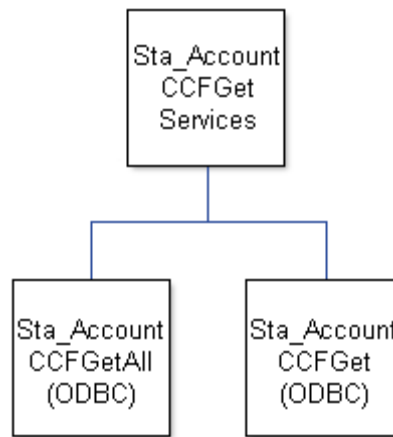
Sta_AccountCCFGetTotalsCollect

An SQL statement queries two tables in the ODBC Data Service, Sta_Available_Services and Sta_Services_History. This query returns a Result set of the specified Account's registered Custom Calling Features for the month requested.

Type	ODBC Data Service
Initiated by	Sta_AccountCCFGetTotalsRefine
Parameters	AccountNumber, Month
Sets Global Var	None
Uses Global Var	None
Returns	Result set containing Custom Calling Feature names and their prices for the indicated Account and Month.



Subscribe/Cancel Custom Calling Features



Sta_AccountCCFGetServices

Based on the result set of the two component rules, this User Service generates an HTML page where the user can cancel and/or subscribe to the various Custom Calling Features.

Type	User Service
Initiated by	The Change Custom Calling Features button on Custom Calling Features Details page
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	AccountNumber, Month
Returns	HTML for Add Custom Calling Features page

Sta_AccountCCFGetAll

An SQL statement queries the Sta_Available_Services table in the ODBC Data Service and returns a Result set of all available Custom Calling Features offered, including their names and IDs.

Type	ODBC Data Service
Initiated by	Sta_AccountCCFGetServices Sta_AccountCCFSaveAll
Parameters	None
Sets Global Var	None
Uses Global Var	None
Returns	Result set containing the available Custom Calling ServiceNames and their ServiceIDs.

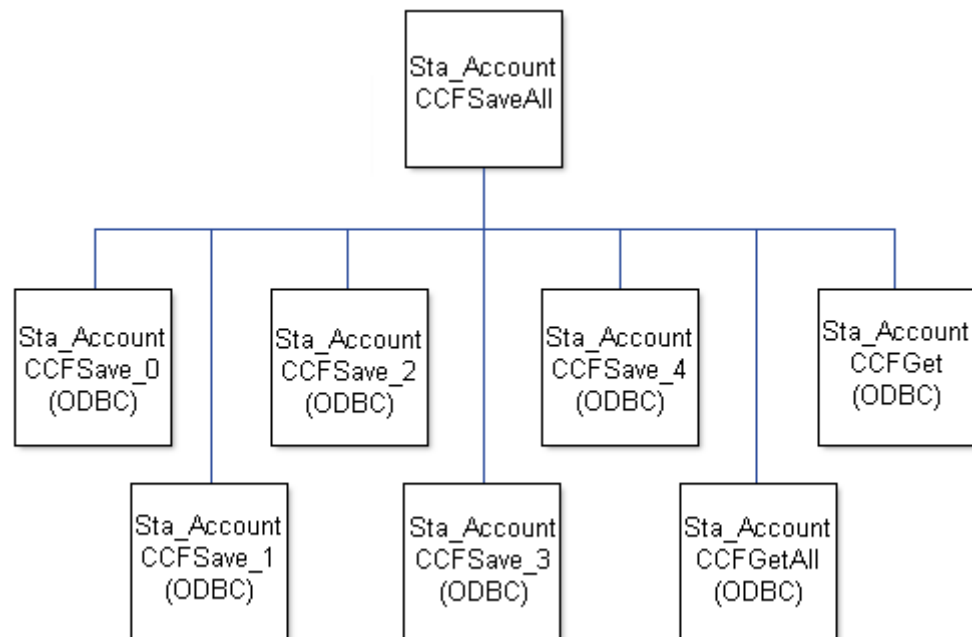
Sta_AccountCCFGet

An SQL statement queries the Sta_Current_Services table in the ODBC Data Service and returns a Result set of the requested Account's current Customer Calling Features.

