SUPPLEMENT TO

New Zealand Gazette

OF THURSDAY, 9 SEPTEMBER 2010

WELLINGTON: WEDNESDAY, 15 SEPTEMBER 2010 — ISSUE NO. 119

MINISTRY OF FISHERIES

FISHERIES (ELECTRONIC SOFTWARE REQUIREMENTS) CIRCULAR 2010 (No. F556)

PURSUANT TO REGULATION 41M OF THE FISHERIES (REPORTING) REGULATIONS 2001

Fisheries (Electronic Software Requirements) Circular 2010 (No. F556)

Pursuant to Regulation 41M of the Fisheries (Reporting) Regulations 2001, the Chief Executive of the Ministry of Fisheries issues the following circular.

Notice

1. **Title**—This circular is the Fisheries (Electronic Software Requirements) Circular 2010.

2. **Commencement**—This circular comes into force the day after its notification in the *New Zealand Gazette*.

3. **Introduction**—The Ministry of Fisheries Catch Effort process provides the means for recording and validating catch, effort, landing and environmental information from the New Zealand commercial fishing fleet, operating within New Zealand's Exclusive Economic Zone (EEZ) or on the high seas. MFish and FishServe have implemented an electronic process for the capture of this data referred to as Catch Effort Electronic Data Transfer (CE EDT). This process allows Commercial Fishers to electronically record and provide catch effort information at the point of fish capture, to sign and send the information to FishServe and for this information to be transferred to MFish. As well as being able to transmit data from the Commercial Fisher to FishServe, the process also includes acknowledgements from the FishServe database that the forms have been received and the ability for the fisher's systems to receive regular updates of any reference information. This CE EDT process is based on Commercial Fishers being able to use approved software to provide the information which is then validated at MFish.

   This circular describes the requirements of Electronic Software that Commercial Fishers can use when they register for CE EDT so that authentic and secure catch effort information can be obtained from those Commercial Fishers fishing in the New Zealand EEZ or on the high seas. The Electronic Software will interface with an Application Programming Interface produced by FishServe that acts as an interface between the Electronic Software and the FishServe database.

   Suppliers wishing to provide Electronic Software to Commercial Fishers must obtain approval from MFish for their proposed Electronic Software.

4. **Interpretation**—In this circular:

   (a) "Access Identifier" means an identifier that is issued to an individual by the chief executive under Regulation 41K of the Fisheries (Reporting) Regulations 2001 for the purpose of completing and providing Catch Effort Forms using electronic software and uniquely identifies that individual. These identifiers will be USB tokens.

   (b) "API" means the Application Programming Interface, which is a piece of software installed on the user's PC that acts as an interface between the Electronic Software and the FishServe and MFish databases.

(c) “Authorised User” means a person approved by the chief executive under Regulation 41C of the Fisheries (Reporting) Regulations 2001 to complete and provide Catch Effort Forms using electronic software.

(d) “Catch Effort Form” means any Catch Effort form required to be completed and provided under the Fisheries (Reporting) Regulations 2001 or any High Seas form required under a High Seas Permit issued in accordance with section 113H of the Fisheries Act 1996 including those provided electronically.

(e) “CE EDT” means Catch Effort Electronic Data Transfer.

(f) “CE EDT Emulator” means a system that replicates the interface with FishServe’s EDT system in order to allow the testing of proposed Electronic Software’s functionality.

(g) “CE EDT System” means a combination of systems that will enable the administration of CE EDT registration information, the collection of CE data and the transfer of CE data from Authorised Users to FishServe and FishServe to MFish.

(h) “CE Form Number” means the form number allocated by MFish to a Catch Effort Form.

(i) “Commercial Fisher” means a person who holds a fishing permit issued under section 91 of the Fisheries Act 1996; and includes a person who holds a high seas fishing permit and a person using a New Zealand ship who, in the judgement of the chief executive, holds a valid authority from a foreign country to take highly migratory species in the national fisheries jurisdiction of that foreign country.

(j) “Compliance Tool” means software that will be used by MFish to view and copy Catch Effort data and form history information.