Type	ODBC Data Service
Initiated by	Sta_AccountCCFGetServices Sta_AccountCCFSaveAll
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	None
Returns	Result set containing the specified Account's SERVICE_IDS indicating their current Customer Calling Features.



Save Custom Calling Features Changes



Sta_AccountCCFSaveAll

Based on the Result set of its components, this User Service generates an HTML displaying the user's new Custom Calling Features configuration.

Type	User Service
Initiated by	The Save button on Add Custom Calling Features page
Parameters	AccountNumber, Old1, New1, Old2, New2, Old3, New3, Old4, New4, Old5, New5
Sets Global Var	None
Uses Global Var	AccountNumber, Month
Returns	Updated Add Custom Calling Features page

Sta_AccountCCFSave_?

(where ? is a number from 0 to 4)

These five (5) rules are called by Sta_AccountCCFSaveAll, once for each of the Custom Calling Features available. The ServiceID that is passed into these Data Services is a constant whose value corresponds to a given Custom Calling Feature in the database. Based on the values passed in through the Old and New parameters, each of these rules will determine if it needs to insert or delete (subscribe or cancel) the indicated service for the customer.

Type	ODBC Data Services
Initiated by	Sta_AccountCCFSaveAll
Parameters	AccountNumber, Old?, New?, ServiceID (a constant value from 0 to 4 depending on which Save_? Component is being run)
Sets Global Var	None
Uses Global Var	None
Returns	Result sets containing nothing, as SQL Update and Delete operations do not return anything.

Sta_AccountCCFGetAll

See the description in “Sta_AccountCCFGetAll” on page 34.

Sta_AccountCCFGet

See the description in “Sta_AccountCCFGet” on page 34.



Mobile Usage



Sta_AccUsage

Displays the current mobile account's Usage Details using the table returned from Sta_AccUsageCollect to fill in the details.

Type	User Service
Initiated by	Usage Details button on Account Details page
Parameters	AccountNumber
Sets Global Var	None
Uses Global Var	None
Returns	HTML display of Usage Details

Sta_AccUsageCollect

Navigates the mainframe application and uses the AccountNumber to get the Usage information for the current mobile phone account.

Type	3270 Terminal Data Service
Initiated by	Sta_AccUsage
Parameters	AccountNumber
Sets Global Var	None

- Uses Global Var** MonthName
Year
- Returns** Result set containing Usage Details associated with the AccountNumber

Application Logout



Sta_AppLogout

Calls Sta_AppLogoutCollect to perform the bulk of the work and redisplay the HTML Welcome Page with the line, "You are now disconnected from the NetTelecom Internet Application", added.

- Type** User Service
- Initiated by** The **Exit** buttons on any page within the application
- Parameters** None
- Sets Global Var** None
- Uses Global Var** None
- Returns** HTML Welcome Page



Sta_AppLogoutCollect

Performs the steps to exit the mainframe's application before disconnecting from the mainframe and terminating the Session on the OnWeb Server.

Type	3270 Terminal Data Service
Initiated by	Sta_AppLogout
Parameters	None
Sets Global Var	None
Uses Global Var	None
Returns	An empty table.



When you want to add a script to an object in Designer, you have a choice of starting with either a sample script or an empty script. Designer automatically presents you with the list that contains only the scripts that are appropriate for the selected object and scripting language. It is recommended that you use Designer to add and modify sample scripts to your application.

This chapter lists all the sample scripts that are available in Designer. The scripts are organized by the object type as follows:

- “Sample Data Source scripts” on page 42
- “Sample Data Services scripts” on page 43
- “Sample User Services scripts” on page 51
- “Sample Presentation File scripts” on page 53
- “Sample HTMLCustomization scripts” on page 54
- “Sample ReturnToParkingScreen scripts” on page 54
- “Sample Host Publishing supporting files” on page 55
- “Sample Host Publishing logon scripts” on page 55
- “Sample LiveConnect scripts” on page 56

Within each object type, the scripts are organized by the scripting language.

These scripts will not work “as is” and require customization to fit your system. For example, you may need to adjust the script for the number of parameters that you have in your object. Read any notes listed at the beginning of the script for further instructions on how to use the script.

The sample script files are located in the Templates subdirectory of the Designer installation directory. You can open them to examine in any text editor.

Notes: Do not delete or rename files located in the Template directory.

The **Script name** column in the tables below shows the names of the scripts as they appear on the list in Designer.

Sample Data Source scripts

In order to select a Data Services sample script to collect data from a Data Source (Terminal Based or ODBC), a relationship must first be created between the Data Source and the Data Services objects. Only Code Only sample scripts are available before this relationship is created.

VBScript scripts

Script name	Description	Filename
Minimum required	Basic declarations required to log on to a Terminal Data Source.	VBScriptTerminalDS_Logon.smpl
Minimum required	Basic declarations required to log off from a Terminal Data Source.	VBScriptTerminalDS_Logoff.smpl
Advanced logon	Connects and logs on to a host.	VBScriptTerminalDS_AdvLogon.smpl

JScript/JavaScript scripts

The following scripts will work in both, the JScript® and the JavaScript™ scripting environments.

Script name	Description	Filename
Minimum required	Basic declarations required to log on to a Terminal Data Source.	JScriptTerminalDS_Logon.smpl
Minimum required	Basic declarations required to log off from a Terminal Data Source.	JScriptTerminalDS_Logoff.smpl
Advanced logon	Connects and logs on to a host.	JScriptTerminalDS_AdvLogon.smpl



REXX scripts

Script name	Description	Filename
Minimum required	Basic declarations required to log on to a Terminal Data Source	RexxTerminalDS_Logon.smpl
Minimum required	Basic declarations required to log off from a Terminal Data Source	RexxTerminalDS_Logoff.smpl
Advanced logon	Connect and logs on to a host	RexxTerminalDS_AdvLogon.smpl

Sample Data Services scripts

VBScript scripts

Script type	Script name	Description	Filename
Code only	Minimum Required	Basic declarations required for any Data Services object	VBScriptCollect_Minimum.smpl
	Simple table	Creates a table of country and capital names.	VBScriptCollect_SimpleTable.smpl
	Using parameters	Creates a table with a city name and population based on user specified values	VBScriptCollect_UsingParam.smpl
ODBC	Minimum required	Basic declarations required to log on to an ODBC Data Service.	VBScriptCollect_Minimum.smpl

Script type	Script name	Description	Filename
	Simple query	Queries Biblio.mdb database for book titles containing the word 'Information'.	VBScriptCollect_ODBC_SimpQuery.smpl
	Parameter query	Queries Biblio.mdb database for user specified book titles.	VBScriptCollect_ODBC_ParamQry.smpl
	Dynamic parameter query	Queries all fields from a user specified table and the WHERE condition.	VBScriptCollect_ODBC_DynParam.smpl
Terminal	Minimum required	Basic declarations required for a Terminal Data Service	VBScriptCollect_Term_Minimum.smpl
	First MVSDEMO screen	Retrieves the first screen from MVSDEMO.	VBScriptCollect_Term_1stMVSDEMO.smpl
	Account profile	Retrieves account profile information from MVSDEMO.	VBScriptCollect_Term_ProMVSDEMO.smpl
Submit	Parameters in variables	Assigns values passed in as parameters to Session Variables so they are available in all of the application.	VBScriptCollect_UsingParam_Submit.smpl



JScript/JavaScript scripts

The following scripts will work in both, the JScript and the JavaScript scripting environments.

Script type	Script name	Description	Filename
Code only	Minimum Required	Basic declarations required for any Data Services object	JScriptCollect_Minimum.smpl
	Simple table	Creates a table of country and capital names.	JScriptCollect_SimpleTable.smpl
	Using parameters	Creates a table with a city name and population based on user specified values	JScriptCollect_UsingParam.smpl
ODBC	Minimum required	Basic declarations required to log on to an ODBC Data Service.	JScriptCollect_Minimum.smpl
	Simple query	Queries Biblio.mdb database for book titles containing the word 'Information'.	JScriptCollect_ODBC_SimpQuery.smpl
	Parameter query	Queries Biblio.mdb database for user specified book titles.	JScriptCollect_ODBC_ParamQry.smpl
	Dynamic parameter query	Queries all fields from a user specified table and the WHERE condition.	JScriptCollect_ODBC_DynParam.smpl
Terminal	Minimum required	Basic declarations required for a Terminal Data Service	JScriptCollect_Term_Minimum.smpl