(k) “Digital Signature” means a unique signature that is created when an Access Identifier and Password are used in combination with one another to sign an activity on an electronic catch effort form.

(l) “Electronic Software” means a software application approved by the chief executive under Regulation 41N of the Fisheries (Reporting) Regulations 2001 that facilitates the submission of Catch Effort Forms via CE EDT.

(m) “Event History” means an electronic record of all data entry activity both before and after a return has been Digitally Signed.

(n) “FishServe” means Commercial Fisheries Services Limited.

(o) “Form Iteration” means iterations of a CE EDT Form capture and records each individual creation, opening and saving of all events on a form and includes the name of Authorised User, date and time of the actions. The data is stored on the local machine on which it was created. This information is summarised as the name, date, time and changes made to each field (but will not include the Digital Signature) and is incorporated onto the relevant form version.

(p) “Form version” means a CE EDT form which has been completed and digitally signed. Versions that are subsequently amended and signed will become additional versions.

(q) “MFish” means the Ministry of Fisheries.

(r) “Reference Data” means the set of codes currently valid for reporting information on Catch Effort Forms as required by MFish.

5. **Legal Requirements for Software**—Any software that is submitted for approval by MFish must comply with and allow the fisher to comply with all current legislative requirements. In particular:

(a) The Electronic Software must comply with the legal requirements of the Fisheries Act 1996, Fisheries (Reporting) Regulations 2001 and any other requirements set out in this circular;

(b) The Electronic Software must allow an Authorised User to enter data into a Catch Effort Form over the course of a day so that they can meet their legislative requirements (for example to enter data "as soon as it becomes available", "on each day at sea", "before the end of the day" or "immediately on landing" or "when information becomes available from an LFR") as appropriate for the form type; and

(c) The Electronic Software must have at least the same fields presented in such manner as to reliably and comprehensibly replicate each form type as specified in Schedule 2 of the Fisheries (Reporting) Regulations 2001.

6. **Minimum System Requirements**—The Electronic Software must:

(a) Comply with the operational workflows specified in Appendix 3 relating to the interaction of the Electronic Software with the FishServe API.

(b) Have at least the same fields presented in such manner as to reliably and comprehensibly replicate form types specified in Schedule 2 of the Fisheries (Reporting) Regulations 2001 (or elsewhere as specified by MFish) and allow data entry into each such field on the relevant Catch Effort Forms. Each form type must be in English, use standard Reference Data, be in a font size large enough to read and allow the collection of at least as many fishing or landing events as the paper form it replicates (although it may allow for more).

(c) Display the unique CE Form Number that has been issued by MFish for each Catch Effort Form created.

(d) Ensure a valid access identifier and password combination has been entered before a user is able to create, endorse, export, import, copy, discontinue, sign or submit a Catch Effort Form.

(e) Display each CE EDT form's current status when the Catch Effort Form is viewed.

(f) Have the facility to allow the Authorised User to manually save and/or endorse the information they have entered into a Catch Effort Form at any time.

(g) Ensure that the event history of a form can be viewed both before and after a Catch Effort Form has been digitally signed.

(h) Transfer the content of a Catch Effort Form Completely and accurately to the API.

(i) Ensure that if a Catch Effort Form is amended and re-signed, all the different versions are identifiable to the Authorised User who made the changes.

(j) Remind the Authorised User that they have legal responsibilities relating to the completion of Catch Effort forms before they digitally sign Catch Effort Forms.

(k) Allow the Authorised User to enter whatever information they wish into a user completed field within the confines detailed in the software development kit provided by FishServe (even if the application has warned the Authorised User that the data has a potential error).

(l) Allow an authorised user to print a copy of a form.

(m) Ensure that a Catch Effort Form with status "COPY" cannot be edited.

(n) Ensure that a Catch Effort Form with status "EXPORTED" cannot be edited.

(o) Ensure that a Catch Effort Form with status "ACCEPTED" cannot be edited and resubmitted.

(p) Allow an authorised user to view reference data stored by the API.