Script type	Script name	Description	Filename
	First MVSDEMO screen	Retrieves the first screen from MVSDEMO.	JScriptCollect_Term_1stMVSDEMO.smpl
	Account profile	Retrieves account profile information from MVSDEMO.	JScriptCollect_Term_ProMVSDEMO.smpl
Submit	Parameters in variables	Assigns values passed in as parameters to Session Variables so they are available in all of the application.	JScriptCollect_UsingParam_Submit.smpl

REXX scripts

Script type	Script name	Description	Filename
Code only	Minimum required	Basic declarations required for a code-only Data Service	RexxCollect_Minimum.smpl
	Simple table	Creates a table of country and capital names	RexxCollect_SimpleTable.smpl
	Using parameters	Creates a table with a city name and population based on user specified values.	RexxCollect_UsingParam.smpl
ODBC	Minimum required	Basic declarations required for an ODBC Data Service	RexxCollect_Minimum.smpl
	Simple query	Queries Biblio.mdb database for book titles containing the word 'Information'.	RexxCollect_ODBC_SimpQuery.smpl



Script type	Script name	Description	Filename
	Parameter query	Queries Biblio.mdb database for user specified book titles.	RexxCollect_ODBC_ParamQry.smpl
	Dynamic parameter query	Queries all fields from a user specified table and the WHERE condition.	RexxCollect_ODBC_DynParam.smpl
Terminal	Minimum required	Basic declarations required for a Terminal Data Service.	RexxCollect_Term_Minimum.smpl
	First MVSDEMO screen	Retrieves the first screen from MVSDEMO.	RexxCollect_Term_1stMVSDEMO.smpl
	Account profile	Retrieves account profile information from MVSDEMO.	RexxCollect_Term_ProMVSDEMO.smpl

SQL scripts

Script name	Description	Filename
Simple query	Queries Biblio.mdb database for book titles containing the word 'Information'.	SQLCollect_SimpleQuery.smpl
Parameter query	Queries Biblio.mdb database for user specified book titles.	SQLCollect_ParameterQuery.smpl

Sample Business Logic scripts

VBScript scripts

Script type	Script name	Description	Filename
Refine	Minimum required	Basic declarations required for a Business Logic object	VBScriptRefine_Minimum.smpl
	Sorting	Sorts the incoming table by the first column values	VBScriptRefine_Sorting.smpl
	Extract column	Creates a new table based on first column of the incoming table.	VBScriptRefine_ExtractColumn.smpl
	Add HTML formatting	Formats a host screen into HTML syntax.	VBScriptRefine_AddHTMLFormat.smpl
	Add IObject	Adds all of the incoming IObjects to the Refine's IObject	VBScriptRefine_AddIObjects.smpl
Plan	Minimum Required	Basic declarations required for any Plan script	VBScriptPlan_Minimum.smpl
	Conditional Plan	Executes a Data Services object if a specified condition is met	VBScriptPlan_Conditional.smpl
Fail	Minimum Required	Basic declarations required for any Fail script	VBScriptFail_Minimum.smpl



JScript scripts

Script type	Script name	Description	Filename
Refine	Minimum required	Basic declarations required for a Business Logic object	JScriptRefine_Minimum.smpl
	Sorting	Sorts the incoming table by the first column values	JScriptRefine_Sorting.smpl
	Extract column	Creates a new table based on first column of the incoming table.	JScriptRefine_ExtractColumn.smpl
	Add HTML formatting	Formats a host screen into HTML syntax.	JScriptRefine_AddHTMLFormat.smpl
	Add IObject	Adds all of the incoming IObjects to the Refine's IObject	JScriptRefine_AddIObjects.smpl
Plan	Minimum Required	Basic declarations required for any Plan script	JScriptPlan_Minimum.smpl
	Conditional Plan	Executes a Data Services object if a specified condition is met	JScriptPlan_Conditional.smpl
Fail	Minimum Required	Basic declarations required for any Fail script	JScriptFail_Minimum.smpl