(q) Allow an authorised user to view explanatory notes stored by the API.

(r) Allow an authorised user to view maps stored by the API.

(s) Be able to initiate the transfer of the content of a Catch Effort Form completely and accurately from the API to the CE EDT System.

(t) Display receipt by the API of an electronic acknowledgement from FishServe.

(u) Display any error messages generated by the API.

7. **Off-line Tests**—All off-line tests must be completed and passed before any on-line testing is performed. Off-line tests shall be carried out by answering all the questions presented in appendix 1. Each question shall be considered a complete test on its own. Answers shall be obtained by thorough examination of the software and any documentation supplied.

Unless otherwise stated, all questions relate to mandatory requirements and must therefore be passed for compliance. A question shall be passed if its answer is the same as the correct answer indicated on the question form in appendix 1.

(a) Creation of a form – The software under test must enable an Authorised User to create each new Catch Effort Form with a unique CE Form Number.

(b) Data entry – The software under test must enable an Authorised User to enter such data as is required on a Catch Effort Form and successfully save this data.

(c) View form data – The software under test must enable the data on a previously saved and/or endorsed form to be viewed.

(d) View form event history – The software under test must allow an Authorised User to view the event history of a form.

(e) View explanatory notes – The software under test must allow the viewing of any explanatory notes.

(f) View reference data – The software under test must allow an Authorised User to determine what codes are valid options for a given field.

8. **On-line Tests**—On-line tests will only be carried out if all mandatory off-line tests are passed. On-line tests involve having the software under test communicate with the CE EDT Emulator to determine if the software can meet the requirements outlined in this circular. On-line tests shall be carried out by answering all the questions presented in appendix 2 to ascertain whether all the requirements have been met.

Unless otherwise stated, all questions relate to mandatory requirements and must therefore all be passed for the software to gain approval. A question shall be passed if its answer is the same as the correct answer indicated on the question form in appendix 2.

(a) Submit data to CE EDT System Emulator – the software under test will send catch effort data to the CE EDT System Emulator via the installed API. Correct reception of catch effort data by the CE EDT System Emulator will confirm that the data report format is correct.

(b) Notification of Acceptance or Rejection of the Catch Effort Form – the CE EDT System Emulator will send an acknowledgement notice to the software under test. Correct recognition of this data (registered as a change of status as appropriate for the Catch Effort Form) will be verified to confirm that the software can communicate with the CE EDT System via the installed API.

9. Submission for Electronic Software Approval—The software shall be approved as Electronic Software if all requirements outlined in this circular are met. The approval will be notified in the *New Zealand Gazette*. The approval shall only apply to the version of software submitted to and approved by MFish. MFish must be notified of any changes to an approved software version and reserves the right to determine whether a change constitutes a new version therefore requiring a new approval.

(a) Application requirements – The application for approval shall be made to the Chief Executive of the MFish. MFish may use approved organisations to assist with the approval process. The application shall include the following:

(i) Application Form – If required by MFish, an application form, obtainable from MFish shall be completed and form a part of the application for approval.

(ii) Application Fees – The application shall be accompanied by the relevant application fees. Details of fees due for approval are available from MFish.

(iii) Proposed Electronic Software – A copy of the software being submitted for approval.

(iv) Documentation – Any instructions that will be supplied to end users.

(b) Approval process – Upon receipt of the software, MFish shall carry out approval testing following the procedures outlined in this circular. MFish may request additional information from the applicant, and may enter into discussions with the applicant regarding how their software compares with the requirements set out in this circular. The results of the approval process shall be communicated to the applicant and the public as outlined in clauses 8(c) or 8(d).

(c) If approval is granted, the applicant and the public are notified as follows:

(i) A Certificate of Approval shall be issued by MFish to the applicant. The name of the software provider, the name of the software and the version of the software for which approval is granted shall be stated on the Certificate of Approval. MFish may impose conditions to the approval.

(ii) MFish shall notify the approval in the *New Zealand Gazette* detailing the following information where relevant:

– Software brand name and version;

– Date Electronic Software is approved;

– Installation requirements; and

– Operational requirements.

(d) If approval is not granted, MFish shall notify the applicant in writing giving reasons for its decision. The applicant should contact MFish if they want to query the declined approval.

10. Cancellation of approval—The Electronic Software approval shall no longer apply if it is cancelled by the Chief Executive of the Ministry of Fisheries. If the Electronic Software approval is cancelled:

(a) The software provider shall be notified by MFish of the details of the cancellation; and

(b) MFish shall notify the cancellation in the *New Zealand Gazette*.

Dated at Wellington this 14th day of September 2010.

WAYNE MCNEE, Chief Executive Ministry of Fisheries.

Appendix 1—Questions for Catch Effort EDT Application Off-line Testing

	Questions	**Actual answer**	**Correct answer**	**Pass Y/N**
1	For each Catch Effort return type included in the Electronic Software does the software reliably and comprehensibly (in English, using standard MFish reference codes and in a font size large enough to read) replicate each such form type as specified in Schedule 2 of the Fisheries (Reporting) Regulations 2001 (or elsewhere as specified by MFish) and record at least as many fishing or landing events as the paper form it replicates?		**Yes**	
2	Can form comment be entered and saved for each form type?		**Yes**	
3	Is it possible to create a return without a CE Form Number, or with the same CE Form Number as another return of the same form type or a CE Form Number issued to a different permit or vessel?		**No**	
4	Can the form number allocated to a created return be edited?		**No**	
5	Is it possible to create, endorse, export, import, copy, discontinue, sign or submit a return without a valid access identifier and password combination being entered by an Authorised User for the permit holder and vessel to which this CE Form Number was issued?		**No**	
6	Does each return have a form status which can be seen by anyone who views the return?		**Yes**	
7	Is it possible to enter data into a return over the course of a day (for example, entering data as soon as it becomes available)?		**Yes**	
8	Does the electronic software accurately save the data entered?		**Yes**	
9	Are all endorsed changes made to a return after it has been created accurately logged, time-stamped (including time zone) and traceable to the individual who endorsed the change?		**Yes**	
10	Is it possible to set a return to “SIGNED” status without an Authorised User digitally signing it using their Access Identifier?		**No**	

	Questions	**Actual answer**	**Correct answer**	**Pass Y/N**
11	Once a version of a return has been digitally signed, can that version of the return be amended in any way without changing the status to "IN PROGRESS"?		**No**	
12	Once a version of a return has been digitally signed, are all further amendments made to a new version of the return?		**Yes**	
13	Is there an accessible Event History record of all data entry activity both before and after a return has been Digitally Signed?		**Yes**	
14	If a form is edited and re-signed, are all the different versions identifiable?		**Yes**	
15	Is there a link to the full explanatory notes for each form type (as there would be on paper forms)?		**Yes**	
16	Does the application remind the Authorised User that they have legal responsibilities relating to the completion of Catch Effort forms before they digitally sign each Catch Effort return?		**Yes**	
17	Does the application allow the user to enter whatever information they wish into a field within the confines detailed in the software development kit provided by FishServe (even if the application has warned the user that the data has a potential error)?		**Yes**	
18	Can the user view the current version of all returns?		**Yes**	
19	Can a user using a Compliance Tool view all form Iterations and Versions?		**Yes**	
20	Can a user using a Compliance Tool copy all CE EDT data (including all Iterations and all Versions)?		**Yes**	
21	When a form is created, does it have a form status "IN PROGRESS"?		**Yes**	
22	If a user digitally signs a return, is the return status changed to "SIGNED"?		**Yes**	
23	If a user makes a copy of a return, is the status of the original return unchanged, and is the status of the copy "COPY"?		**Yes**	