REXX scripts

Script type	Script name	Description	Filename
Refine	Minimum required	Basic declarations required for a Business Logic object.	RexxRefine_Minimum.smpl
	Sorting	Sorts the incoming table by the first column values.	RexxRefine_Sorting.smpl
	Extract column	Creates a new table based on the first column of the incoming table.	RexxRefine_ExtractColumn.smpl
	Add HTML formatting	Formats a host screen into HTML syntax.	RexxRefine_AddHTMLFormat.smpl
	Add IObjects	Adds all of the incoming IObjects to the Refine's IObject	RexxRefine_AddIObjects.smpl
Plan	Minimum Required	Basic declarations required for any Plan script	RexxPlan_Minimum.smpl
	Conditional Plan	Executes a Data Services object if a specified condition is met	RexxPlan_Conditional.smpl
Fail	Minimum Required	Basic declarations required for any Fail script	RexxFail_Minimum.smpl



Sample User Services scripts

VBScript scripts

Script name	Description	Filename
Minimum required	Basic declarations required for a User Services object.	VBScriptPresent_Minimum.smpl
Present HTML	Creates an HTML presentation for all tables in the IObject.	VBScriptPresent_HTML.smpl
Present XML	Creates an XML presentation for all tables in the IObject.	VBScriptPresent_XML.smpl
Present One Table	Creates an HTML presentation for the first table in the IObject	VBScriptPresent_OneTable.smpl
Present Multiple Tables	Creates an HTML presentation for all the tables in the IObject	VBScriptPresent_MultipleTables.smpl

JScript scripts

Script name	Description	Filename
Minimum required	Basic declarations required for a User Services object.	JScriptPresent_Minimum.smpl
Present HTML	Creates an HTML presentation for all tables in the IObject.	JScriptPresent_HTML.smpl
Present XML	Creates an XML presentation for all tables in the IObject.	JScriptPresent_XML.smpl
Present One Table	Creates an HTML presentation for the first table in the IObject	JScriptPresent_OneTable.smpl
Present Multiple Tables	Creates an HTML presentation for all the tables in the IObject	JScriptPresent_MultipleTables.smpl

REXX scripts

Script name	Description	Filename
Minimum required	Basic declarations required for a User Service object.	RexxPresent_Minimum.smpl
Present HTML	Creates an HTML presentation for all tables in the IObject.	RexxPresent_HTML.smpl
Present XML	Creates an XML presentation for all tables in the IObject.	RexxPresent_XML.smpl
Present One Table	Creates an HTML presentation for the first table in the IObject	RexxPresent_OneTable.smpl
Present Multiple Table	Creates an HTML presentation for all the tables in the IObject	RexxPresent_MultipleTables.smpl

HTML scripts

Script name	Description	Filename
Minimum required	Basic declarations required for an HTML User Services object.	HTMLPresent_Minimum.html
First cell	Displays the first cell of a table in an HTML page.	HTMLPresent_FirstCell.html
All tables	Displays all tables contained in the IObject in an HTML page.	HTMLPresent_AllTables.html



XSL scripts

Script name	Description	Filename
Minimum required	Basic declarations required for an XSL User Services object.	XSLPresent_Minimum.xsl
First cell	Displays the first cell of a table in an HTML page.	XSLPresent_FirstCell.xsl
All tables	Transforms all tables contained in the IObject from XML to HTML.	XSLPresent_AllTables.xsl

Sample Presentation File scripts

HTML scripts

Script name	Description	Filename
Minimum required	Basic declarations required for an HTML Presentation File object.	HTMLPFO_Minimum.html
First cell	Displays the first cell of a table in an HTML page.	HTMLPFO_FirstCell.html
All tables	Displays all tables contained in the IObject in an HTML page.	HTMLPFO_AllTables.html

XSL scripts

Script name	Description	Filename
Minimum required	Basic declarations required for an XSL Presentation File object.	XSLPFO_Minimum.xsl
First cell	Displays the first cell of a table in an HTML page.	XSLPFO_FirstCell.xsl
All tables	Transforms all tables contained in the IObject from XML to HTML.	XSLPFO_AllTables.xsl

Sample HTMLCustomization scripts

VBScript scripts

Script name	Description	Filename
Minimum required	Basic declarations required for HTMLCustomization.	VBScriptHTMLCustomization_Function.smpl

JScript scripts

Script name	Description	Filename
Minimum required	Basic declarations required for HTMLCustomization.	JScriptHTMLCustomization_Function.smpl