	Questions	**Actual answer**	**Correct answer**	**Pass Y/N**
24	Is the creation of a copy reflected as an event in the Event History for the return?		**Yes**	
25	Is it possible to edit a form with a status of "COPY"?		**No**	
26	If a user exports a return to a different device, is the status of the original return on the source device "EXPORTED"?		**Yes**	
27	If a return is Exported, does the Exported version of the return have the same status the return had before it was Exported?		**Yes**	
28	Is it possible to export a return that has a status other than "SIGNED", or "IN PROGRESS" ?		**No**	
29	Is it possible to import a return that already exists in the application with status other than "EXPORTED"?		**No**	
30	Is it possible to edit a return with a status of "EXPORTED"?		**No**	
31	Are all Iterations of all Versions included in the export?		**Yes**	
32	Is an Event History event recorded whenever a return is either Created or Imported?		**Yes**	
33	Does the Electronic Software record the reference data set in use at the time of the data being entered by the user?		**Yes**	
34	Are the returns printed in a format that emulates the equivalent paper form?		**Yes**	
35	Does the data on the printed form accurately match the data entered into the software?		**Yes**	

Appendix 2—Questions for Catch Effort EDT Application On-line Testing

	Questions	**Actual answer**	**Correct answer**	**Pass Y/N**
1	Could the software be loaded and activated successfully from the start up package?		**Yes**	
2	Can the user initiate a synchronisation manually at any time?		**Yes**	
3	If the application allows the viewing of maps; once synchronisation has taken place are updated versions of the maps viewable?		**Yes**	
4	Once synchronisation has taken place, is the updated version of reference data viewable?		**Yes**	
5	Can the user view the number of CE Form Numbers available for each form type?		**Yes**	
6	Does the software allow the API to update each form type with a new batch of CE Forms?		**Yes**	
7	Once synchronisation has taken place, are the updated explanatory notes viewable?		**Yes**	
8	Is it possible to submit a return without having the status of “SIGNED”?		**No**	
9	Is it possible to submit a return without a valid access identifier and password combination being entered?		**No**	
10	Is it possible to submit a return without having form submission authority for that permit holder?		**No**	
11	Upon submission, is the content of a return completely and accurately transferred to the CE EDT System?		**Yes**	
12	Upon submission, are all Versions transferred to the CE EDT System?		**Yes**	
13	Do all Versions transferred to the CE EDT System accurately report the user who made the change, the nature of the change, the MAC number of the machine where the version was created and the time (including time zone) that the version was created?		**Yes**	
14	When a return is submitted, does it include the Event History of the return (in addition to all Versions of the return)?		**Yes**	

	Questions	**Actual answer**	**Correct answer**	**Pass Y/N**
15	Does the Electronic Software transmit the data to the CE EDT System using a secure internet connection?		**Yes**	
16	If the return was successfully submitted (ie a receipt notification was received), was the return saved within the Electronic Software with a status of "ACCEPTED"?		**Yes**	
17	When a submission fails due to a transmission failure, is the return saved within the Electronic Software with a status of "SUBMITTING" and able to be re-submitted when transmission is again possible?		**Yes**	
18	When a submission fails due to invalid data, is the return saved within the Electronic Software with a status of "REJECTED"?		**Yes**	
19	Does the Electronic Software register the receipt of an acknowledgement from the CE EDT System?		**Yes**	
20	Does the Electronic Software comply with the Electronic Transactions Act 2002?		**Yes**	
21	Does the Electronic Software display all error messages generated by the API?		**Yes**	

Appendix 3—Electronic Software Workflows

This appendix will define the operational workflows to which the electronic software must comply. The FishServe API will consist of several services and each service will consist of several methods that the User Interface can utilise to comply with the following workflows. For more detailed information on how these methods will operate refer to the FishServe API Software Development Kit (SDK) documentation for more technical information. At the time of this appendix being written, the method names were in line with the FishServe API SDK Documentation, which may change over the course of the standard Software Development Life Cycle (SDLC).

1. Form Data Entry Workflow

The Form Data Entry Workflow is a combination of operations that can be performed prior to submission of the form to FishServe. The form is required to be signed before it can be submitted to FishServe.

a. Create Operation

The Create Operation is the process in which the user selects to create a new form for data entry. The User Interface will use the ReturnService and invoke the CreateReturn method, upon successful completion of this method a form instance will be created in the client database. The form status will be set to “In Progress” for processing. This is a secure operation and will require the Access Identifier to be present, prior to completion of the operation. This operation will only be performed once at the beginning of the lifecycle of the form.

b. Open Operation

The Open Operation is the process in which the user will open a specific form instance from the database in edit mode. The User Interface will use the ReturnService and invoke the OpenReturn method, upon successful completion of this method a form instance will be returned from the client database in edit mode. This is a secure operation and will require the Access Identifier to be present prior to completion of the operation.

c. Save Operation

The Save Operation is the process in which the user will save his changes on a specific form instance that was placed in edit mode to be persisted to the database. These changes will not be committed until the Endorse Operation or Sign Operation has been performed on the form. The changes are held as temporary instance of the form until committed. The User Interface will use the ReturnService and invoke the SaveReturn method, upon successful completion of this method a form instance will be returned from the client database in edit mode with the user changes so that further changes could be made. The form status will be set to “In Progress” for processing, if not already set. This operation requires the Open operation to be performed prior to its execution; this is to prevent loss of data based on concurrency of the database. This operation can only be performed on forms with a status of “In Progress”, “Signed” and “Rejected”.

d. Endorse Operation

The Endorse Operation is the process in which the user will commit his changes on a specific form instance in the database. The User Interface will use the ReturnService and invoke the EndorseReturn method. Upon successful completion of this method, a form instance will be returned from the client database in edit mode with the user changes so that further changes could be made. The form status will be set to "In Progress" for processing, if not already set. This is a secure operation and will require the Access Identifier to be present prior to completion of the operation. This operation can only be performed on forms with a status of "In Progress".

e. Sign Operation

The Sign Operation is the process in which the user will digitally sign the form instance in the database. The User Interface will use the ReturnService and invoke the SignReturn method. Upon successful completion of this method, a form instance will be returned from the client database in read-only mode with the form status being set to "Signed". This is a secure operation and will require the Access Identifier to be present prior to completion of the operation. This operation can only be performed on forms with a status of "In Progress".

The above operations can be performed multiple times, unless specified by the requirements of the operation.

2. Form Submission Workflow

The Form Submission Workflow is a combination of operations that are used to submit the form to FishServe. All forms prior to submission need to be digitally signed and have a status of "Signed".

a. Prepare Operation

The Prepare Operation is the process in which the user will prepare a form instance in the database to be sent to FishServe. The User Interface will use the SubmissionService and invoke the PrepareForSubmission method. Upon successful completion of this method, a form instance will be returned from the client database in read-only mode with the form status being set to "Submitting" and the Submission status will be set to "PendingUpload". This is a secure operation and will require the Access Identifier to be present prior to completion of the operation. In order to perform this operation, the user will also be required to have submission privileges for Catch Effort. This operation can only be performed on forms with a status of "Signed".

b. Upload Operation

The Upload Operation is the process in which the user will upload prepared form instances from the database to FishServe. The User Interface will use the SubmissionService and invoke the Submit method. Upon successful completion of this method, a form instance will be returned from the client database in read-only mode with the form status being set to "Submitting" and the Submission status will be set to

"Uploading". This operation can only be performed on forms with a status of "PendingUpload".

c. Confirm Operation

The Confirm Operation is the process in which the user will confirm the status of an uploaded form instances to FishServe. The User Interface will use the SubmissionService and invoke the Confirm method. Upon successful completion of this method, a form instance will be returned from the client database in read-only mode with the form status being set to "Submitting" and the Submission status will be set to either "PendingUpload", "Confirming", "Accepted" or "Rejected". This operation can only be performed on forms with a status of "Uploading" or "Confirming".

PUBLISHED BY THE
DEPARTMENT OF INTERNAL AFFAIRS
PRINTED BY BLUE STAR GROUP

The Department of Internal Affairs
Te Tari Taiwhenua

ISSN 0111-5650
ISSN 1177-8415
Price $3.95 (inc. GST)