Sample ReturnToParkingScreen scripts

VBScript scripts

Script name	Description	Filename
Minimum required	Basic declarations and code required for the ReturnToParkingScreen script.	VBScriptReturnToParkingScreen_Minimum.smpl



JScript scripts

Script name	Description	Filename
Minimum required	Basic declarations and code required for the ReturnToParkingScreen script.	JScriptReturnToParkingScreen_Minimum.smpl

Sample Host Publishing supporting files

Name	Description	Filename
Standard Launch Page	Default startup page for a Host Publishing application.	_LaunchPage1.htm
Skip Launch Page	Use this page not to display the startup page. It automatically submits application information to the server.	_LaunchPage2.htm
Mobile Device Launch Page	A default startup page for use with applications that will be accessed through a mobile device.	mobile_launchpage.htm
Simple header	A sample header file. See instructions in the HTML code.	_Header1.htm
Simple footer	A sample footer file. See instructions in the HTML code.	_Footer1.htm

Sample Host Publishing logon scripts

Script name	Description	Filename
Host logon	Retrieves session variables and connects to the host.	JScript_HostPub_logon.smpl

Sample LiveConnect scripts

Script name	Description	Filename
LiveConnectDB2 Query (1)	A JDBC query to a DB2® database, using the OnWeb createConnection method.	JScriptCollect_LiveConnect_DB2_Connect_Query.smpl
LiveConnectDB2 Query (2)	A JDBC query to a DB2 database, using the DB2DataSource class.	JScriptCollect_LiveConnect_DB2_Query.smpl
LiveConnect MySQL Query	A JDBC query to a MySQL database using the OnWeb createConnection method.	JScriptCollect_LiveConnect_MySql_Connect_Query.smpl
LiveConnect MySQLUpdate	A JDBC update to a MySQL database using the OnWeb createConnection method.	JScriptCollect_LiveConnect_MySql_Connect_Update.smpl
LiveConnectJDBC-ODBC Query	An ODBC query to a MS Accessdatabase using the JDBC-ODBC bridge, utilizing the OnWeb createConnection method.	JScriptCollect_LiveConnect_JDBC_ODBC_Connect.smpl
LiveConnect Oracle Query	A JDBC query to an Oracle database using the OracleDataSource class.	JScriptCollect_LiveConnect_Oracle_Query.smpl



B

Business Logic object, sample scripts • 48

D

Data Services object, sample scripts • 43

Data Source object, sample scripts • 42

E

EnhancedB sample application • 6

H

Host Publishing supporting files, samples • 55

Host Services Platform • 5

HSP • 5

HTML Customization, sample scripts • 54

HTML scripts, samples • 52, 53

I

importing sample applications to Designer • 7

J

JavaScript scripts, samples • 42, 45

JScript scripts, samples • 42, 45, 49, 51, 54, 55

L

LiveConnect, sample scripts • 56

logon script, sample • 55

N

NetTelecom sample application • 6

adding user and group • 10

architecture • 17

developing • 12

R

ReturnToParkingScreen, sample scripts • 54

REXX scripts, samples • 43, 46, 50, 51, 52

S

sample applications

EnhancedB • 6

importing to Designer • 7

installing • 6

NetTelecom • 6

Visual Basic • 6

Visual C++ • 6

sample scripts • 7

Business Logic • 48

Data Services • 43

Data Source • 42

HTML • 52, 53

HTML Customization • 54

JavaScript • 42, 45

JScript • 42, 45, 49, 51, 54, 55

LiveConnect • 56

logon • 55

ReturnToParkingScreen • 54

REXX • 43, 46, 50, 52

SQL • 47

User Services • 51

VBScript • 42, 43, 48, 51, 54

XSL • 53

sample supporting files • 55

SQL scripts, samples • 47

U

User Services object, sample scripts • 51

V

- VB_Async, sample code • 6
- VB_Sync, sample code • 6
- VBScript scripts, samples • 42, 43, 48, 51, 54
- VC_UseCOM, sample code • 6
- VC_UseLib, sample code • 6
- Visual Basic sample applications • 6
- Visual C++ sample applications • 6

X

- XSL scripts, samples • 53

